

ICINCO 2008

**FIFTH INTERNATIONAL CONFERENCE ON
INFORMATICS IN CONTROL, AUTOMATION AND ROBOTICS**

Proceedings

Robotics and Automation - Vol. 1

FUNCHAL, MADEIRA - PORTUGAL · MAY 11 - 15, 2008

CO-ORGANIZED BY



IN COOPERATION WITH



CO-SPONSORED BY



IEEE Systems, Man, and
Cybernetics (SMC) Society



ICINCO 2008

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

Volume RA-1

Funchal, Madeira, Portugal

May 11 – 15, 2008

Co-organized by
**INSTICC – Institute for Systems and Technologies of Information, Control
and Communication**
and
UMa – Universidade da Madeira

Co-sponsored
IEEE SMC – IEEE Systems, Man and Cybernetics Society
and
IFAC – International Federation of Automatic Control

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

Copyright © 2008 INSTICC – Institute for Systems and Technologies of
Information, Control and Communication
All rights reserved

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

Printed in Portugal

ISBN: 978-989-8111-31-9

Depósito Legal: 273830/08

<http://www.icinco.org>

secretariat@icinco.org

BRIEF CONTENTS

INVITED SPEAKERS.....	IV
ORGANIZING AND STEERING COMMITTEES	V
PROGRAM COMMITTEE	VI
AUXILIARY REVIEWERS	XI
SELECTED PAPERS BOOK	XI
FOREWORD.....	XIII
CONTENTS.....	XV

INVITED SPEAKERS

Miguel Ayala Botto

Instituto Superior Técnico

Portugal

Peter Simon Sapaty

Institute of Mathematical Machines and Systems

National Academy of Sciences

Ukraine

Ronald C. Arkin

Georgia Institute of Technology

U.S.A.

Marco Dorigo

IRIDIA, Université Libre de Bruxelles

Belgium

ORGANIZING AND STEERING COMMITTEES

CONFERENCE CO-CHAIRS

Jorge Cardoso, University of Madeira (UMa), Madeira, Portugal

Joaquim Filipe, INSTICC / Polytechnic Institute of Setúbal, Portugal

PROGRAM CO-CHAIRS

Juan Andrade Cetto, Universitat Autònoma de Barcelona, Spain

Jean-Louis Ferrier, University of Angers, France

LOCAL ARRANGEMENTS

Laura Rodriguez, University of Madeira (UMa), Portugal

PROCEEDINGS PRODUCTION

Andreia Costa, INSTICC, Portugal

Bárbara Morais, INSTICC, Portugal

Bruno Encarnação, INSTICC, Portugal

Helder Coelhas, INSTICC, Portugal

Paulo Brito, INSTICC, Portugal

Vera Coelho, INSTICC, Portugal

Vera Rosário, INSTICC, Portugal

Vitor Pedrosa, INSTICC, Portugal

CD-ROM PRODUCTION

Elton Mendes, INSTICC, Portugal

WEBDESIGNER AND GRAPHICS PRODUCTION

Marina Carvalho, INSTICC, Portugal

SECRETARIAT AND WEBMASTER

Marina Carvalho, INSTICC, Portugal

PROGRAM COMMITTEE

Arturo Hernandez Aguirre, Centre for Research in Mathematics, Mexico

Eugenio Aguirre, University of Granada, Spain

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

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, Lamar University, U.S.A.

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

Wudhichai Assawinchaichote, King Mongkut's University of Technology Thonburi, Thailand

Robert Babuska, TU Delft, The Netherlands

Ruth Bars, Budapest University of Technology and Economics, Hungary

Adil Baykasoglu, University of Gaziantep, Turkey

Laxmidhar Behera, Indian Institute of Technology, India

Maren Bennewitz, University of Freiburg, Germany

Karsten Berns, University Kaiserslautern, Germany

Arijit Bhattacharya, The Patent Office, India

Robert Bicker, Newcastle University, U.K.

Sergio Bittanti, Politecnico Di Milano, Italy

Stjepan Bogdan, University of Zagreb, Croatia

Jean-Louis Boimond, LISA, France

Djamel Bouchaffra, Grambling State University, U.S.A.

Patrick Boucher, SUPELEC, France

Guy Boy, European Institute of Cognitive Sciences and Engineering (EURISCO International), France

Bernard Brogliato, INRIA, France

Edmund Burke, University of Nottingham, U.K.

Kevin Burn, University of Sunderland, U.K.

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

Dídac Busquets, Universitat de Girona, Spain

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

Marc Carreras, University of Girona, Spain

Jorge Martins de Carvalho, FEUP, Portugal

Alessandro Casavola, University of Calabria, Italy

Riccardo Cassinis, University of Brescia, Italy

Chien Chern Cheah, Nanyang Technological University, Singapore

Tongwen Chen, University of Alberta, Canada

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

Yiu-ming Cheung, Hong Kong Baptist University, Hong Kong

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.

Danilo De Rossi, University of Pisa, Italy

Elena De Santis, University of L'Aquila, Italy

Matthias Dehmer, TU Vienna, Austria

Angel P. del Pobil, Universitat Jaume I, Spain

Mingcong Deng, Okayama University, Japan

PROGRAM COMMITTEE (CONT.)

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

Jorge Dias, ISR - Institute of Systems and Robotics, Portugal

Rüdiger Dillmann, University of Karlsruhe, Germany

Denis Dochain, Université Catholique de Louvain, Belgium

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

Alexandre Dolgui, Ecole des Mines de Saint Etienne, France

Marco Dorigo, Université Libre de Bruxelles, Belgium

Petr Ekel, Pontifical Catholic University of Minas Gerais, Brazil

Sebastian Engell, TU Dortmund, Germany

Simon Fabri, University of Malta, Malta

Sergej Fatikow, University of Oldenburg, Germany

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

Jean-Louis Ferrier, Université d'Angers, France

Limor Fix, Intel, U.S.A.

Juan F. Flores, University of Michoacan, Mexico

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

Nicholas Gans, University of Florida, U.S.A.

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.

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

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 Hagrass, University of Essex, U.K.

Wolfgang Halang, Fernuniversität, Germany

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

Uwe D. Hanebeck, Universität Karlsruhe (TH), Germany

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

Dominik Henrich, University of Bayreuth, Germany

Francisco Herrera, University of Granada, Spain

Victor HInostroza, University of Ciudad Juarez, Mexico

Wladyslaw Homenda, Warsaw University of Technology, Poland

Alamgir Hossain, Bradford University, U.K.

Dimitrios Hristu-Varsakelis, University of Macedonia, Greece

Guoqiang Hu, University of Florida, U.S.A.

Nor Ashidi Mat Isa, Universiti Sains Malaysia, Malaysia

Ray Jarvis, Monash University, Australia

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

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

Agustin Jimenez, Universidad Politécnica de Madrid, Spain

Ivan Kalaykov, Örebro University, Sweden

Michail Kalogiannakis, University Paris 5 - René Descartes, France

Dimitrios Karras, Chalkis Institute of Technology, Greece

Fakhri Karray, University of Waterloo, Canada

Dusko Katic, Mihailo Pupin Institute, Serbia

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

PROGRAM COMMITTEE (CONT.)

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

George L. Kovács, Hungarian Academy of Sciences, Hungary

Krzysztof Kozłowski, Poznan University of Technology, Poland

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

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

Cecilia Laschi, Scuola Superiore Sant'Anna, Italy

Jean-Claude Latombe, Stanford University, U.S.A.

M. Kemal Leblebicioglu, Middle East Technical University, Turkey

Loo Hay Lee, National University of Singapore, Singapore

Soo-Young Lee, KAIST, Korea

Graham Leedham, University of New South Wales, Singapore

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.

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.

Jose Tenreiro Machado, Institute of Engineering of Porto, Portugal

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

N. P. Mahalik, California State University, Fresno, U.S.A.

Bruno Maione, Politecnico di Bari, Italy

Frederic Maire, Queensland University of Technology, Australia

Om Malik, University of Calgary, Canada

Jacek Mandziuk, Warsaw University of Technology, Poland

Hervé Marchand, INRIA, France

Philippe Martinet, LASMEA, France

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), Ireland

Patrick Millot, Université de Valenciennes, France

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

Masoud Mohammadian, University of Canberra, Australia

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

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

Rafael Muñoz-Salinas, University of Cordoba, Spain

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

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 Nedeveschi, Technical University of Cluj-Napoca, Romania

Maria Neves, Instituto Superior de Engenharia do Porto, Portugal

Anton Nijholt, University of Twente, The Netherlands

Hendrik Nijmeijer, Eindhoven University of Technology, The Netherlands

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

Urbano Nunes, University of Coimbra, Portugal

Tsukasa Ogasawara, Nara Institute of Science and Technology, Japan

PROGRAM COMMITTEE (CONT.)

José Valente de Oliveira, Universidade do Algarve, Portugal

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

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

Christos Panayiotou, University of Cyprus, Cyprus

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

Evangelos Papadopoulos, NTUA, Greece

Michel Parent, INRIA, France

Igor Paromtchik, RIKEN, Japan

Mario Pavone, University of Catania, Italy

Witold Pedrycz, University of Alberta, Canada

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

Duc Pham, Cardiff University, U.K.

J. Norberto Pires, University of Coimbra, Portugal

Marios Polycarpou, University of Cyprus, Cyprus

Marie-Noëlle Pons, CNRS, France

Raul Marin Prades, Jaume I University, Spain

Libor Preucil, Czech Technical University in Prague, Czech Republic

José Ragot, Institut National Polytechnique de Lorraine, France

A. Fernando Ribeiro, Universidade do Minho, Portugal

Robert Richardson, University of Manchester, U.K.

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

António Ruano, CSI, Portugal

Fariba Sadri, Imperial College London, U.K.

Carlos Sagüés, University of Zaragoza, Spain

Mehmet Sahinkaya, University of Bath, U.K.

Priti Srinivas Sajja, Sardar Patel University, India

Antonio Sala, Universidad Politecnica de Valencia, Spain

Abdel-Badeeh Salem, Ain Shams University, Egypt

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

Matthias Scheutz, Indiana University, U.S.A.

Klaus Schilling, University Würzburg, Germany

Carla Seatzu, University of Cagliari, Italy

Rodolphe Sepulchre, University of Liege, Belgium

Michael Short, University of Leicester, U.K.

Bruno Siciliano, Università di Napoli Federico II, Italy

João Silva Sequeira, Instituto Superior Técnico, Portugal

Silvio Simani, University of Ferrara, Italy

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

Michael Small, Hong Kong Polytechnic University, China

Cyrill Stachniss, University of Freiburg, Germany

Burkhard Stadlmann, University of Applied Sciences Wels, Austria

Tarasiewicz Stanislaw, Université Laval, Canada

Olaf Stursberg, Technical University of Munich, Germany

Chun-Yi Su, Concordia University, Canada

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

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

PROGRAM COMMITTEE (CONT.)

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

Avgoustos Tsinakos, T.E.I. Kavalas, Greece

Antonios Tsourdos, Cranfield University, U.K.

Nikos Tsourveloudis, Technical University of Crete, Greece

Ivan Tyukin, University of Leicester, U.K.

Anthony Tzes, University of Patras, Greece

Masaru Uchiyama, Tohoku University, Japan

Dariusz Ucinski, University of Zielona Gora, Poland

Nicolas Kemper Valverde, Universidad Nacional Autónoma de México, Mexico

Marc Van Hulle, K. U. Leuven, Belgium

Gerrit van Straten, Wageningen University, Netherlands

Eloisa Vargiu, University of Cagliari, Italy

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

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

Luigi Villani, Università di Napoli Federico II, Italy

Bernardo Wagner, University of Hannover, Germany

Axel Walthelm, sepp.med GmbH, Germany

Dianhui Wang, La Trobe University, Australia

Lipo Wang, Nanyang Technological University, Singapore

Zidong Wang, Brunel University, U.K.

Vincent Wertz, Université catholique de Louvain, Belgium

Dirk Wollherr, Technische Universität München, Germany

Sangchul Won, Pohang University of Science and Technology, Korea

Peter Xu, Massey University, New Zealand

Bin Yao, Purdue University, U.S.A.

Xinghuo Yu, Royal Melbourne Institute of Technology, Australia

Marek Zaremba, Université du Québec, Canada

Janan Zaytoon, University of Reims Champagne Ardenne, France

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

Changjiu Zhou, Singapore Polytechnic, Singapore

Dayong Zhou, Cirrus Logic Inc., U.S.A.

Primo Zingaretti, Università Politecnica delle Marche, Italy

Argyrios Zolotas, Loughborough University, U.K.

AUXILIARY REVIEWERS

Hyo-Sung Ahn, Gwangju Institute of Science and Technology, Korea

Prasanna Balaprakash, IRIDIA, CoDE, Université Libre de Bruxelles, Belgium

Majid Chauhdry, University of Connecticut, U.S.A.

Ying Chen, University of Connecticut, U.S.A.

Pedro Fernandes, Institute of Systems and Robotics, UC, Portugal

Matteo De Felice, Univ. Roma TRE, Italy

Michele Folgheraiter, German Research Center for Artificial Intelligence, Germany

Jun Fu, Concordia University, Canada

Andrea Gasparri, University Roma TRE, Italy

Emmanuel Godoy, Supelec, France

Che Guan, University of Connecticut, U.S.A.

Istvan Harmati, Budapest University of Technology and Economics, Hungary

Abhinaya Joshi, University of Connecticut, U.S.A.

Balint Kiss, Budapest University of Technology and Economics, Hungary

Gabor Kovacs, Budapest University of Technology and Economics, Hungary

Roland Lenain, Cemagref, France

Nikolay Manyakov, K. U. Leuven, Germany

Philippe Martinet, LASMEA, Blaise Pascal University, France

Sandro Meloni, University Roma TRE, Italy

Eduardo Montijano Muñoz, University of Zaragoza, Spain

A. C. Murillo, Universidad de Zaragoza, Spain

Gonzalo Lopez Nicolas, University of Zaragoza, Spain

Sorin Olaru, Supelec, France

Federica Pascucci, Univ. Roma TRE, Italy

Karl Pauwels, K. U. Leuven, Germany

Paulo Peixoto, University of Coimbra, Portugal

Jun Peng, Chongqing University of Science and Technology, China

Luis Puig, Universidad de Zaragoza, Spain

Maurizio di Rocco, University Roma, TRE, Italy

Marco Montes De Oca Roldan, IRIDIA, CoDE, Université Libre de Bruxelles, Belgium

Joerg Stueckler, University of Freiburg, Germany

Jin Sun, Tsinghua University, China

Emese Szadeczky-Kardos, Budapest University of Technology and Economics, Hungary

Sihem Tebbani, Supelec, France

Benoit Thuilot, LASMEA, Blaise Pascal University, France

Guoyu Tu, Tsinghua University, Beijing, China

Peng Wang, University of Connecticut, U.S.A.

Weihua Wang, University of Connecticut, U.S.A.

Bingjie Zhang, University of Connecticut, U.S.A.

Yige Zhao, University of Connecticut, U.S.A.

Ying Zhao, University of Connecticut, U.S.A.

SELECTED PAPERS BOOK

A number of selected papers presented at ICINCO 2008 will be published by Springer-Verlag in a LNEE Series book. 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 2008 program committee members.

FOREWORD

This book contains the proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2008) which was organized by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC) in collaboration with the University of Madeira (UMa) and held in Madeira. ICINCO 2008 was technically co-sponsored by the IEEE Systems Man and Cybernetics Society (IEEE-SMC) and the International Federation for Automatic Control (IFAC), and held in cooperation with the Association for the Advancement of Artificial Intelligence (AAAI).

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

In the Conference Program we have included oral presentations (full papers and short papers) and posters, organized in three simultaneous tracks: “Intelligent Control Systems and Optimization”, “Robotics and Automation” and “Systems Modeling, Signal Processing and Control”. We have included in the program four plenary keynote lectures, given by internationally recognized researchers, namely - Miguel A. Botto (Instituto Superior Técnico, Portugal), Peter S. Sapaty (Institute of Mathematical Machines and Systems, National Academy of Sciences, Ukraine), Ronald C. Arkin (Georgia Institute of Technology, U.S.A.), and Marco Dorigo (IRIDIA, Université Libre de Bruxelles, Belgium). These keynote speakers participated also on a plenary panel entitled “*The new frontiers of Control, Automation and Robotics*”.

The meeting is complemented with two satellite workshops and two special sessions, focusing on specialized aspects of Informatics in Control, Automation and Robotics; namely, the International Workshop on Artificial Neural Networks and Intelligent Information Processing (ANNIIP), the International Workshop on Intelligent Vehicle Control Systems (IVCS), the Special Session on Service Oriented Architectures for SME robots and Plug-and-Produce, and the Special Session on Multi-Agent Robotic Systems.

ICINCO received 392 paper submissions, not including those of workshops and special sessions, 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 highly qualified researchers in ICINCO topic areas. Finally, only 190 papers are published in these proceedings and presented

at the conference. Of these, 114 papers were selected for oral presentation (33 full papers and 81 short papers) and 76 papers were selected for poster presentation. The full paper acceptance ratio was 8,4%, and the oral acceptance ratio (including full papers and short papers) was 29%. As in previous editions of the Conference, based on the reviewer's evaluations and the presentations, a short list of authors will be invited to submit extended versions of their papers for a book that will be published by Springer with the best papers of ICINCO 2008.

Conferences are also meeting places where collaboration projects can emerge from social contacts amongst the participants. Therefore, in order to promote the development of research and professional networks the Conference includes in its social program a Welcome Drink to all participants in the afternoon of May 11 (Sunday) and a Conference and Workshops Social Event & Banquet in the evening of May 14 (Wednesday).

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 the 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 2009, to be held in July in Milan.

Joaquim Filipe

Polytechnic Institute of Setúbal / INSTICC, Portugal

Juan Andrade-Cetto

Institut de Robotica i Informatica Industrial, CSIC-UPC, Spain

Jean-Louis Ferrier

LISA-ISTIA – Université d'Angers, France

CONTENTS

INVITED SPEAKERS

KEYNOTE LECTURES

DEALING WITH UNCERTAINTY IN THE HYBRID WORLD <i>Luis Pina and Miguel Ayala Botto</i>	IS-5
DISTRIBUTED TECHNOLOGY FOR GLOBAL DOMINANCE <i>Peter Simon Szepesy</i>	IS-15
BEHAVIORAL DEVELOPMENT FOR A HUMANOID ROBOT - Towards Life-Long Human-Robot Partnerships <i>Ronald C. Arkin</i>	IS-27
SWARM INTELLIGENCE AND SWARM ROBOTICS - The Swarm-Bot Experiment <i>Marco Dorigo</i>	IS-29

ROBOTICS AND AUTOMATION

FULL PAPERS

GENETIC-ALGORITHM SEEDING OF IDIOTYPIC NETWORKS FOR MOBILE-ROBOT NAVIGATION <i>Amanda M. Whitbrook, Uwe Aickelin and Jonathan M. Garibaldi</i>	5
DIAGNOSIS OF DISCRETE EVENT SYSTEMS WITH PETRI NETS AND CODING THEORY <i>Dimitri Lefebvre</i>	15
DYNAMIC-BASED SIMULATION FOR HUMANOID ROBOT WALKING USING WALKING SUPPORT SYSTEM <i>Aiman Musa M. Omer, Yu Ogura, Hideki Kondo, Hun-ok Lim and Atsuo Takamishi</i>	23
PEOPLE TRACKING USING LASER RANGE SCANNERS AND VISION <i>Andreas Krüßfling, Bernd Brüggemann, Dirk Schulz and Armin B. Cremers</i>	29
ROBOTIC WHEELCHAIR CONTROL CONSIDERING USER COMFORT - Modeling and Experimental Evaluation <i>Razvan Solea and Urbano Nunes</i>	37
VISUAL TRACKING ON THE GROUND - A Comparative Analysis <i>Jorge Raul Gomez, Jose J. Guerrero and Elias Herrero-Jaraba</i>	45
PARAMETER TUNING OF ROUTING PROTOCOLS TO IMPROVE THE PERFORMANCE OF MOBILE ROBOT TELEOPERATION VIA WIRELESS AD-HOC NETWORKS <i>Florian Zeiger, Nikolaus Kraemer and Klaus Schilling</i>	53
LEARNING BY EXAMPLE - Reinforcement Learning Techniques for Real Autonomous Underwater Cable Tracking <i>Andres El-Fakdi, Marc Carreras, Javier Antich and Alberto Ortiz</i>	61

VIDEO TRANSMISSION WITH ADAPTIVE QUALITY BASED ON NETWORK FEEDBACK FOR MOBILE ROBOT TELEOPERATION IN WIRELESS MULTI-HOP NETWORKS <i>Florian Zeiger, Markus Sauer and Klaus Schilling</i>	69
ALTITUDE CONTROL OF SMALL HELICOPTERS USING A PROTOTYPE TEST-BED <i>Nikos I. Vitzilaios and Nikos C. Tsourveloudis</i>	77
CLOCK SYNCHRONIZATION IN INDUSTRIAL AUTOMATION NETWORKS - Comparison of Different Syntonization Methods <i>Dragan Obradovic, Roxandra Lupas Scheiterer, Chongning Na, Günter Steindl and Franz-Josef Goetz</i>	85
TEMPORAL SMOOTHING PARTICLE FILTER FOR VISION BASED AUTONOMOUS MOBILE ROBOT LOCALIZATION <i>Walter Nisticò and Matthias Hebbel</i>	93
SLAM AND MULTI-FEATURE MAP BY FUSING 3D LASER AND CAMERA DATA <i>Ayman Zureiki, Michel Devy and Raja Chatila</i>	101
TORQUE CONTROL WITH RECURRENT NEURAL NETWORKS <i>Guillaume Jouffroy</i>	109
INTELLWHEELS - A Development Platform for Intelligent Wheelchairs for Disabled People <i>Rodrigo A. M. Braga, Marcelo Petry, Antonio Paulo Moreira and Luis Paulo Reis</i>	115
HOW TO ASSESS RELIABILITY OF INDUSTRIAL WIRELESS SOLUTIONS <i>Lutz Rauchhaupt and Marko Krätzig</i>	122
SHORT PAPERS	
WALKING PLANNING AND CONTROL FOR A BIPED ROBOT UPSTAIRS <i>Chenbo Yin, Donghua Zheng and Le Xiao</i>	133
TEMPORAL MATCH OF MULTIPLE SOURCE DATA IN AN ETHERNET BASED INDUSTRIAL ENVIRONMENT <i>Daniela Hossu and Andrei Hossu</i>	140
THE APPLICATION OF REFERENCE-PATH CONTROL TO VEHICLE PLATOONS <i>Drago Matko, Gregor Klančar, Sašo Blažič, Olivier Simonin, Franck Gächter, Jean-Michel Contet and Pablo Gruer</i>	145
DESIGN AND BALANCING CONTROL OF AIT LEG EXOSKELETON-I (ALEX-I) <i>Narong Aphiratsakun, Kittipat Chirungsarpsook and Manukeid Parnichkun</i>	151
SHOE GRINDING CELL USING VIRTUAL MECHANISM APPROACH <i>Bojan Nemeč and Leon Zlajpah</i>	159
REAL TIME GRASPING OF FREELY PLACED CYLINDRICAL OBJECTS <i>Mario Richtsfeld, Wolfgang Ponweiser and Markus Vincze</i>	165
MODIFIED LOCAL NAVIGATION STRATEGY FOR UNKNOWN ENVIRONMENT EXPLORATION <i>Sajaa Amin, Andry Tanoto, Ulf Witkowski, Ulrich Rückert and Saied Abdel-Wahab</i>	171
LegOSC - Mindstorms NXT Robotics Programming for Artists <i>Jorge Cardoso, Manuel Ferreira and Cristina Santos</i>	177
FOOTSTEP PLANNING FOR BIPED ROBOT BASED ON FUZZY Q-LEARNING APPROACH <i>Christophe Sabourin, Kurosh Madani, Weimei Yu and Jie Yan</i>	183

DYNAMICAL MODELS FOR OMNI-DIRECTIONAL ROBOTS WITH 3 AND 4 WHEELS <i>Hélder P. Oliveira, Armando J. Sousa, A. Paulo Moreira and Paulo J. Costa</i>	189
POSITION CONTROL METHOD OF A NON-CONTACTING CONVEYANCE SYSTEM FOR STEEL STRIP <i>Yeongseob Kueon, Hyoung Jin Yoon and Yoon Su Baek</i>	197
STABLE STATES TRANSITION APPROACH - A New Strategy for Walking Robots Control in Uncertain Environments <i>Anca Petrișor, Nicu George Bîzdoacă, Adrian Drighiciu, Ilie Diaconu, Sonia Degeratu, Gabriela Canureci and Gabriela Petropol Serb</i>	202
ROBOT GOES BACK HOME DESPITE ALL THE PEOPLE <i>Paloma de la Puente, Diego Rodriguez-Losada, Luis Pedraza and Fernando Matia</i>	208
USING STEREO VISION AND TACTILE SENSOR FEATURES - For Grasp Planning Control <i>Madjid Boudaba, Nikolas Gorges, Heinz Woern and Alicia Casals</i>	214
TRAFFIC SIGN RECOGNITION WITH CONSTELLATIONS OF VISUAL WORDS <i>Toon Goedemé</i>	222
A KNOWLEDGE-BASED COMPONENT FOR HUMAN-ROBOT TEAMWORK <i>Pedro Santana, Luis Correia, Mário Salgueiro, Vasco Santos and José Barata</i>	228
PROSPECTIVE ROBOTIC TACTILE SENSORS - Elastomer-Carbon Nanostructure Composites as Prospective Materials for Flexible Robotic Tactile Sensors <i>Maris Knite, Gatis Podins, Sanita Zike, Juris Zavickis and Velta Tupureina</i>	234
DCT DOMAIN VIDEO WATERMARKING - Attack Estimation and Capacity Evaluation <i>O. Dumitru, M. Mitrea and F. Prêteux</i>	239
A QUADRATIC PROGRAMMING APPROACH TO THE MINIMUM ENERGY PROBLEM OF A MOBILE ROBOT <i>Alain Segundo Potts, José Jaime da Cruz and Reinaldo Bernardi</i>	245
PARAMETERIZATION AND INITIALIZATION OF BEARING-ONLY INFORMATION - A Discussion <i>R. Aragues and C. Sagues</i>	252
EFFICIENT PLANNING OF AUTONOMOUS ROBOTS USING HIERARCHICAL DECOMPOSITION <i>Matthias Rungger, Olaf Stursberg, Bernd Spanfelner, Christian Leuxner and Wassiou Sitou</i>	262
LICENSE PLATE NUMBER RECOGNITION - New Heuristics and a Comparative Study of Classifiers <i>César García-Osorio, José-Francisco Díez-Pastor, Juan J. Rodríguez and Jesús Mandes</i>	268
OBJECT EXPLORATION WITH A HUMANOID ROBOT - Using Tactile and Kinesthetic Feedback <i>Nicolas Gorges, Stefan Gaa and Heinz Wörn</i>	274
USING THE OAG TO BUILD A MODEL DEDICATED TO MODE HANDLING OF FMS <i>Nadia Hamani, Nathalie Dangouman and Etienne Craye</i>	282
IMPLEMENTATION OF A HOMOGRAPHY-BASED VISUAL SERVO CONTROL USING A QUATERNION FORMULATION <i>T. Koenig and G. N. De Souza</i>	288
ROBOT NAVIGATION MODALITIES <i>Ray Jarvis</i>	295

A WAIT-FREE REALTIME SYSTEM FOR OPTIMAL DISTRIBUTION OF VISION TASKS ON MULTICORE ARCHITECTURES <i>Thomas Müller, Pujan Ziaie and Alois Knoll</i>	301
FROM CAD MODEL TO HUMAN-SCALE MULTIMODAL INTERACTION WITH VIRTUAL MOCK-UP - An Automotive Application <i>Damien Chamaret, Paul Richard and Jean-Louis Ferrier</i>	307
DETECTING TRANSIENT WEATHER PHENOMENA ON MARS - An Overview of the Design and Calibration of Multiparametric Detection Algorithms for the REMS/MSL Mission <i>J. Verdasca, J.-A. Manfredi and V. Peinado</i>	311
(A) VISION FOR 2050 - The Road Towards Image Understanding for a Human-Robot Soccer Match <i>Udo Frese and Tim Laue</i>	317
WHAT'S THE BEST ROLE FOR A ROBOT? - Cybernetic Models of Existing and Proposed Human- Robot Interaction Structures <i>Victoria Groom</i>	323
AUTHOR INDEX.....	329

**INVITED
SPEAKERS**

**KEYNOTE
LECTURES**

DEALING WITH UNCERTAINTY IN THE HYBRID WORLD*

Luís Pina and Miguel Ayala Botto

Department of Mechanical Engineering, IDMEC, Instituto Superior Técnico

Technical University of Lisbon, Portugal

luispina@dem.ist.utl.pt, ayalabotto@ist.utl.pt

Keywords: Hybrid systems, Hybrid Estimation, Interacting multiple-model estimation, Observability.

Abstract: This paper presents an efficient state estimation algorithm for hybrid systems based on a least-squares Interacting Multiple-Model setup. The proposed algorithm is shown to be computationally efficient when compared with the Moving Horizon Estimation algorithm that is a brute force optimization algorithm for simultaneous discrete mode and continuous state estimation of a hybrid system. The main reason has to do with the fact that the proposed algorithm is able to disregard as many discrete mode sequence estimates as possible. This is done by rapidly computing good estimates, separating the constrained and unconstrained estimates, and using some auxiliary coefficients computed off-line. The success of this state estimation algorithm is shown for a fault detection problem of the benchmark AMIRA DTS200 three-tanks system experimental setup.

1 INTRODUCTION

In the last decade hybrid systems have become a major research topic in Control Engineering (Antsaklis, 2000). Hybrid systems are dynamical systems composed by both discrete valued and continuous valued states. The dynamics of a hybrid system is governed by a mode selector that determines, at each time instant, which discrete mode is active from endogenous and/or exogenous variables. The continuous state is then updated through a dynamic relation that is selected from a set of possible dynamics according to the value of the active discrete mode. In fact, the presence of physical components such as on/off switches or valves, gears or speed selectors, or behaviors dependent on if-then-else rules imply explicitly or implicitly the discrete/continuous interaction. This interaction can be found in many real world applications such as automotive control, urban and air traffic control, communications networks, embedded control systems, and in the control of complex industrial systems via the combination of classical continuous control laws with supervisory switching logic.

The hybrid nature has attracted the interest of mathematicians, control engineers and computer scientists, therefore leading to different modeling lan-

guages and paradigms that influenced the line of research on hybrid systems in several different ways. For instance, the computer science research community is more focused on systems whose variables take values in a finite set, so adopted the discrete events modeling formalism to model hybrid systems, using finite state machines, Petri nets, temporal logic, etc. On the other hand, the control systems community typically considers a continuous valued world, where time is continuously changing, thus considering a hybrid system as described by a differential (or difference) equation with some switching mechanism. Examples of such hybrid models include Piece-Wise Affine (PWA) (Sontag, 1981) and Mixed Logical Dynamical (MLD) (Bemporad and Morari, 1999) models. A PWA model is the most intuitive representation of a hybrid system since it provides a direct relation to linear systems while still capturing very complex dynamical behaviors. However, a MLD representation is most adequate to be used in optimization problems since it is able to embed both propositional logic statements (if-then-else rules) and operating constraints in a state linear dynamics equation by transforming them to mixed-integer linear inequalities. Despite these differences, PWA and MLD are equivalent models of hybrid systems in respect to well-posedness and boundness of input, state, output or auxiliary variables (Heemels et al., 2001). This fact allows to interchange analysis and synthesis tools between them.

*This work was supported by project PTDC/EME-CRO/69117/2006 co-sponsored by FEDER, Programa Operacional Ciência e Inovação 2010, Portugal.

Research on hybrid systems spans to a wide range of topics (and approaches), from modeling to stability analysis, reachability analysis and verification, study of the observability and controllability properties, methods of state estimation and fault detection, identification techniques, and control methodologies. Typically, hybrid tools rely on the solution of optimization problems. However, due to the different nature of the optimization variables involved (integer and continuous) the main source of complexity becomes the combinatorial (yet finite) number of possible switching sequences that have to be considered. A hybrid optimal solution thus requires solving mixed-integer non-convex optimization algorithms with NP-complete complexity (Torrìsi and Bemporad, 2001).

Analysis and synthesis procedures for hybrid systems when disturbances are present either on the continuous dynamics or on the discrete mode of the hybrid system, is still an open research topic that has been tackled by several authors using distinct approaches. In the state estimation problem two distinct approaches are usually followed, the main difference being the knowledge of the active mode: some approaches consider only continuous state uncertainty with known discrete mode, while others assume that both the discrete mode and the continuous state are unknown. The combination of both uncertainties (state and mode) on the estimation process of a hybrid system presents a very difficult problem for which a global solution is not yet found. When the discrete mode is known in advance, the problem is greatly simplified and the state estimation methodologies for linear systems can be applied with very little modifications. For example in (Böker and Lunze, 2002) a bank of Kalman filters is used and in (Alessandri and Coletta, 2003) an LMI based algorithm computes the stabilizing gains for a set of Luenberger observers. If, on the other hand, the discrete mode must also be estimated the estimation problem becomes much more complex and every discrete mode sequence (*dms*) must be checked to choose the one that provides the best fit for the observed data. The continuous state estimates are then computed for the estimated *dms*. Several works address this problem, see (Balluchi et al., 2002) where a location observer is used to estimate the discrete mode and a Luenberger observer is then used to estimate the continuous state. In (Ferrari-Trecate et al., 2002) and (Pina and Botto, 2006) a Moving Horizon Estimation (MHE) scheme simultaneously estimates the discrete mode and the continuous state, differing in the fact that the latter can also estimate the input disturbances.

The derivation of the truly optimal filter for systems with switching parameters was first presented in

(Athans and Chang, 1976). The objective was to perform simultaneous system identification and state estimation for linear systems but the derivation is quite general and is directly applicable to the hybrid state estimation problem. This method requires the consideration of all admissible *dms* starting from the initial time instant, being obviously unpractical since the number of *dms* grows exponentially in time, and so, suboptimal methods were developed. From the various possibilities, considering all the admissible *dms* of a given length is usually the preferred methodology. In view of this, suboptimal multiple model estimation schemes were then developed and applied for tracking maneuvering vehicles, as surveyed in (Mazor et al., 1998), and systems with Markovian switching coefficients, (Blom and Bar-Shalom, 1988), proving their efficiency for state estimation in multiple model systems. Multiple model estimation algorithms use a set of filters, one for each possible dynamic of the system. In this paper an efficient state estimation algorithm for stochastic hybrid systems, based on the Interacting Multiple-Model (IMM) estimation algorithm, is proposed. The method is applicable to most of the existing models of hybrid systems subject to disturbances with explicitly known probability density function, so being rather general. This estimation method will be further compared to the Moving Horizon Estimation (MHE) algorithm and tested in the benchmark AMIRA DTS200 three-tanks system experimental setup.

The paper is organized as follows. Section 2 provides a description of the considered PWA model and in section 3 the proposed Interacting Multiple-Model estimation algorithm is presented. Section 4 presents an experimental application of the proposed algorithms to the AMIRA DTS200 three-tanks system experimental setup. First the experimental setup is presented and modelled, including a full characterization of all uncertainties. Then the proposed algorithms are tested and their performance is compared. Finally, in section 5 some conclusions are drawn along with some possible future developments.

2 SYSTEM DESCRIPTION

The proposed estimation algorithm is developed for PWA systems which were introduced in (Sontag, 1981). The following stochastic PWA model will be considered:

$$x(k+1) = A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} + L_{i(k)}w(k) \quad (1a)$$

$$y(k) = C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)} + v(k) \quad (1b)$$

$$\text{iff } \begin{bmatrix} x(k) \\ u(k) \\ w(k) \end{bmatrix} \in \Omega_{i(k)} \quad (1c)$$

where k is the discrete time, $x(k) \in \mathbb{X} \subset \mathbb{R}^{n_x}$ is the continuous state, $u(k) \in \mathbb{U} \subset \mathbb{R}^{n_u}$ is the input, $y(k) \in \mathbb{R}^{n_y}$ is the output, $i(k) \in I = \{1, \dots, s\}$ is the discrete mode, and s is the total number of discrete modes. The matrices and vectors A_i , B_i , f_i , L_i , C_i , D_i , g_i depend on the discrete mode $i(k)$ and have appropriate dimensions. The input disturbance $w(k)$ and the measurement noise $v(k)$ are modelled as independent identically distributed random variables, belonging to the sets \mathbb{W}_i and \mathbb{V}_i , with expected values $E\{w(k)\} = 0$, $E\{v(k)\} = 0$ and covariances Σ_{w_i} and Σ_{v_i} , respectively. These conditions are not restrictive at all since the zero mean can be imposed by summing a constant vector to the disturbances and compensated in the affine term of the system dynamics (1) and, the sets \mathbb{W}_i and \mathbb{V}_i can be considered large enough to contain all possible disturbances relevant for practical applications, for instance 99.99% of all admissible values. Notice that the input disturbance and measurement noise *pdfs* may depend on the actual mode of the system $i(k)$. The sets \mathbb{W}_i and \mathbb{V}_i are respectively defined for each mode $i(k)$ by:

$$H_{\mathbb{W}_{i(k)}} w(k) \leq h_{\mathbb{W}_{i(k)}} \quad , \quad \forall k \in \mathbb{N}_0 \quad (2)$$

$$H_{\mathbb{V}_{i(k)}} v(k) \leq h_{\mathbb{V}_{i(k)}} \quad , \quad \forall k \in \mathbb{N}_0 \quad (3)$$

The discrete mode $i(k)$ is a piecewise constant function of the state, input and input disturbance of the system whose value is defined by the regions Ω_i :

$$\Omega_i : S_i x(k) + R_i u(k) + Q_i w(k) \leq T_i \quad (4)$$

Some helpful notation regarding the time-compressed representation of (Kamen, 1992) for system (1) will now be introduced. The time-compressed representation of a system defines the dynamics of the system over a sequence of time instants in opposition to the single time step state-space representation. Consider the time interval $[k, k+T-1]$, the sequence of discrete modes over this interval is represented as $\mathbf{i}_T = \mathbf{i}_T(k) \triangleq \{i(k), \dots, i(k+T-1)\}$. To simplify the notation, the time index k is removed from the discrete mode sequence (*dms*) whenever it is obvious from the other elements in the equations. In view of this, the output sequence over the same interval can be computed by:

$$Y_T(k) = \mathbf{C}_{\mathbf{i}_T} x(k) + \mathbf{D}_{\mathbf{i}_T} U_T(k) + \mathbf{g}_{\mathbf{i}_T} + \mathbf{L}_{\mathbf{i}_T} W_T(k) + V_T(k) \quad (5)$$

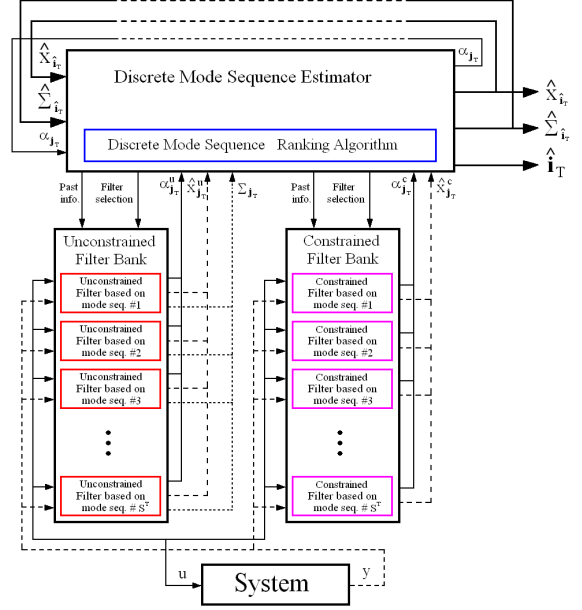


Figure 1: Interacting Multiple-Model Estimation Algorithm.

where the input, input disturbance and measurement noise sequences $U_T(k)$, $W_T(k)$ and $V_T(k)$ respectively are defined in the same way as the output sequence $Y_T(k) \triangleq [y(k)^T, \dots, y(k+T-1)^T]^T$. The matrices and vectors $\mathbf{C}_{\mathbf{i}_T}$, $\mathbf{D}_{\mathbf{i}_T}$, $\mathbf{g}_{\mathbf{i}_T}$ and $\mathbf{L}_{\mathbf{i}_T}$ are computed from the system dynamics (1a-1b) according to what is presented in (Kamen, 1992). The same reasoning can be applied to the constraints Ω_{i_T} :

$$\Omega_{\mathbf{i}_T} : \mathbf{S}_{\mathbf{i}_T} x(k) + \mathbf{R}_{\mathbf{i}_T} U_T(k) + \mathbf{Q}_{\mathbf{i}_T} W_T(k) \leq \mathbf{T}_{\mathbf{i}_T} \quad (6)$$

where the matrices $\mathbf{S}_{\mathbf{i}_T}$, $\mathbf{R}_{\mathbf{i}_T}$, $\mathbf{Q}_{\mathbf{i}_T}$ and $\mathbf{T}_{\mathbf{i}_T}$ can be computed from the system dynamics (1a) and partitions (4). The inequalities that define the disturbance and noise sets over a *dms* \mathbf{i}_T , $\mathbb{W}_{\mathbf{i}_T}$ and $\mathbb{V}_{\mathbf{i}_T}$ respectively, can also be easily found from equations (2) and (3):

$$\mathbf{H}_{\mathbb{W}_{\mathbf{i}_T}} W_T(k) \leq \mathbf{h}_{\mathbb{W}_{\mathbf{i}_T}} \quad (7)$$

$$\mathbf{H}_{\mathbb{V}_{\mathbf{i}_T}} V_T(k) \leq \mathbf{h}_{\mathbb{V}_{\mathbf{i}_T}} \quad (8)$$

3 INTERACTING MULTIPLE MODEL ESTIMATION

The proposed Interacting Multiple-Model (IMM) Estimation algorithm is composed of three parts; the Unconstrained Filter Bank (UFB), the Constrained Filter Bank (CFB) and, the Discrete Mode Sequence Estimator (DMSE). A schematic representation is presented in figure 1.

The estimation algorithm works as follows: first the continuous state estimates are computed in the UFB without considering the constraints. Then, the DMSE computes the squared errors of these estimates and ranks them. Finally, starting with the estimate with the lowest squared error, the estimates are re-computed in the CFB considering the presence of constraints. When the most accurate estimate is already a constrained estimate the whole process stops.

As the estimation is based on sequences of measurements $Y_T(k)$ and discrete modes $\mathbf{i}_T(k)$, two distinct time instants must be considered: the time instant at the beginning of the sequences, k , and the time instant at the end of these sequences, which is the present time instant $t = k+T-1$. The state estimates will be computed at time instant k , and can be propagated to the present time instant according to the estimated dynamics.

3.1 Unconstrained Filter Bank

The UFB computes the unconstrained state estimates. It is composed by a set of unconstrained least-squares filters, one for each possible dms \mathbf{j}_T :

$$\hat{x}_{\mathbf{j}_T}^u(k|t) = \hat{x}_{\mathbf{j}_T}(k|t-1) + \quad (9)$$

$$\mathbf{K}_{\mathbf{j}_T}(k|t-1) [(Y_T(k) - \mathbf{D}_{\mathbf{j}_T} U_T(k) - \mathbf{g}_{\mathbf{j}_T}) - \mathbf{C}_{\mathbf{j}_T} \hat{x}_{\mathbf{j}_T}(k|t-1)]$$

where $\hat{x}_{\mathbf{j}_T}(k|t-1)$ is the *a priori* continuous state estimate for mode sequence \mathbf{j}_T using measurements up to time instant $t-1$. $\mathbf{K}_{\mathbf{j}_T}(k|t-1)$ is the filter gain:

$$\mathbf{K}_{\mathbf{j}_T}(k|t-1) = \left(\Sigma_{x_{\mathbf{j}_T}}^{-1}(k|t-1) + \mathbf{C}_{\mathbf{j}_T}^T \Sigma_{y_{\mathbf{j}_T}}^{-1} \mathbf{C}_{\mathbf{j}_T} \right)^{-1} \mathbf{C}_{\mathbf{j}_T}^T \Sigma_{y_{\mathbf{j}_T}}^{-1} \quad (10)$$

$$\Sigma_{y_{\mathbf{j}_T}} = [\mathbf{L}_{\mathbf{j}_T} \ I_{T,n_y}] \begin{bmatrix} \Sigma_{w_{\mathbf{j}_T}} & 0 \\ 0 & \Sigma_{v_{\mathbf{j}_T}} \end{bmatrix} [\mathbf{L}_{\mathbf{j}_T} \ I_{T,n_y}]^T \quad (11)$$

The covariance of the obtained unconstrained estimate can also be computed:

$$\Sigma_{x_{\mathbf{j}_T}}(k|t) = \left(\Sigma_{x_{\mathbf{j}_T}}^{-1}(k|t-1) + \mathbf{C}_{\mathbf{j}_T}^T \Sigma_{y_{\mathbf{j}_T}}^{-1} \mathbf{C}_{\mathbf{j}_T} \right)^{-1} \quad (12)$$

This covariance matrix not only provides some insight on the accuracy of the continuous state estimate $\hat{x}_{\mathbf{j}_T}^u(k|t)$, but also defines the confidence on the past information at the subsequent time instant $\hat{x}_{\mathbf{j}_T}(k+1|t)$:

$$\Sigma_{x_{\mathbf{j}_T}}(k+1|t) = A_{j(k)} \Sigma_{x_{\mathbf{j}_T}}(k|t) A_{j(k)}^T + L_{j(k)} \Sigma_{w_{j(k)}} L_{j(k)}^T \quad (13)$$

When computing the unconstrained state estimate, no *a priori* information may be available or one may be interested in discarding it, then $\Sigma_{x_{\mathbf{j}_T}}^{-1}(k|t-1)$ should be set to 0. The corresponding unconstrained state estimate is referred to as $\hat{x}_{\mathbf{j}_T}^{u*}(k|t)$.

3.2 Constrained Filter Bank

The CFB will recompute the state estimates but now considering the constraints (6), (7) and (8). The constrained least-squares filter is somehow more complicated. First the least-squares state vector must be augmented to incorporate both the input disturbance and measurement noise vectors, since there exist explicit constraints on these variables:

$$\begin{bmatrix} x_{\mathbf{j}_T}(k) \\ \mathbf{W}_{\mathbf{j}_T}(k) \\ \mathbf{V}_{\mathbf{j}_T}(k) \end{bmatrix} \quad (14)$$

Notice that by explicitly considering the input disturbance and measurement noise sequences, all the uncertainty is removed from the observation equation (5) and it becomes an equality constraint:

$$\mathbf{H}_e \cdot \begin{bmatrix} x_{\mathbf{j}_T}(k) \\ \mathbf{W}_{\mathbf{j}_T}(k) \\ \mathbf{V}_{\mathbf{j}_T}(k) \end{bmatrix} = \mathbf{h}_e \quad \Leftrightarrow \quad (15)$$

$$\Leftrightarrow [\mathbf{C}_{\mathbf{j}_T} \ \mathbf{L}_{\mathbf{j}_T} \ I_{n_y}] \cdot \begin{bmatrix} x_{\mathbf{j}_T}(k) \\ \mathbf{W}_{\mathbf{j}_T}(k) \\ \mathbf{V}_{\mathbf{j}_T}(k) \end{bmatrix} = [Y_T(k) - \mathbf{D}_{\mathbf{j}_T} U_T(k) - \mathbf{g}_{\mathbf{j}_T}]$$

The constraints of the dms (6) and the bounds on the input disturbance and measurement noise vectors defined by the sets $\mathbb{W}_{\mathbf{j}_T}$ and $\mathbb{V}_{\mathbf{j}_T}$ described by equations (7) and (8) compose the inequality constraints of the least-squares problem, according to:

$$\mathbf{H}_i \cdot \begin{bmatrix} x_{\mathbf{j}_T}(k) \\ \mathbf{W}_{\mathbf{j}_T}(k) \\ \mathbf{V}_{\mathbf{j}_T}(k) \end{bmatrix} \leq \mathbf{h}_i \quad \Leftrightarrow \quad (16)$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{S}_{\mathbf{j}_T} & \mathbf{Q}_{\mathbf{j}_T} & 0 \\ 0 & \mathbf{H}_{w_{\mathbf{j}_T}} & 0 \\ 0 & 0 & \mathbf{H}_{v_{\mathbf{j}_T}} \end{bmatrix} \cdot \begin{bmatrix} x_{\mathbf{j}_T}(k) \\ \mathbf{W}_{\mathbf{j}_T}(k) \\ \mathbf{V}_{\mathbf{j}_T}(k) \end{bmatrix} \leq \begin{bmatrix} \mathbf{T}_{\mathbf{j}_T} - \mathbf{R}_{\mathbf{j}_T} U_T(k) \\ \mathbf{h}_{w_{\mathbf{j}_T}} \\ \mathbf{h}_{v_{\mathbf{j}_T}} \end{bmatrix}$$

Having defined the constraints matrices, the constrained least-squares filter corresponding to the mode sequence \mathbf{j}_T is given by:

$$\begin{bmatrix} \hat{x}_{\mathbf{j}_T}(k|t) \\ \hat{\mathbf{W}}_{\mathbf{j}_T}(k|t) \\ \hat{\mathbf{V}}_{\mathbf{j}_T}(k|t) \end{bmatrix} = \begin{bmatrix} \hat{x}_{\mathbf{j}_T}(z, k|t-1) \\ \hat{\mathbf{W}}_{\mathbf{j}_T}(k|t-1) \\ \hat{\mathbf{V}}_{\mathbf{j}_T}(k|t-1) \end{bmatrix} + \mathbf{K}_{\mathbf{j}_T}(k|t) \left(\begin{bmatrix} \mathbf{h}_e \\ \mathbf{h}_i \end{bmatrix} - \begin{bmatrix} \mathbf{H}_e \\ \mathbf{H}_i \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_{\mathbf{j}_T}(k|t-1) \\ \hat{\mathbf{W}}_{\mathbf{j}_T}(k|t-1) \\ \hat{\mathbf{V}}_{\mathbf{j}_T}(k|t-1) \end{bmatrix} \right) \quad (17)$$

The constrained least-squares filter gain is defined as:

$$\mathbf{K}_{\mathbf{j}_T}(k|t) = \left(\begin{bmatrix} \Sigma_{x_{\mathbf{j}_T}}(k|t-1) & 0 & 0 \\ 0 & \Sigma_{w_{\mathbf{j}_T}} & 0 \\ 0 & 0 & \Sigma_{v_{\mathbf{j}_T}} \end{bmatrix}^{-1} + \begin{bmatrix} \mathbf{H}_e \\ \mathbf{H}_i \end{bmatrix}^T \mathbf{Z}_{\mathbf{j}_T}(k|t) \begin{bmatrix} \mathbf{H}_e \\ \mathbf{H}_i \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{H}_e \\ \mathbf{H}_i \end{bmatrix}^T \mathbf{Z}_{\mathbf{j}_T}(k|t) \quad (18)$$

where $\Sigma_{x_{j_T}}(k|t-1)$ is the covariance matrix associated with the *a priori* state estimate $\hat{x}_{j_T}(k|t-1)$. $\mathbf{Z}_{j_T}(k|t)$ is the diagonal matrix that defines the active constraints.

There are several methods, most of them iterative, for determining the matrix $\mathbf{Z}_{j_T}(k|t)$, or equivalently the set of active constraints. Here, the active set method presented in (Fletcher, 1987) will be used.

As in the unconstrained case, *a priori* information may be discarded by setting $\Sigma_{x_{j_T}}^{-1}(k|t-1)$ to 0. The corresponding constrained state estimate is referred to as $\hat{x}_{j_T}^{c*}(k|t)$.

3.3 Discrete Mode Sequence Estimator

The DMSE deals with the estimation of the discrete mode sequence and, consequently, selects the filter which will provide the final continuous state estimate.

According to the least-squares philosophy, an approximation of the measured output sequence is computed for every possible *dms* and then, the one providing the smallest squared error should be selected as the least-squares estimate.

The *dms* estimate is then selected as the one that presents the lowest constrained squared error, $\alpha_{j_T}^c$:

$$\hat{\mathbf{i}}_T(k|t) = \arg \min_{j_T} \alpha_{j_T}^c(k|t) \quad (19)$$

The squared error associated with the *dms* j_T is given by:

$$\begin{aligned} \alpha_{j_T}(k|t) &= \|\hat{\mathbf{Y}}_{j_T}^*(k|t) - Y_T(k)\|_{\Sigma_{Y_{j_T}}^{-1}}^2 = \\ &= \left[\hat{\mathbf{Y}}_{j_T}^*(k|t) - Y_T(k) \right]^T \Sigma_{Y_{j_T}}^{-1} \left[\hat{\mathbf{Y}}_{j_T}^*(k|t) - Y_T(k) \right] \end{aligned} \quad (20)$$

where:

$$\hat{\mathbf{Y}}_{j_T}^*(k|t) = \mathbf{C}_{j_T} \hat{x}_{j_T}^*(k|t) + \mathbf{D}_{j_T} U_T(k) + \mathbf{g}_{j_T} \quad (21)$$

and $\hat{x}_{j_T}^*(k|t)$ is the estimated state of the *dms* j_T when all past information is discarded, ($\Sigma_{x_{j_T}}^{-1}(k|t-1) = 0$).

The squared errors computed by equation and (20) are useful when comparing continuous state estimates from the same *dms*. However, when the covariance matrices are different, an additional factor, $\bar{\alpha}_{j_T}$, must be considered to allow a meaningful comparison between squared errors. Recalling the relation between least-squares and the maximization of the Gaussian likelihood function (or its logarithm), the value of $\bar{\alpha}_{j_T}$ should be defined as:

$$\bar{\alpha}_{j_T} = -\frac{1}{2} \ln \left((2\pi)^{n_Y} \det(\Sigma_{Y_{j_T}}) \right) \quad (22)$$

Equation (20) should be modified to:

$$\alpha_{j_T}(k|t) = \bar{\alpha}_{j_T} + \|\hat{\mathbf{Y}}_{j_T}^*(k|t) - Y_T(k)\|_{\Sigma_{Y_{j_T}}^{-1}}^2 \quad (23)$$

Equation (23) can be used to compute the squared errors of both the unconstrained estimates, $\alpha_{j_T}^u(k|t)$, and the constrained estimates, $\alpha_{j_T}^c(k|t)$, using $\hat{x}_{j_T}^{u*}(k|t)$ and $\hat{x}_{j_T}^{c*}(k|t)$, respectively.

3.4 Computational Issues

Concerning computational requirements, it is noticed that there can be as many as n_s^T *dms*, which becomes an extremely large number even for relatively small n_s and T . So, computationally demanding calculations should be preformed for the minimum number of *dms* possible.

Analyzing the required computations one concludes that $\hat{x}_{j_T}^{u*}(k|t)$ can be determined by simple matrix sums and multiplications if the filter gain $\mathbf{K}_{j_T}(k|t-1)$ is computed off-line, since there are no varying terms as can be seen in equation (9). The corresponding squared error $\alpha_{j_T}^u(k|t)$, computed through equation (23), can also be determined using simple matrix sums and multiplications from $\hat{x}_{j_T}^{u*}(k|t)$. The continuous state estimate $\hat{x}_{j_T}^u(k|t)$ on the other hand, requires a matrix inversion to determine the corresponding filter gain using equation (10) since the matrix $\Sigma_{x_{j_T}}^{-1}(k|t-1)$ is not known in advance.

The constrained estimates require much more complex computations in the solution of the inequality constrained least-squares problem. An iterative algorithm has to be preformed online, and involves one matrix inversion at each iteration which is computationally heavy. There is the possibility that the solution corresponding to the true *dms* is the same as the unconstrained solution and the iterative algorithm stops at the first iteration. In general, however, this will not be the case. So, the computation of constrained solutions should only be done in cases of absolute necessity. The squared error of the constrained estimates $\alpha_{j_T}^c(k|t)$ can be determined using simple matrix sums and multiplications from $\hat{x}_{j_T}^{c*}(k|t)$.

The proposed algorithm should take these knowledge into account and arrive at the final estimates in the most efficient way possible.

To avoid the computation of the constrained least-squares estimates from all discrete mode sequences, the following relation between the constrained and unconstrained squared errors for a given discrete mode sequence is used:

$$\alpha_{j_T}^u(k|t) \leq \alpha_{j_T}^c(k|t) \quad (24)$$

An efficient reduction on the number of constrained estimates that have to be computed can be achieved by computing all unconstrained estimates $\hat{x}_{j_T}^{u*}(k|t)$ and the corresponding squared errors $\alpha_{j_T}^u(k|t)$ and then, start replacing the unconstrained solutions with the

corresponding constrained ones, from the lower values of the squared error. Whenever the lowest squared error corresponds to a constrained solution, the algorithm stops since no further reduction of the squared error can be done. The discrete mode sequence and continuous state estimates are the ones corresponding to that lowest squared error.

This algorithmic procedure may provide a substantial reduction in the number of inequality constrained least-squares problems to be solved since the increase in the squared error should be small, or even zero, for the true *dms*. However, the unconstrained solutions of incorrect *dms* may have low squared errors, which rise substantially only when the respective constrained solutions are computed. An efficient procedure to detect these incorrect *dms* before computing the respective constrained estimates would reduce the computational requirements even more.

To further improve the algorithm, the following \mathcal{B} matrix must be introduced. Each coefficient $\beta_{\mathbf{i}_T, \mathbf{j}_T}$ of the matrix \mathcal{B} is defined as the maximum value of $\alpha_{\mathbf{i}_T}^c$ under which $\alpha_{\mathbf{i}_T}^c$ is always smaller than $\alpha_{\mathbf{j}_T}^c$, or in an even more restrictive way, under which \mathbf{j}_T is never the estimated sequence. The coefficients $\beta_{\mathbf{i}_T, \mathbf{j}_T}$ can be computed off-line by the following optimization problem, which falls in the general class of Second-Order Cone Programs for which efficient solvers have already been developed, for instance, by (Alizadeh and Goldfarb, 2001):

$$\begin{aligned} \beta_{\mathbf{i}_T, \mathbf{j}_T} &= \min_{Y_T, U_T} \alpha_{\mathbf{i}_T}^c(Y_T, U_T) \\ \text{subject to :} & \\ U_T &\in \mathbb{U}^T \\ \hat{\mathbf{i}}_T &= \mathbf{j}_T \end{aligned} \quad (25)$$

By this definition of $\beta_{\mathbf{i}_T, \mathbf{j}_T}$, when the constrained solution of a *dms* \mathbf{i}_T is computed, all *dms* \mathbf{j}_T such that $\beta_{\mathbf{i}_T, \mathbf{j}_T}$ is greater than $\alpha_{\mathbf{i}_T}^c(k|t)$ can be discarded. This algorithmic procedure provides an even greater reduction on the number of constrained problems to be solved. Notice that this procedure does not even require the computation of the unconstrained solutions of the *dms* to be discarded.

Both previous modifications to the algorithm require the existence of one constrained solution to discard any other *dms*. Furthermore, the number of discarded *dms* depends on the quality of the constrained solution. In the following, some attention will be given to the recursiveness of the DMSE and the methodology to determine the *dms* that will most likely provide good constrained estimates.

At a given time instant $t+1$ the following quantities have been computed at the previous time instant: the discrete mode sequence estimate, $\hat{\mathbf{i}}_T(k|t)$, the squared errors (or lower bounds) of all *dms*, $\alpha_{\mathbf{i}_T}^c(k|t)$

and, the continuous state estimates $\hat{x}_{\mathbf{j}_T}^c(k|t)$ and the values of the estimated input disturbances $\hat{W}_{\mathbf{j}_T}(k|t)$ for the *dms* whose squared errors have been computed, including the *dms* estimate. These quantities allow the computation of the *a priori* continuous state estimate corresponding to the discrete mode sequence estimate at the following time instant:

$$\begin{aligned} \hat{x}_{\mathbf{j}_T}^*(t+1|t) &= \left(A_{j(t)} \dots A_{j(k)} \right) \hat{x}_{\mathbf{j}_T}^*(k|t) + \\ &\left[A_{j(t)} \dots A_{j(k+1)} B_{j(k)}, \dots, B_{j(t)} \right] U_T(k) + \\ &\left[A_{j(t)} \dots A_{j(k+1)} W_{j(k)}, \dots, W_{j(t)} \right] \hat{W}_{\mathbf{j}_T}(k|t) + \\ &\left(A_{j(t)} \dots A_{j(k+1)} f_{j(k)} + \dots + f_{j(t)} \right) \end{aligned} \quad (26)$$

This estimate can be used to obtain some insight on the likelihood of the discrete mode at the next time instant $j(t+1)$. The discrete modes $j(t+1)$ can be sorted by ascending values of:

$$\begin{aligned} \gamma_{\mathbf{j}_T, j}(t+1|t) &= \\ \max \left(\mathcal{S}_j \hat{x}_{\mathbf{j}_T}^*(t+1|t) + R_j u(t+1) + Q_j \hat{w}(t+1|t) - T_j \right) \end{aligned} \quad (27)$$

The value of $\hat{w}(t+1)$ should be set to $E\{w_j\}$.

The discrete modes $j(t+1)$ that provide the lower values of $\gamma_{\mathbf{j}_T, j}(t+1|t)$ correspond the discrete mode sequences $\mathbf{j}_T = \{j(k+1), \dots, j(t), j(t+1)\}$ at time instant $t+1$ most likely to succeed to \mathbf{j}_T at time instant t .

Applying this methodology to the discrete mode sequence estimate at the previous time instant, $\hat{\mathbf{i}}_T(k|t)$, should provide *dms* with very low squared errors that discard most of the other candidate *dms*. The same reasoning should be applied to all other discrete mode sequences of the previous time instant that have not been discarded yet, starting from the ones that present lowest squared errors and then the ones with the lowest bounds.

4 EXPERIMENTAL APPLICATION

To demonstrate the applicability of the hybrid estimation algorithms, the laboratory setup of the DTS200 three-tanks system from AMIRA[®] (Amira, 2002) will be used to simulate different situations common in hybrid estimation. A photo of the three-tanks system is presented in figure 2 showing the different components of the experimental setup. The plant consists of three plexiglas cylinders or tanks, T_1 , T_2 and

T_3 with similar cross section. These are connected in series with each other by cylindrical pipes with cross section S_n . Located at T_2 is the single so called nominal outflow valve V_0 which also has a circular cross section S_n . The outflowing liquid (colored distilled water) is collected in a reservoir, which supplies the pumps P_1 and P_2 . Here the water circuit is closed. h_{max} denotes the highest possible liquid level in any of the tanks. In case the liquid level of T_1 or T_2 exceeds this limit the corresponding pump will be switched off automatically. Q_1 and Q_2 are the flow rates from pumps P_1 and P_2 , respectively.

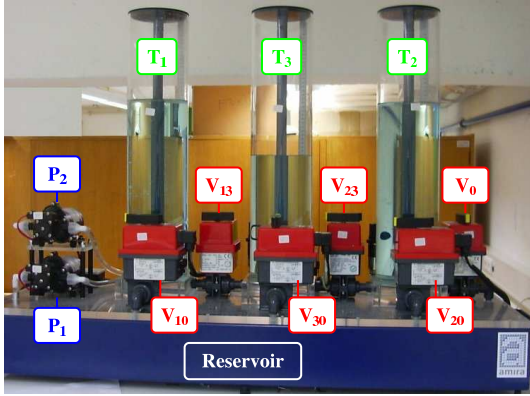


Figure 2: The three-tanks setup.

The pump flow rates Q_1 and Q_2 and the position of the valves V_{13} , V_{23} , V_0 , V_{10} , V_{20} , V_{30} , denote the controllable variables, while the liquid levels of h_1 , h_2 and h_3 are the output variables. The necessary level measurements are carried out by piezo-resistive difference pressure sensors. There are also potentiometric sensors that measure the position of each valve. The sensor signals are preprocessed to the interval $[0; 1]$ and so need to be adjusted to $[0; h_{max}]$ for the water levels. For the remainder of this section the three-tanks system will be adapted so that more realistic hybrid estimation problems can be studied while simultaneously simplifying the presentation of results. The new model is present in figure 3 where the elements in grey are assumed to be nonexistent, the elements in green are fully operational and the elements in red may be subject to faults and will be used to model input disturbances.

Pump P_1 is considered to be a fully operational on/off valve. Valve V_{13} will have two nominal values “on” and “off”, while Valve V_{10} will remain closed. Both these valves are subject to a possible fault resulting in an unmeasurable flow to cross them and described as an input disturbance. The water level sensor of tank 3 can also be subject to a fault. The Valve V_{30} is considered to be a fully operational “on/off”

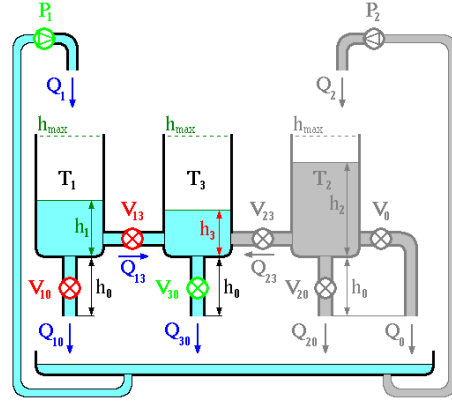


Figure 3: Final model of the three-tanks system.

valve with no possible faults, while Valves V_{20} , V_{23} and V_0 will remain closed and so can be considered to be nonexistent.

The system can exhibit a large number of different dynamics, depending on the state of each discrete variable. The full hybrid model description of the system can be found in (Pina, 2007).

4.1 Estimation of the Fault in Valve V_{10}

In this example, the estimation algorithm will have to estimate the discrete mode that indicates a fault on valve V_{10} . As the analysis will focus on valve V_{10} , the faults on valve V_{13} and sensor h_3 will be considered nonexistent. A single test will be performed where various situations arise and are then analyzed separately. The system is excited according to the discrete variables presented in table 1. Various positions for the valve V_{10} are considered, corresponding to different intensities of the fault.

Table 1: Evolution of the discrete variables.

Time(s)	0-49	50-99	100-149	150-199	200-249	250-300
V_{10}	“ok”	“faulty” “med”	“faulty” “max”	“faulty” “med”	“faulty” “max”	“ok”
V_{13}	“ok”	“ok”	“ok”	“ok”	“ok”	“ok”
h_3	“ok”	“ok”	“ok”	“ok”	“ok”	“ok”
P_1	“on”	“on”	“on”	“on”	“on”	“on”
V_{13}	“open”	“open”	“open”	“open”	“open”	“open”
V_{30}	“open”	“open”	“open”	“open”	“open”	“open”

The measured outputs and the estimated water levels are presented in figure 4, where the influence of the intensity of the fault can be clearly seen.

The real (observed) and estimated values of the fault using the IMM algorithm are shown in figure 5. As the fault in valve V_{10} takes one time instant to be reflected in the water level measurements, only the value of $f_{V_{10}}(k-1|k)$ is relevant. Note that $f_{V_{10}}(k-1|k)$ is a discrete variable that takes value 1 when a leak occurs, and value 0 when there is no fault.

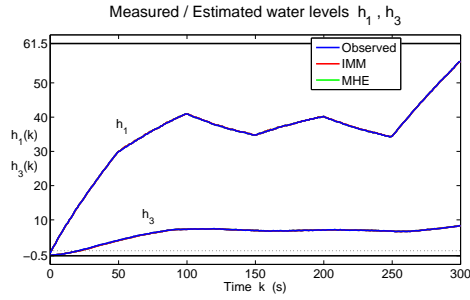


Figure 4: Water levels estimation using the IMM and MHE estimation algorithms.

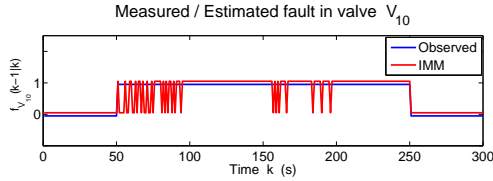


Figure 5: Estimation of the discrete mode sequence relative to the fault in valve \mathbf{V}_{10} .

The corresponding estimated continuous input disturbances by both algorithms are shown in figure 6. As the fault in valve \mathbf{V}_{10} takes one time instant to be reflected in the water level measurements, only the value of $w_{V_{10}}(k-1|k)$ is estimated. The variable $w_{V_{10}}$ determines the leaking flow and is considered to be a uniformly distributed random variable defined in the interval $[-0.4; 0.4]$ cm, with zero mean and variance $\frac{0.8^2}{12}$ cm² for all k , where 0.8 is the maximum water level change when the valve \mathbf{V}_{10} is fully open.

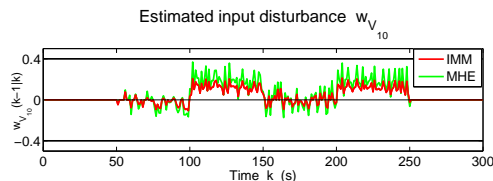


Figure 6: Estimation of the input disturbance $w_{V_{10}}(k-1|k)$ corresponding to the fault in valve \mathbf{V}_{10} .

The difference observed in both algorithms for the estimation of the disturbance $w_{V_{10}}(k-1|k)$ shows that the MHE algorithm is not able to weight the disturbance with any prior value so allowing it to change freely, which increases the variation of the input disturbance estimates.

The estimation results presented in figures 4 and 5 will now be analyzed independently for the 3 considered valve \mathbf{V}_{10} fault intensities.

4.1.1 Case 1 - Fault Inactive

For time intervals $[0; 50]$ s and $[250; 300]$ s valve \mathbf{V}_{10} remained closed and the fault is considered inactive. Despite being inactive, there is still a possibility of a wrong estimate reflected on the value of the discrete variable $f_{V_{10}}$. However, as shown in figure 7, the valve's true state was correctly estimated during these time periods.

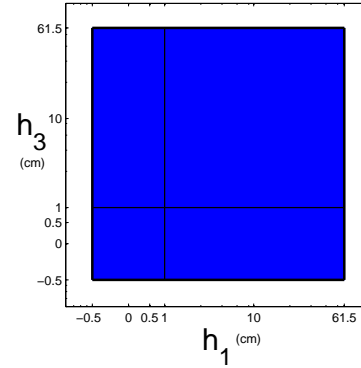


Figure 7: Map of probability of correct mode estimation, with 1s delay, when valve \mathbf{V}_{10} is fully closed. (Red - probability of correct mode estimation 0, Blue - probability of correct mode estimation 1)

Figure 7 shows that if the valve \mathbf{V}_{10} is closed there is no possibility of estimating a discrete mode sequence corresponding to an open valve condition. Thus the inactive fault is always correctly estimated.

4.1.2 Case 2 - Fault Active with Intermediate Intensity

The valve \mathbf{V}_{10} has an intermediate open position during time intervals $[50; 100]$ s and $[150; 200]$ s allowing an unmeasured flow to cross it. In this case, a fully closed valve was estimated by the IMM algorithm in several time instants. These wrong estimates are understandable since the effect on the water level of tank 1 is not too drastic and can be mistaken by any other source of uncertainty, like measurement noise for instance. This difficulty in discerning whether the valve is slightly open or fully closed is patent in the map of probability of correct mode estimation shown in figure 8. It can also be concluded that the probability of an incorrect estimation of the valve's condition increases as the water level of tank 3 becomes lower.

The map of probability of correct mode estimation is not able to show the existing dependence between the probability of correctly determining the valve's condition and its real position. It is clear from figure

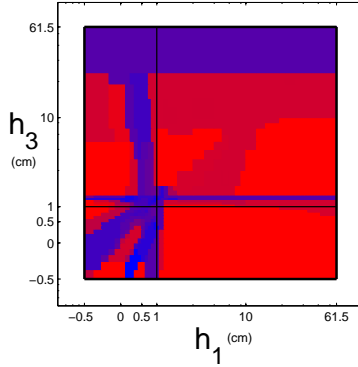


Figure 8: Map of probability of correct mode estimation, with 1s delay, when fault $f_{V_{10}}$ is active. (Red - probability of correct mode estimation 0, Blue - probability of correct mode estimation 1)

4 that the valve V_{10} is more closed during the time interval $[50 ; 100[s$ than in $[150 ; 200[s$. This fact is reflected in a higher number of incorrect mode sequence estimations in case the valve remains closer to its nominal closed position. The following case will further explore this dependence.

4.1.3 Case 3 - Fault Active with Maximum Intensity

If valve V_{10} is fully open it becomes much easier to determine its position, thus allowing the IMM algorithm to provide correct estimates for the discrete mode sequence during time intervals $[100 ; 150[s$ and $[200 ; 250[s$. This is quite obvious since the effect on the water level of tank 1 is very intense and can not be mistaken by any other source of uncertainty. This result is depicted in figure 9.

This map of probability of correct mode estimation was computed considering an hypothetical model for the system where valve V_{10} can only be fully open or fully closed.

Figure 9 shows that when the fault $f_{V_{10}}$ has maximum intensity, $w_{V_{10}} = 0.4$, it is always correctly estimated. However, further results have shown that for very low water levels in tank 1 the difference between a fully open or fully closed valve are reduced, being even undetectable when the tank is empty. This is explained by the fact that the maximum fault intensity allowed by the model, $w_{V_{10}} = 0.4$, can not be achieved in practice when tank 1 is almost empty but rather when it is full.

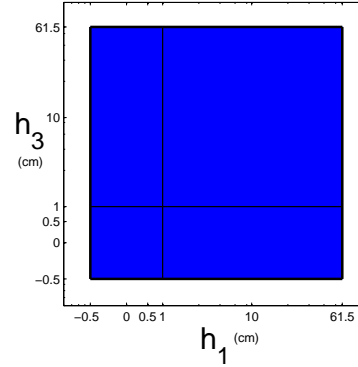


Figure 9: Map of probability of correct mode estimation, with 1s delay, considering that fault $f_{V_{10}}$ has maximum intensity, $w_{V_{10}} = 0.4$. (Red - probability of correct mode estimation 0, Blue - probability of correct mode estimation 1)

5 CONCLUSIONS

This paper presented an efficient hybrid estimation algorithm based on an IMM setup composed by a set of least-squares filters. The computational efficiency is obtained by some algorithmic procedures that discard many candidate dms before performing heavy computations. These procedures rely on the early determination of good estimates, on the separation of constrained and unconstrained estimates and on some bounding parameters for the squared errors.

The IMM was able to provide accurate online estimates for both continuous states and discrete variables when applied to the hybrid model of the benchmark AMIRA DTS200 three-tanks system experimental setup. The potential of the IMM algorithm was demonstrated when comparing its computational efficiency with the MHE with unknown inputs algorithm for a fault detection problem.

One of the most relevant issues that influence the computational efficiency of hybrid methodologies has to do with the high number of discrete modes that are typically involved in a medium size hybrid system model. This fact eventually turns most of the problems untractable. For the case of the three-tanks system experimental setup, it was noticed that the consideration of all three tanks in the same hybrid model requires huge computational resources. Thus, authors believe that a multi-agent modeling architecture can significantly simplify the all model complexity while being able to retain its full hybrid dynamical flavour. As the size of the problems to be solved with hybrid systems grows exponentially with the number of discrete modes involved, multi-agent architectures may be the solution to the huge complexity of hybrid

methodologies, thus being a very interesting and possibly fruitful research topic.

REFERENCES

- Alessandri, A. and Coletta, P. (2003). Design of observers for switched discrete-time linear systems. In *Proc. American Control Conference*, pages 2785–2790, Denver, Colorado.
- Alizadeh, F. and Goldfarb, D. (2001). Second-order cone programming. Technical Report RRR Report number 51-2001, RUTCOR, Rutgers University, Piscataway, New Jersey.
- Amira (2002). *DTS200 - Laboratory Setup Three-tank-system*. Amira, Duisburg, Germany.
- Antsaklis, P. (2000). A brief introduction to the theory and applications of hybrid systems. *Proc. IEEE, Special Issue on Hybrid Systems: Theory and Applications*, 88(7):879–886.
- Athans, M. and Chang, C. (1976). Adaptive estimation and parameter identification using multiple model estimation algorithm. Technical Report 28, M.I.T. - Lincoln Laboratory, Lexington, Massachusetts.
- Balluchi, A., Benvenuti, L., Benedetto, M. D., and Sangiovanni-Vincentelli, A. (2002). Design of observers for hybrid systems. In *Hybrid Systems: Computation and Control*, volume 2289 of *Lecture Notes in Computer Science*, pages 76–89. Springer Verlag.
- Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427.
- Blom, H. A. P. and Bar-Shalom, Y. (1988). The interactive multiple model algorithm for systems with markovian switching coefficients. *IEEE Trans. on Automatic Control*, 33(8):780–783.
- Böker, G. and Lunze, J. (2002). Stability and performance of switching Kalman filters. *International Journal of Control*, 75(16/17):1269–1281.
- Ferrari-Trecate, G., Mignone, D., and Morari, M. (2002). Moving horizon estimation for hybrid systems. *IEEE Trans. on Automatic Control*, 47(10):1663–1676.
- Fletcher, R. (1987). *Practical methods of optimization*. A Wiley Interscience Publication, Chichester, New York, 2nd edition.
- Heemels, W., Schutter, B. D., and Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091.
- Kamen, E. (1992). Study of linear time-varying discrete-time systems in terms of time-compressed models. In *Proc. 31th IEEE Conf. on Decision and Control*, pages 3070–3075, Tucson, Arizona.
- Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J. (1998). Interacting multiple model methods in target tracking: A survey. *IEEE Trans. on Aerospace and Electronic Systems*, 34(1):103–123.
- Pina, L. (2007). *Hybrid state estimation*. PhD thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal.
- Pina, L. and Botto, M. A. (2006). Simultaneous state and input estimation of hybrid systems with unknown inputs. *Automatica*, 42(5):755–762.
- Sontag, E. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Trans. on Automatic Control*, 26(2):346–358.
- Torrisi, F. and Bemporad, A. (2001). Discrete-time hybrid modeling and verification. In *Proc. 40th IEEE Conf. on Decision and Control*, pages 2899–2904, Orlando, Florida.

BRIEF BIOGRAPHY

Miguel Ayala Botto received the master degree in Mechanical Engineering in 1992 and the Ph.D. in Mechanical Engineering in 1996 from Instituto Superior Técnico, Technical University of Lisbon, Portugal. He spent the year of 1995 at the Control Laboratory, Department of Electrical Engineering, Delft University of Technology, Holland. Further, in the winter semester of the academic year 1999/2000 he held a postdoctoral position at the same laboratory. Since 2001 he is Associate Professor at the Department of Mechanical Engineering, Instituto Superior Técnico, Portugal. He is currently coordinator of the research group on Systems and Control from the Center of Intelligent Systems of IDMEC - Institute of Mechanical Engineering. Since 2005 he is the head of the Portuguese Association on Automatic Control, the National Member Organization from IFAC. He has published more than 70 journal papers, book chapters, and communications in international conferences. He has been awarded in 1999 with "The Heaviside Premium", attributed by the Council IEE - The Institution of Electrical Engineers, UK. Currently he is Associate Editor of the International Journal of Systems Science (Taylor & Francis) and member of the IFAC Technical Committee on Discrete Event and Hybrid Systems. His main research interest is in the field of estimation and control of hybrid dynamical systems.

DISTRIBUTED TECHNOLOGY FOR GLOBAL DOMINANCE

Peter Simon Sapaty

Institute of Mathematical Machines and Systems, National Academy of Sciences

Glushkova Ave 42, 03187 Kiev, Ukraine

Tel: +380-44-5265023, Fax: +380-44-5266457

sapaty@immsp.kiev.ua

Keywords: Global dominance, spatial scenarios, world processing language, distributed interpretation, emergency management, sensor networks, directed energy systems, avionics, electronic warfare, distributed objects tracking, collective behavior.

Abstract: A flexible, ubiquitous, and universal solution for management of distributed dynamic systems will be presented. It allows us to grasp complex systems on a higher than usual, semantic level, penetrating their infrastructures, also creating and modifying them, while establishing local and global dominance over the system organizations and coordinating their behavior in the way needed. The approach may allow the systems to maintain high runtime integrity and automatically recover from indiscriminate damages, preserving global goal orientation and situation awareness in unpredictable and hostile environments.

1 INTRODUCTION

We are witnessing a rapid growth of world dynamics caused by consequences of global warming, globalization of economy, numerous ethnic, religious and military conflicts, and international terrorism. To match this dynamics and withstand numerous threats and possible adversaries, effective integration of any available human and technical resources is crucial. These resources may be scattered and emergent, lacking the infrastructures and authorities for organization of the solutions needed, in real time and ahead of it.

Just communication between predetermined parts and systems with possible sharing a common vision, often called “interoperability”, may not be sufficient. The whole distributed system (or system of systems) should rather represent a highly dynamic and integral organism, in which parts may be defined and interlinked dynamically in subordination to the global organization and system goals, which can vary at runtime, with the coined term “overoperability” (Sapaty, 2002) becoming more appropriate.

A related ideology and accompanying information & control technology, allowing us to provide a much higher than usual level of system understanding and control, will be outlined in this paper.

2 THE WORLD PROCESSING PARADIGM

Within the approach developed, a network of intelligent modules (U, see in Fig. 1), embedded into important system points, collectively interprets mission scenarios in a special high-level language, which can start from any nodes, covering the networked systems at runtime.

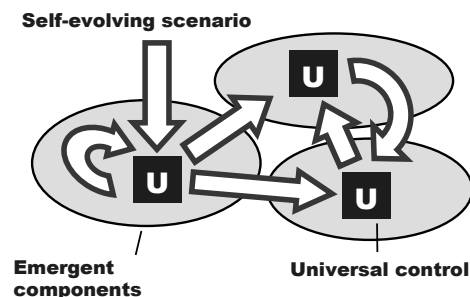


Figure 1: Runtime coverage of a distributed system.

The system “conquering” scenarios are integral and compact, being often capable of self-recovery after damages. They may be created on the fly, as traditional synchronization, data, code, and agents handling and exchanges are effectively shifted to the automatic implementation. This (parallel and fully

distributed, without central resources) spatial process can take into account details of the environments, which may be unpredictable and hostile, in which mission scenarios evolve.

Initially represented in a unified and compact form, the scenario and resources which may be needed for its development, can start from any system point (as shown in Fig.2).

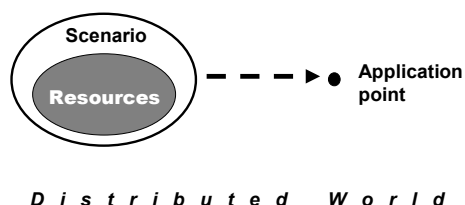


Figure 2: The initial state.

The scenarios can self-split, replicate, and modify while covering the distributed world or its part(s) needed at runtime, bringing operations and (both virtual and physical) resources into different points, also lifting, activating, and spreading further other scenarios and resources, already accumulated in the navigated world, as in Fig. 3.

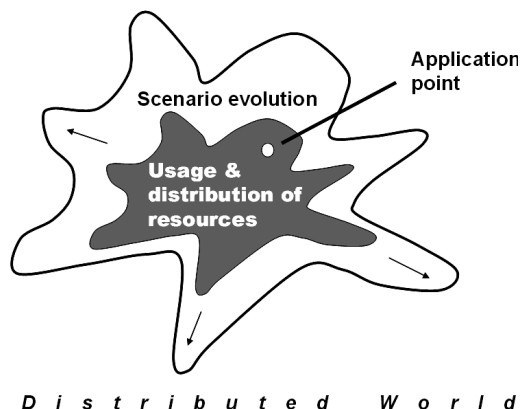


Figure3: Spreading operations and resources.

This is causing movement of information and physical matter, as well initiating interactions between manned and unmanned components, command and control (C2) including, as in Fig. 4 (S is for spatial scenarios or their parts, and R – for resources to implement the scenarios).

The main difference of this approach with the other works is that it describes on a higher level, in a concise way, of what the system should do or how should behave as a whole, while delegating numerous routines of partitioning into components (agents), with their interaction and synchronization, to the effective automatic level, while other

approaches used to do the latter manually, and from the start. The approach can, however, describe and implement the system organization and its behavior at any levels needed, which may include:

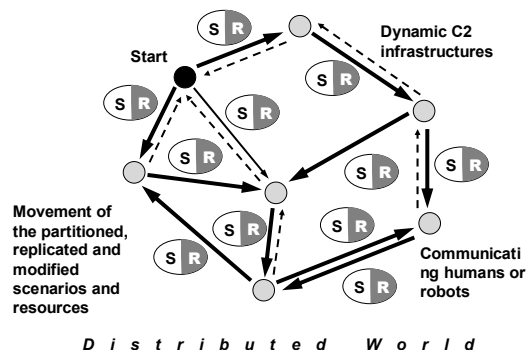


Figure 4: Resultant interactions between system parts.

- Most general, semantic, task formulation.
- Explicit projecting intelligence, information, matter, and power into particular physical or virtual locations, with doing jobs directly in the places reached, and if needed, cooperatively.
- Creating new active physical, virtual, or combined worlds, and organizing & coordinating their activity.
- Setting up implementation details, at any levels, say, for optimization of the use of scarce resources.

3 THE WORLD PROCESSING LANGUAGE (WPL)

This ideology and technology are based on the World Processing Language, WPL (Sapaty, 2005) describing what to do in distributed spaces rather than how to do, and by which resources (or even system organization), leaving these to the automatic interpretation in networked environments. The WPL fundamentals include:

- Association of any action with a position in physical, virtual, or combined space.
- Working with both information and physical matter.
- Runtime creation of distributed knowledge networks.
- Unlimited parallelism.
- Free movement or navigation in physical, virtual, or combined worlds.
- Fully distributed decision making with high integrity as a whole.
- Automatic command and control.

It is a higher-level language to efficiently command and control emergent human teams and armies. It is also a fully formal language suitable for automatic interpretation by mobile robots and their groups. Due to peculiar syntax and semantics, its parallel interpretation in distributed systems is straightforward, transparent, and does not need any central resources. Such complex problems as synchronization of multiple activities and collective (swarm as well as centrally or hierarchically controlled) behavior can be solved automatically by the networked interpreter, without traditional load on human managers and programmers.

This dramatically simplifies application programming, which is often hundreds of times more concise (and simpler) than in traditional programming languages. WPL allows for a direct access to the distributed world, performing any operations in any its points over local or remote data, which may represent both information and physical matter. Navigating in the world, WPL can modify it or even create from scratch, if required. Different movements and operations can be performed simultaneously and in parallel, and these may be free or may depend on each other.

WPL has a recursive syntax which can be expressed on the top level as follows (square brackets are for an optional construct, braces mean construct repetition with a delimiter at the right, and vertical bar separates alternatives).

```

wave    → constant | variable | [ rule ] ( {wave , } )
constant → information | matter
variable → nodal | frontal | environmental
rule    → evolution | fusion | verification | essence
evolution → expansion | branching | advancing |
repetition | granting
fusion   → echoing | processing | constructing |
assignment
verification → comparison | membership | linkage
essence  → type | usage

```

A rule is a very general construct, which, for example, can be:

- Elementary arithmetic, string or logic operation.
- Hop in physical, virtual, or combined space.
- Hierarchical fusion and return of (remote) data.
- Parallel and distributed control.
- Special context for navigation in space.
- Sense of a value for its proper interpretation.

Different types of variables, especially when used together, allow us to create efficient spatial algorithms which work “in between components” of distributed systems rather than in them. The

variables called *nodal* can store and access local results in the system points visited, while others ones can move data in space together with the evolving control (*frontal variables*) or can access and impact the world navigated (*environmental variables*).

4 ELEMENTARY EXAMPLES

4.1 Setting Global Dominance

Let us assume that a node in the distributed system (see Fig.5) wants to establish the field of its dominance over other nodes which have a lower rank than itself (here the content, or name, of each node is considered as its rank).

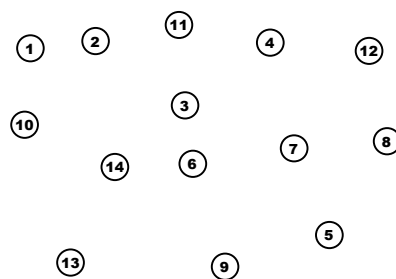


Figure 5: Distributed system nodes.

The following parallel and distributed program, applied in this node, spreads its own rank throughout the whole system in the frontal variable Rank. This puts the rank into the nodal variable Dominance in each visited node, if Rank exceeds the already existing value in Dominance (by the first access, the variable Dominance is assigned the value of the personal rank of each node).

```

frontal Rank = CONTENT;
nodal Dominance;
repeat (
  if (Dominance == nil, Dominance = CONTENT);
  if (Dominance < Rank,
    (Dominance = Rank; hop all neighbors),
    stop))

```

If applied, say, in node 11, this distributed program establishes only a partial dominance in the system, as shown in Fig. 6.

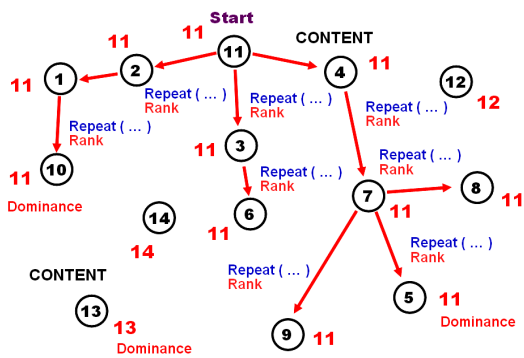


Figure 6: Resulting in partial dominance.

That will not be the case for node 14, which will set up its absolute dominance over the whole world by the program above, as in Fig. 7.

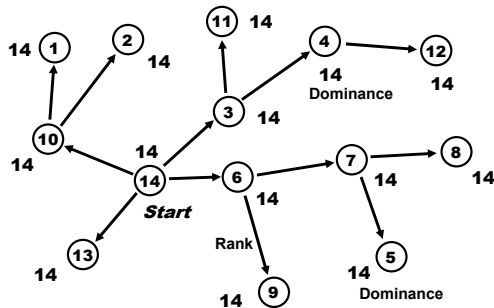


Figure 7: Setting absolute dominance of node 14.

4.2 Creating Infrastructures in the Distributed Space

It is easy to set up any infrastructures in the distributed space by the approach presented, with any topology. The following program, starting from node 3, will create (in parallel and distributed way) the networked structure shown in Fig. 8 over the set of already existing nodes.

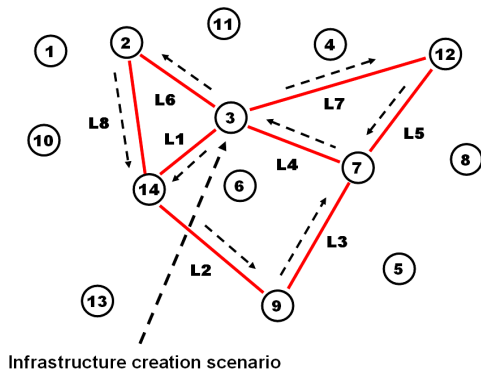


Figure 8: Creating a distributed infrastructure.

hop node 3;
create links ((L6# 2; L8#14),
(L7#12; L5#7; L4#3), (L1#14; L2#9; L3#7))

Any functionality can be associated with both nodes and links of the obtained infrastructure at runtime, which will be operating as a system for the purpose needed.

4.3 Finding Patterns in the Infrastructure

It is convenient to find any patterns in the distributed infrastructures in WPL. Let such a pattern be a triangle, and we would like to find all of them in the infrastructure created. The following spatial program, starting in any node, does this, with listing resultant nodes of the triangles in their descending ranks.

hop all nodes; frontal (Triangle) = CONTENT;
twice (hop all links; CONTENT < BACK;
Triangle &= CONTENT);
hop all links; element (Triangle, first) == CONTENT;
output Triangle

The result, issued in the node where the program was injected, will be as: (14, 3, 2), (12, 7, 3) -- see Fig.9.

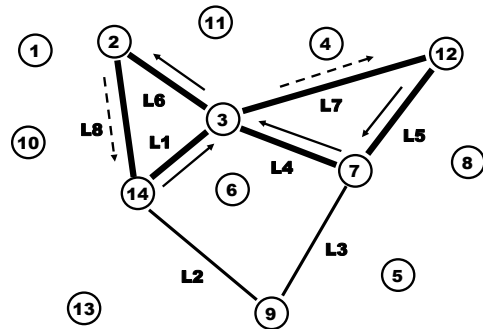


Figure 9: Finding all triangles in the infrastructure.

5 WPL INTERPRETER

The WPL interpreters may be embedded in internet hosts, robots, mobile phones, or smart sensors (an interpreter can also be a human being herself, understanding and executing high-level orders in WPL, while communicating with other humans or robots via WPL too). The interpreters may be concealed, if needed (say, to work in a hostile system); they can also migrate freely, collectively executing (also mobile) mission scenarios, resulting

altogether in the extremely flexible and ubiquitous system organization.

The basic WPL interpreter organization (Sapaty, 1993, 1999, 2005) is shown in Fig. 10, which may have both software and hardware implementation (the latter as “wave chip”).

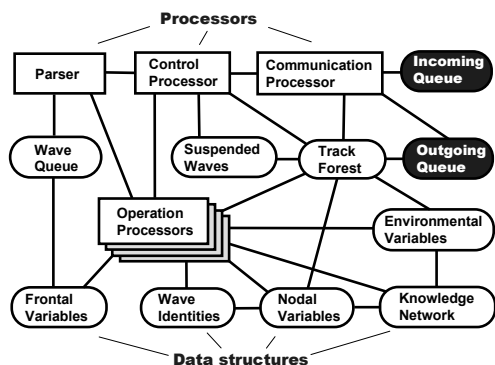


Figure 10: The WPL interpreter architecture.

The interpreter consists of a number of specialized modules working in parallel and handling and sharing specific data structures, which are supporting persistent virtual worlds and temporary hierarchical control mechanisms. The whole network of the interpreters can be mobile and open, changing the number of nodes and communication structure between them.

The heart of the distributed interpreter is its spatial track system enabling hierarchical command and control and remote data and code access, with high integrity of emerging parallel and distributed solutions. The interpreters can be embedded into any other systems, like mobile robots, allowing them to behave as integral teams, as shown in Fig. 11.

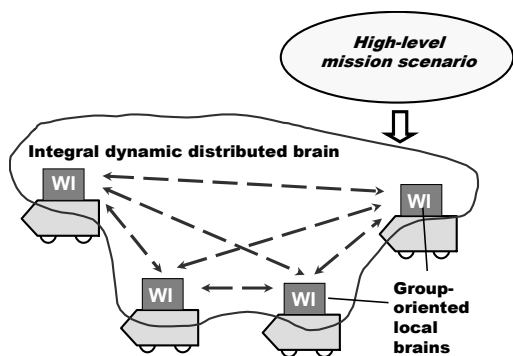


Figure 11: WPL interpreters (WI) forming distributed robotic brain.

6 EMERGENCY MANAGEMENT

Emergency management, EM (Sapaty, Sugisaka, Finkelstein, et al., 2006), due to the increased world dynamics, is becoming one of the hottest topics today. The emergency managers around the world are faced with new threats, new responsibilities, and new opportunities. Novel technologies, like the one of this paper, can alleviate consequences of natural (say, due to global warming) or manmade (like war conflicts) disasters. They can allow law enforcement and intelligence investigators to identify potential terrorist plots and then mount preemptive strikes to stop their plans.

The technology described can help in solving many EM problems by using communicating interpreters embedded in different electronic devices like, for example, laptops or mobile phones, with some disaster situation shown in Fig 12.

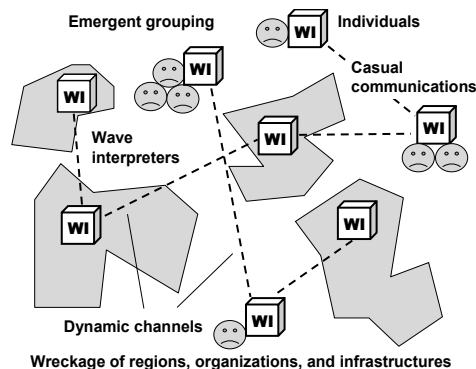


Figure 12: A disaster area with WPL interpreters embedded.

A very simple example may be here as a necessity to count the total number of casualties in the disaster area, on all its affected regions.

The following program can be applied from any WI as an entry one, which can reside within the disaster area or be away from it, and then can self-spread via local communications, organizing the whole region with embedded interpreters to work as an integral spatial supercomputer.

```
frontal Area = <disaster area definition>;
output sum (
  hop (directly, first come, nodes(Area));
  repeat(
    done(count casualties),
    hop(any links, first come, nodes(Area))))
```

More complex operations which can be organized in WPL may include the delivery of relief aid, an organized evacuation from the disaster area, and

organization of and cooperation with the rescue teams (which may include robotic components).

7 SENSOR NETWORKS

Sensor networks are a sensing, computing and communication infrastructure that allows us to instrument, observe, and respond to phenomena in the natural environment, and in our physical and cyber infrastructure. The sensors themselves can range from small passive microsensors to larger scale, controllable platforms. Typical applications of wireless sensor networks (WSN) include monitoring, tracking, and controlling. Some of the specific applications are habitat monitoring, object tracking, nuclear reactor controlling, fire detection, traffic monitoring, etc. Any distributed problems can be solved by dynamic self-organized sensor networks working in WPL (Sapaty, 2007a).

Starting from all transmitter nodes, the following program regularly (with interval of 20 sec.) covers stepwise, through local communications between sensors, the whole sensor network with a spanning forest, lifting information about observable events in each node reached, as shown in Fig. 13. Through this forest, by the internal interpretation infrastructure, the data lifted in nodes is moved and fused upwards the spanning trees, with final results collected in transmitter nodes and subsequently sent outside the system in parallel.

```

hop (all transmitters);
loop (
sleep (20);
IDENTITY = TIME;
transmit (
fuse (
repeat (free (observe (events));
hop (directly reachable, first come))))))

```

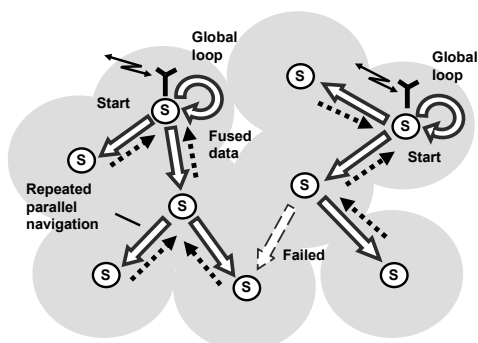


Figure 13: Collecting data by a sensor network.

Another program, below, provides for spanning tree coverage of some distributed phenomenon, with hierarchical collection, merging and fusing partial results got from different sensors into the global picture. The latter will be forwarded to a nearest transmitter via the previously created infrastructure with links infra, as shown in Fig. 14.

```

hop (random, all nodes, detected phenomenon).
loop (
frontal Full = fuse (
repeat (
free (collect phenomenon),
hop (directly reachable, first come,
detected phenomenon));
repeat (hop links (-infra)). Transmit Full)

```

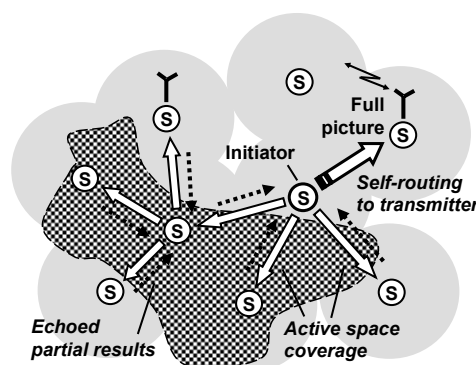


Figure 14: Space coverage with hierarchical assembling of a distributed phenomenon.

In more complex situations, which can be effectively programmed in WPL too, we may have a number of simultaneously existing phenomena, which can intersect in a distributed space. We may also face a combined phenomenon integrating features of different ones. The phenomena (like flocks of birds, manned or unmanned groups or armies, spreading fire or flooding) covering certain regions may change in size and shape, they may also move as a whole, preserving internal organization. All these situations can be managed in WPL.

8 DIRECTED ENERGY SYSTEMS

Directed energy (DE) systems are of a growing interest for broad applications in the nearest future, especially in infrastructure protection and defense. The DE-based systems will be able to operate under flexible command and control in WPL, restructuring and recovering in unpredictable environments without loss of functionality (Sapaty, Morozov, Sugisaka, 2007).

An elementary DE-based system may consist of a control center, DE source, relay mirror (RM), and target. Using WPL, the system functionality can be set up dynamically, on the fly, as by the following program:

```

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

```

Three snapshots of the system operation under this program are shown in Figs. 15-17.

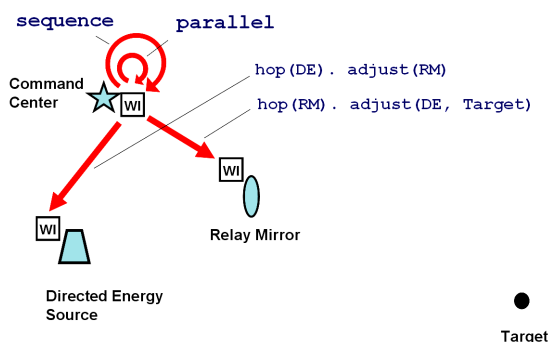


Figure 15: DE system operation, Snapshot 1.

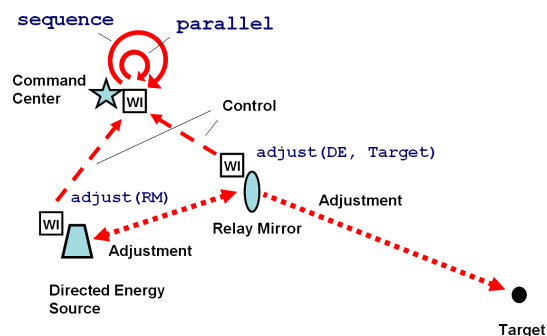


Figure 16: DE system operation, Snapshot 2.

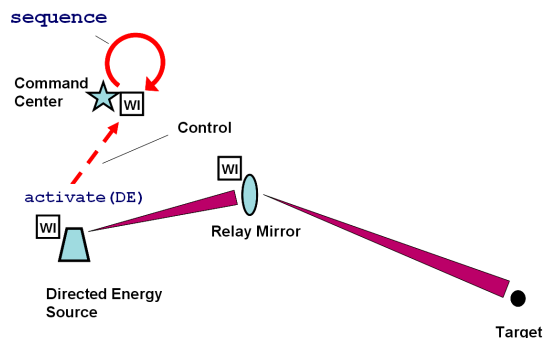


Figure 17: DE system operation, Snapshot 3.

Boeing's Advanced Relay Mirror System (ARMS) concept plans to entail a constellation of as many as two dozen orbiting mirrors that would allow a constant coverage of every corner of the globe. When activated, this would enable a directed energy response to critical trouble spots anywhere.

We will show here, be the program below, how the shortest path tree (SPT) starting from any DE source and covering the whole set of distributed mirrors can be created at runtime with the use of the technology presented. This will enable us to make optimal delivery of the directed energy to any point of the globe. The distributed SPT creation process is shown in Fig. 18.

```

nodal (Distance, Predecessor);
frontal (Length, Range = 400);
hop (DE);
Distance = 0. Length = 0;
repeat (
hop (Range, all);
Length += between (WHERE, BACK);
or (Distance == nil, Distance > Length);
Distance = Length; Predecessor = BACK)

```

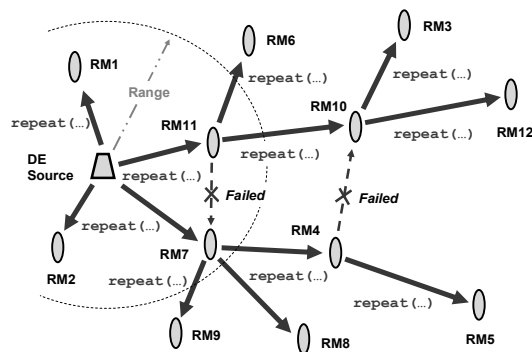


Figure 18: Dynamic shortest path tree through all RMs.

In case the target is defined, the following program forms a path from the DE source to the target via the relay mirrors, using the SPT formed, with a subsequent activation of the DE source to impact the target, as depicted in Fig. 19.

```

adjust (Seen (range), Predecessor);
repeat (
hop (Predecessor, first);
adjust (BACK, Predecessor));
activate (DE)

```

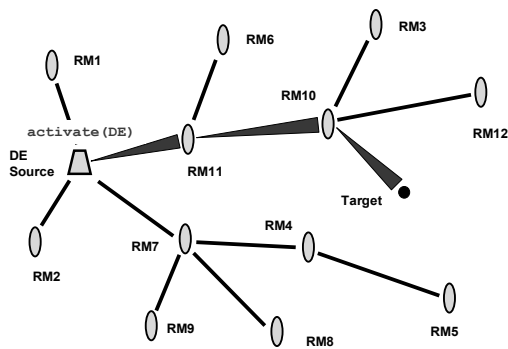


Figure 19: Energy delivery via the path found.

9 ELECTRONIC WARFARE

Electronic warfare (EW) is becoming one of the main technological challenges of this century. All existing and being developed electronic support, attack, and protection measures usually have a very limited scope and effect if used alone. But taken together they may provide a capability for fulfilling the rapidly growing needs. Traditional communication and cooperation between these systems may not be sufficient. They should comprise altogether a much more integral system of systems with global situation awareness and “global will”, which can be expressed and provided in WPL (Sapaty, 2007).

One of the typical EW tasks is fighting malicious intrusions and viruses in computer networks. Being itself a super-virus on the implementation level, the technology proposed, via the embedded network of WPL interpreters, can simultaneously discover and analyze electronic viruses, with blocking their spread and inferring attack sources. For example, the following scenario can find all virus sources in parallel, as shown in Fig 20:

```
nodal (Trace, Predecessor);
sequence (
  (hop (all nodes);
  nonempty (check general (viruses));
  repeat (
    increment (Trace);
    nonempty (Predecessor = check special
(viruses));
    hop (Predecessor))),
  output (
  sort (
    hop (all nodes); empty (Predecessor);
    nonempty (Trace); Trace & ADDRESS)))
```

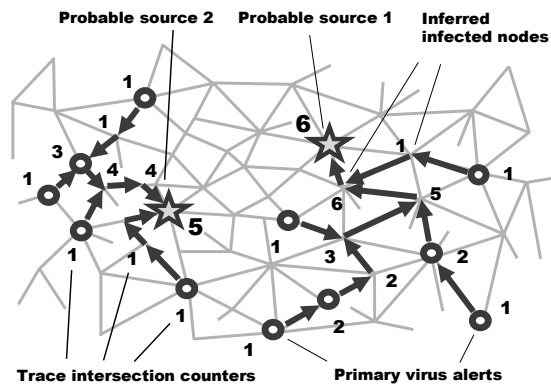


Figure 20: Finding virus sources in parallel.

10 AVIONICS

Avionics, or aviation electronics, represents a substantial share of the cost of any modern flying devices.

- Any avionics system, whether for a single aircraft or a group of them with manned or unmanned units, may be considered as a complex organization consisting of numerous components properly interacting with each other to pursue global goals. This organization can be effectively expressed in WPL on a variety of levels.
- This organization can be made flexible enough to recover from indiscriminate damages and restructure at runtime.
- The WP approach may offer real possibilities for a runtime recovery after damages, including reassembling of the whole system (or what remains of it) from any point.

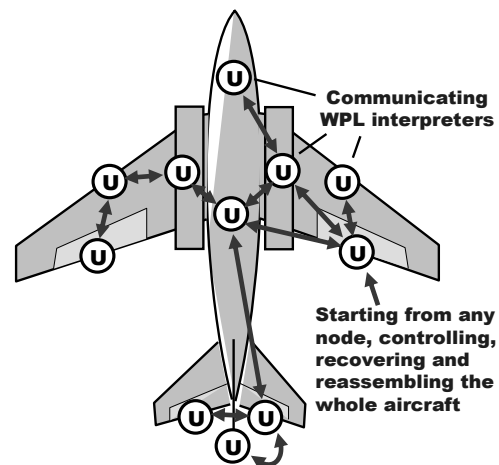


Figure 21: Aircraft self-analysis by the WPL network.

Implanting communicating WPL interpreters into main components of an aircraft, as universal control modules U (see Fig. 21), may allow us to convert the whole distributed object into a parallel computer capable of solving a variety of complex problems at runtime, including aircraft's safety and recovery (Sapaty, 2008).

The following program, starting from any point, is collecting availability of vital mechanisms of a damaged aircraft, analyzing their completeness to operate as a system, with making proper decisions (which may include the alarm with emergent evacuation of the crew).

```

nodal Available_Set =
  repeat (
    free (if CONTENT belongs_to
      (left_aileron, right_aileron, left_elevator,
       right_elevator, rudder, left_engine,
       right_engine, left_chassis,
       right_chassis, ...)
      then CONTENT),
    hop_first all_neighbors);
  if sufficient Available_Set
    then control_with Available_Set
    otherwise alarm
  
```

11 DISTRIBUTED OBJECTS TRACKING

Tracking mobile objects in distributed environments is an important task in a number of areas like air and road traffic, infrastructure protection, national and international crime, or missile defense. The example here relates to tracking aerial objects by a dynamic network of unmanned aerial vehicles, UAVs (Sapaty, 2008), with the following features to be taken into account.

- Each UAV can observe only a limited part of space.
- To keep the whole observation continuous, the object discovered should be handed over between neighboring UAVs during its movement, along with the data accumulated about it.
- The model can catch each object and accompany it individually by the mobile intelligence, while propagating between the WPL interpreters in UAVs.
- Many such objects can be picked up and chased in parallel by a dynamic UAV network.

The following program, starting in all units, catches the object it sees and follows it wherever it goes, if it is not seen from this point any more (its visibility becomes lower than a given threshold).

```

hop all_nodes; Frontal Threshold = 0.1;
frontal Object =
  select_max_visible (aerial, Threshold);
repeat (
  loop (visibility (Object) > Threshold );
  choose_destination_with_max_value (
    hop all_neighbors.
    visibility (Object) > Threshold))
  
```

A snapshot of a possible situation in a distribute space is shown in Fig. 22. The information about the tracked objects can be accumulated by individual mobile intelligences (Sapaty, Corbin, Seidensticker, 1995), which can cooperate with each other, making individual or collective decisions about the further fate of the objects (e.g. classifying them as friendly or hostile).

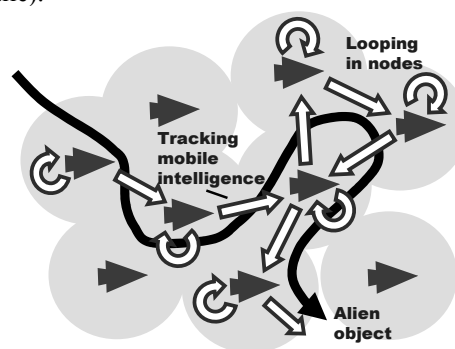


Figure 22: Collective tracking of a mobile object.

12 COLLECTIVE BEHAVIOR

The higher-level, semantic WPL scenarios are well understandable by humans, who can perform jobs written in the language and delegate other jobs to other group members, establishing runtime relations with each other. These scenarios also represent fully formal descriptions that can be effectively interpreted by robots and their groups automatically.

Both human and robotic suitability allow for a fully unified approach to organization of teams that can range from purely human to purely robotic. These teams can be open and emergent, and can operate in unpredictable environments, where team members can indiscriminately fail at any time but the mission scenario, collectively interpreted by the distributed group, can survive and fulfill objectives. The collective team behavior can be based on a loose organization like swarms, or can be strictly and hierarchically controlled. Different solutions in WPL throughout this organizational range are possible, including any combined ones (Sapaty, 2005).

With the initial distribution of units shown in Fig. 23, let us consider a collective swarm-like movement, where each unit randomly, within certain hop limits defining general direction, tries to move in new positions, keeping the established threshold distance to other units.

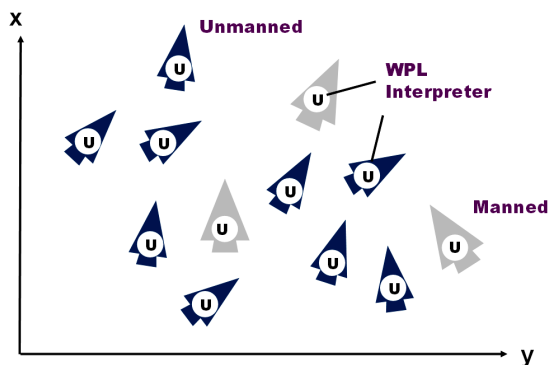


Figure 23: Initial distribution of units.

This can be done by the following program, which can start from any unit, manned or unmanned.

```
nodal (Limits, Range, Shift);
hop all_nodes;
Limits = (dx (0, 8), dy (- 2, 5)); Range = 5;
repeat (
  Shift = random (Limits );
  if empty hop (Shift, Range) then move Shift)
```

A snapshot of the group movement by this spatial program is depicted in Fig.24.

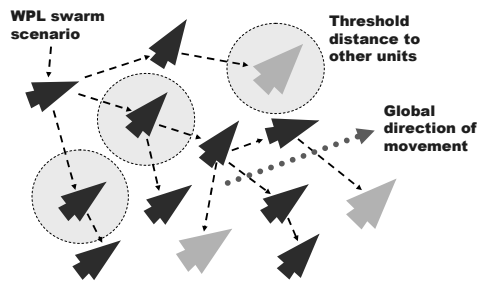


Figure 24: A swarm movement snapshot.

To have more coordinated actions of the group, we may set up a distributed hierarchical infrastructure over it, to be used in command and control and in maintaining global awareness. As the group is distributed in space and distances between units can change, such an infrastructure should be preferably based on the current physical position of the units, with top of the hierarchy to be close to the group's center, in order to optimize global coordination. We will consider here how the

topologically central unit can be found at runtime, during the movement within a swarm, and how the C2 hierarchy can be formed starting from this central unit. The following distributed program, starting from any unit, finds topologically central unit of the distributed swarm, which is shown in Fig. 25.

```
frontal Aver =
  average (hop all_nodes; WHERE);
nodal Center =
  element (
    min (
      hop all_nodes;
      distance (Aver, WHERE) & ADDRESS), 2)
```

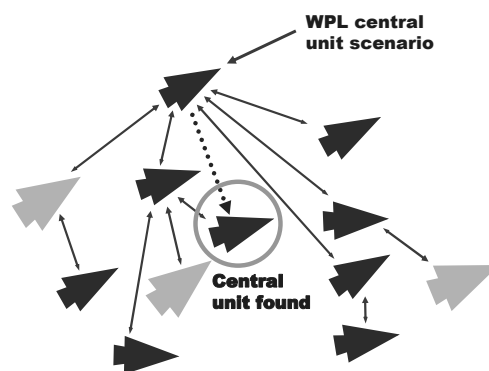


Figure 25: Finding topologically central unit.

Starting from the central unit found, the next program creates runtime hierarchical infrastructure with oriented links infra, as shown in Fig. 26.

```
frontal Range = 20.
repeat (
  create_links (
    + infra, first_come, nodes (Range)))
```

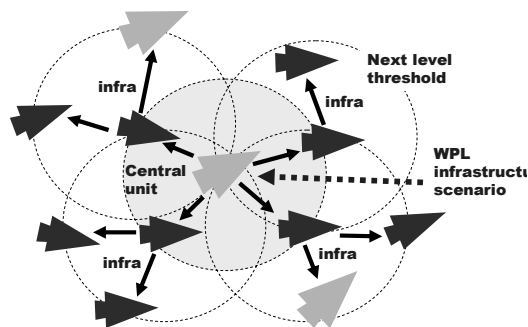


Figure 26: Hierarchical infrastructure built.

This runtime hierarchy created may be effectively used for maintaining global awareness in the distributed space, collection and fusion of targets seen by individual units, spreading the set of collected targets back to all units, which may select the most

suitable ones for an individual impact. The following program, navigating the infrastructure created, follows this scenario, as shown in Fig. 27.

```
repeat (
  if nonempty (
    frontal Seen = Repeat (
      Free (detect targets),
      Hop_links + infra)) then
    repeat (
      free (if TYPE == UAV then
        select_move_shoot Seen),
      hop_links + infra)
```

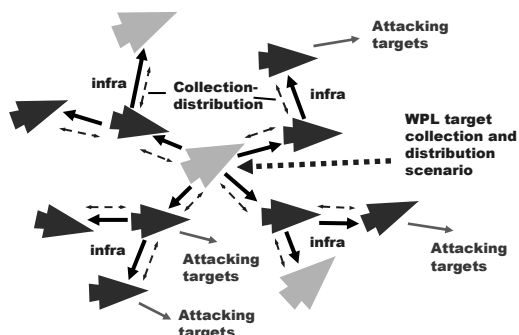


Figure 27: Hierarchical fusion and distribution of targets.

As the whole group moves and distances between separate units may change in the swarm, the programs of finding the center and hierarchical infrastructure may be repeated with a certain regularity, which will help to maintain the group's optimal spatial organization in a distributed environment. Any position in this dynamic hierarchy (the top one including) may happen to be occupied by any unit at any moment of time, regardless of whether it is manned or unmanned.

Many more applications of this world processing paradigm (previously known as WAVE) can be found in (Sapaty, 1999, 2005), also (Sapaty, Morozov, Finkelstein, et al., 2007).

13 CONCLUSIONS

We have touched only some of the areas currently in active investigation for the WP technology being developed. The experience obtained allows us to claim the following.

- The proposed technology converts any distributed system into a universal spatial computer capable of solving complex problems on itself and on the surrounding environment.

- This system, for example, can be a single unit or a group (or army) of them, with individual units being manned or unmanned.
- The whole system is driven by high-level scenarios setting how to behave as a whole and what to do, while omitting traditional implementation details which are effectively delegated to intelligent distributed interpretation system.
- The system scenarios in the World Processing Language are very compact and can be created at runtime, on the fly, swiftly reacting on a rapidly changing environment and mission goals.
- Any scenario can start from any available component and cover the system at runtime, during its evolution.
- The approach may offer real possibilities for a runtime recovery after indiscriminate damages, including reassembling of the whole system (or what remains of it) from any point.
- The technology can help dominate over other distributed system organizations, especially those explicitly based on communicating and interacting parts (agents).

REFERENCES

Sapaty, P. S., 1993. *A distributed processing system*, European Patent No. 0389655, Publ. 10.11.93, European Patent Office.

Sapaty, P. S., Corbin, M. J., Seidensticker, S., 1995. Mobile intelligence in distributed simulations, *Proc. 14th Workshop on Standards for the Interoperability of Distributed Simulations*, IST UCF, Orlando, FL, March.

Sapaty, P. S., 1999. *Mobile Processing in Distributed and Open Environments*, John Wiley & Sons, ISBN: 0471195723, New York, February, 436p.

Sapaty, P.S., 2002. Over-Operability in Distributed Simulation and Control, *The MSIAC's M&S Journal Online*, Winter Issue, Volume 4, No. 2, Alexandria, VA, USA, 9p.

Sapaty, P. S., 2005. *Ruling Distributed Dynamic Worlds*, John Wiley & Sons, New York, May, 256p, ISBN 0-471-65575-9

Sapaty, P., Sugisaka, M., Finkelstein, R., Delgado-Frias, J., Mirenkov, N., 2006. Advanced IT support of crisis relief missions, *Journal of Emergency Management*, Vol.4, No.4, July/August.

Sapaty, P., Morozov, A., Finkelstein, R., Sugisaka, M., Lambert, D., 2007. A new concept of flexible organization for distributed robotized systems. *Proc. Twelfth International Symposium on Artificial Life and Robotics (AROB 12th '07)*, Beppu, Japan, Jan 25-27, 8p.

Sapaty, P., Morozov, A., Sugisaka, M., 2007. DEW in a network enabled environment, *Proc. international conference Directed Energy Weapons 2007*, Feb. 28 - March 1, Le Meridien Piccadilly, London, UK.

- Sapaty, P., 2007. Global management of distributed EW-related systems, *Proc. Electronic Warfare: Operations & Systems 2007*, 19-20 Sept., Thistle Selfridge, London, UK.
- Sapaty, P., 2007a. Intelligent management of distributed sensor networks, In: *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VI*, edited by Edward M. Carapezza, Proc. of SPIE Vol. 6538, 653812.
- Sapaty, P., 2008. Grasping the whole by spatial intelligence: A higher level for distributed avionics, *Proc. international conference Military Avionics 2008*, Jan. 30 - Feb.1, Café Royal, London, UK.

BRIEF BIOGRAPHY

Dr. Peter Simon Sapaty, educated as power networks engineer, is with distributed systems for 40 years, implementing heterogeneous computer networks from the end of the sixties. Being chief research scientist and director of distributed simulation and control at the Institute of Mathematical Machines and Systems, National Academy of Sciences of Ukraine, also worked in Czechoslovakia, Germany, UK, Canada, and Japan as project leader, research professor, department head, and special invited professor; chaired a special interest group on mobile cooperative technologies within Distributed Interactive Simulation project in the US. Peter invented and prototyped a distributed networking technology (supported by Siemens/Nixdorf, Ericsson UK, and Japan Society for the Promotion of Science) used in different countries and resulted in a European Patent and two John Wiley books. His interests include models and languages for coordination and simulation of distributed dynamic systems with application in intelligent network control, emergency management, infrastructure protection, and cooperative robotics.

BEHAVIORAL DEVELOPMENT FOR A HUMANOID ROBOT

Towards Life-Long Human-Robot Partnerships

Ronald C. Arkin

Mobile Robot Laboratory, College of Computing, Georgia Tech, Atlanta, GA, U.S.A. 30332
arkin@cc.gatech.edu

EXTENDED ABSTRACT

A significant research effort was conducted at Sony's Intelligence Dynamics Laboratory (SIDL), involving personnel from Georgia Tech, MIT, CMU, Osaka University, and SIDL, working towards the implementation of a theory of designed development for a humanoid robot. This research involves numerous insights gleaned from cognitive psychology (drawn from both new and old theories of behavior) and integrating these techniques into Sony's humanoid robot QRIO architecture with the long-term goal of providing highly satisfying longterm interaction and attachment formation by a human partner. Included are models of deliberative (willed) reasoning and its interfacing with a reactive (automatic) controller (Glasspool 00, Shallice and Burgess 96, Ulam and Arkin 07). In particular aspects of skill transference from planned to routine activity are incorporated (Cooper and Glasspool 01, Cooper and Shallice 97, Chernova and Arkin 07). In addition, a multi-method learning technique inspired by assimilation models of Piaget provides for runtime incorporation of disparate learned skills into the existing behavioral substrate (Takamuku and Arkin 07). Finally non-verbal communication mechanisms that overlay ongoing behavior performance and utilize both proxemics (spatial separation) and kinesics (body language) are described (Brooks and Arkin 07). All of the underlying models, their implementation and the results obtained on QRIO are presented.

REFERENCES

- Brooks, A. and Arkin, R.C., "Behavioral Overlays for Non-Verbal Communication Expression on a Humanoid Robot", *Autonomous Robots*, Vol. 22, No.1, pp. 55-75, Jan. 2007.
- Chernova, S. and Arkin, R.C., "From Deliberative to Routine Behaviors: A Cognitively-Inspired Action Selection Mechanism for Routine Behavior Capture",

Adaptive Behavior, Vol. 15, No. 2, pp. 199-216, June 2007.

Cooper, R., & Glasspool, D., "Learning action affordances and action schemas", *Connectionist Models of Learning, Development, and Evolution*, 133-142, 2001.

Cooper, R., and Shallice, T., "Modeling the selection of routine action: Exploring the criticality of parameter values", in *Proceedings of the 19th annual conference of the cognitive science society* p. 130-135, 1997.

Glasspool, D., "The integration of control and behavior: Insights from neuroscience and AI", in *Proceedings of the How to Design a Functioning Mind Symposium at AISB-2000*, pp. 77-84, 2000.

Shallice, T. and Burgess, P., "The domain of supervisory processes and temporal organization of behavior", in *Philosophical Transactions of the Royal Society of London B*, vol. 351, pp. 1405-1412, 1996.

Takamuku, S. and Arkin, R.C., "Multi-method Learning and Assimilation", *Robotics and Autonomous Systems*, Vol. 55, No. 8, pp. 618-627, 2007.

Ulam, P. and Arkin, R.C., "Biasing Behavioral Activation with Intent", to appear in *Intelligent Service Robotics*, 2008.

BRIEF BIOGRAPHY

Ronald C. Arkin is Regents' Professor and the Director of the Mobile Robot Laboratory in the College of Computing at the Georgia Institute of Technology. He has held visiting positions at the Royal Institute of Technology in Stockholm, the Sony Intelligence Dynamics Laboratory in Tokyo, and LAAS/CNRS in Toulouse. Dr. Arkin's research interests include behavior-based reactive control and action-oriented perception for mobile robots and unmanned aerial vehicles, hybrid deliberative/reactive software architectures, robot survivability, multiagent robotic systems, biorobotics, human-robot interaction, robot ethics, and learning in autonomous systems. He has over 130 technical publications in these areas and has written a textbook entitled Behavior-Based Robotics and is the Series Editor for the MIT Press book series

Intelligent Robotics and Autonomous Agents. Prof. Arkin served two terms on the Administrative Committee of the IEEE Robotics and Automation Society, serves as the co-chair of the IEEE RAS Technical Committee on Robot Ethics, and also served on the National Science Foundation's Robotics Council. He was elected a Fellow of the IEEE in 2003, and is a member of AAAI and ACM.

SWARM INTELLIGENCE AND SWARM ROBOTICS

The Swarm-Bot Experiment

Marco Dorigo

IRIDIA, Université Libre de Bruxelles
Belgium

Abstract: Swarm intelligence is the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization. In particular, it focuses on the collective behaviors that result from the local interactions of the individuals with each other and with their environment. The characterizing property of a swarm intelligence system is its ability to act in a coordinated way without the presence of a coordinator or of an external controller. Swarm robotics could be defined as the application of swarm intelligence principles to the control of groups of robots. In this talk I will discuss results of Swarm-bots, an experiment in swarm robotics. A swarm-bot is an artifact composed of a swarm of assembled s-bots. The s-bots are mobile robots capable of connecting to, and disconnecting from, other s-bots. In the swarm-bot form, the s-bots are attached to each other and, when needed, become a single robotic system that can move and change its shape. S-bots have relatively simple sensors and motors and limited computational capabilities. A swarm-bot can solve problems that cannot be solved by s-bots alone. In the talk, I will shortly describe the s-bots hardware and the methodology we followed to develop algorithms for their control. Then I will focus on the capabilities of the swarm-bot robotic system by showing video recordings of some of the many experiments we performed to study coordinated movement, path formation, self-assembly, collective transport, shape formation, and other collective behaviors..

BRIEF BIOGRAPHY

Marco Dorigo received the Laurea (Master of Technology) degree in industrial technologies engineering in 1986 and the doctoral degree in information and systems electronic engineering in 1992 from Politecnico di Milano, Milan, Italy, and the title of Agrégé de l'Enseignement Supérieur, from the Université Libre de Bruxelles, Belgium, in 1995. From 1992 to 1993 he was a research fellow at the International Computer Science Institute of Berkeley, CA. In 1993 he was a NATO-CNR fellow, and from 1994 to 1996 a Marie Curie fellow. Since 1996 he has been a tenured researcher of the FNRS, the Belgian National Fund for Scientific Research, and a research director of IRIDIA-CoDE, the artificial intelligence laboratory of the Université Libre de Bruxelles. He is the inventor of the ant colony optimization metaheuristic. His current research interests include swarm intelligence, swarm robotics, and metaheuristics for discrete optimization. Dr. Dorigo is the Editor-in-Chief of the Swarm Intelligence journal. He is an Associate Editor for the IEEE Transactions on Evolutionary Computation, the IEEE Transactions on Systems, Man, and Cybernetics, and the ACM Transactions

on Autonomous and Adaptive Systems. He is a member of the Editorial Board of numerous international journals, including: Adaptive Behavior, AI Communications, Artificial Life, Cognitive Systems Research, Evolutionary Computation, Information Sciences, Journal of Heuristics and Journal of Genetic Programming and Evolvable Machines. In 1996 he was awarded the Italian Prize for Artificial Intelligence, in 2003 the Marie Curie Excellence Award, and in 2005 the Dr A. De Leeuw-Damry-Bourlart award in applied sciences. He is a fellow of the IEEE and of the ECCAI, the European Coordinating Committee for Artificial Intelligence.

**ROBOTICS
AND AUTOMATION - 1**

FULL PAPERS

GENETIC-ALGORITHM SEEDING OF IDIOTYPIC NETWORKS FOR MOBILE-ROBOT NAVIGATION

Amanda M. Whitbrook, Uwe Aickelin and Jonathan M. Garibaldi
*ASAP Research Group, School of Computer Science, University of Nottingham, Jubilee Campus
Wollaton Road, Nottingham, NG8 1BB, U.K.
amw@cs.nott.ac.uk, uxa@cs.nott.ac.uk, jmg@cs.nott.ac.uk*

Keywords: Mobile-robot navigation, genetic algorithm, artificial immune system, idiotypic network.

Abstract: Robot-control designers have begun to exploit the properties of the human immune system in order to produce dynamic systems that can adapt to complex, varying, real-world tasks. Jerne's idiotypic-network theory has proved the most popular artificial-immune-system (AIS) method for incorporation into behaviour-based robotics, since idiotypic selection produces highly adaptive responses. However, previous efforts have mostly focused on evolving the network connections and have often worked with a single, pre-engineered set of behaviours, limiting variability. This paper describes a method for encoding behaviours as a variable set of attributes, and shows that when the encoding is used with a genetic algorithm (GA), multiple sets of diverse behaviours can develop naturally and rapidly, providing much greater scope for flexible behaviour-selection. The algorithm is tested extensively with a simulated e-puck robot that navigates around a maze by tracking colour. Results show that highly successful behaviour sets can be generated within about 25 minutes, and that much greater diversity can be obtained when multiple autonomous populations are used, rather than a single one.

1 INTRODUCTION

Short-term learning can be defined as the training that takes place over the lifetime of an individual, and long-term learning as that which evolves and develops as a species interacts with its environment and reproduces itself. The vertebrate immune system draws on both types since, at birth, an individual possesses a pool of antibodies that has evolved over the lifetime of the species; the repertoire also adapts and changes over the lifetime of the individual as the living body responds to invading antigens. Recently, researchers have been inspired by the learning and adaptive properties of the immune system when attempting to design effective robot-navigation systems. Many artificial-immune-system (AIS) methodologies adopt the analogy of antibodies as robot behaviours and antigens as environmental stimuli. Farmer's computational model (Farmer *et al.*, 1986) of Jerne's idiotypic-network theory (Jerne, 1974), which assumes this relation, has proved an extremely popular choice, since the antibody (behaviour) that best matches the invading antigen (current environment) is not necessarily selected for execution, producing a flexible and dynamic system.

The idiotypic architecture has produced some encouraging results, but has generally suffered from the same problems as previous approaches, as most designs have used small numbers of pre-engineered behaviours, limiting the self-discovery and learning properties of the schemes. This research aims to solve the problem by encoding behaviours as a set of variable attributes and using a genetic-algorithm (GA) to obtain diverse sets of antibodies for seeding the AIS. Here, the first phase of the design is described, i.e. the long-term phase that seeks to produce the initial pool of antibodies.

The long-term phase is carried out entirely in simulation so that it can execute as rapidly as possible by accelerating the simulations to maximum capacity. The population size is varied and, in addition, two different population models are considered, since it is imperative that an idiotypic system is able to select from a number of very diverse behaviours. In the first scheme there is only one population, but in the second, separate populations evolve in series but never interbreed. In each case the derived antibody-sets are scored in terms of diversity, solution quality, and how quickly they evolve.

The paper is arranged as follows. Section 2 shows how the vertebrate immune system depends on both short-term and long-term learning, and discusses how AIS has been used as a model for robotic controllers. It also highlights some of the problems with previous approaches to AIS robot-control and with evolutionary robotics in general. Section 3 describes the test environments and the problem used, and section 4 focuses on the architecture of the long-term phase including details of the GA. The experimental procedures are outlined in Section 5 and the results are presented and discussed in Section 6. Section 7 concludes the paper.

2 BACKGROUND AND MOTIVATION

Throughout the lifetime of an individual, the adaptive immune system learns to recognise antigens by building up high concentrations of antibodies that have proved useful in the past, and by eliminating those deemed redundant. This is a form of short-term learning. However, the antibody repertoire is not random at birth and the mechanism by which antibodies are replaced is not a random process. Antibodies are built from gene libraries that have evolved over the lifetime of the species. This demonstrates that the immune system depends on both short-term and long-term learning in order to achieve its goals.

When using the immune system as inspiration for robot controllers, many researchers opt to implement an idiotypic network based on Farmer's model of continuous antibody-concentration change. In this model the concentrations are not only dependent on the antigens, but also on the other antibodies present in the system, i.e. antibodies are suppressed and stimulated by each other as well as being stimulated by antigens. In theory this design permits great variability of robot behaviour since the antibodies model the different behaviours, and the complex dynamics of stimulation and suppression ensure that alternative antibodies are tried when the need arises (Whitbrook *et al.*, 2007). However, past work in this area has mostly focused on how the antibodies in the network should be connected and, for simplicity, has used a single set of pre-engineered behaviours for the antibodies, which limits the potential of the method. For example, Watanabe *et al.* (1998a, 1998b) use an idiotypic network to control a garbage-collecting robot,

utilizing GAs to evolve their initial set of antibodies. The antibodies are composed of a precondition, a behaviour, and an idiotope part that defines antibody connection. However, the sets of possible behaviours and preconditions are fixed; the GA works simply by mixing and evolving different combinations with various parameters for the idiotope. Michelan and Von Zuben (2002) and Vargas *et al.* (2003) also use GAs to evolve the antibodies, but again only the idiotypic-network connections are derived. Krautmacher and Dilger (2004) apply the idiotypic method to robot navigation, but their emphasis is on the use of a variable set of antigens; they do not change or develop the initial set of handcrafted antibodies, as only the network links are evolved. Luh and Liu (2004) address target-finding using an idiotypic system, modelling their antibodies as steering directions. However, although many behaviours are technically possible since any angle can be selected, the method is limited because a behaviour is defined only as a steering angle and there is no scope for the development of more complex functions. Hart *et al.* (2003) update their network links dynamically using reinforcement learning, but use a skill hierarchy so that more complex tasks are achieved by building on basic ones, which are hand-designed at the start.

It is clear that the idiotypic AIS methodology holds great promise for providing a system that can adapt to change, but its potential has never been fully explored because of the limits imposed on the fundamental behaviour-set. This research aims to widen the scope of the idiotypic network by providing a technique that rapidly evolves simple, distinct behaviours in simulation. The behaviours can then be passed to a real robot as a form of intelligent initialization, i.e. a starting set of behaviours would be available for each known antigen, from which the idiotypic selection-mechanism could pick.

In addition, long-term learning in simulation coupled with an idiotypic AIS in the real world represents a novel combination for robot-control systems, and provides distinct advantages, not only for AIS initialization, but also for evolutionary robotics. In the past, much evolutionary work has been carried out serially on physical robots, which requires a long time for convergence and puts the robot and its environment at risk of damage. For example, Floreano and Mondada (1996) adopt this approach and report a convergence time of ten days. More recent evolutionary experiments with physical robots, for example Marocca and Floreano (2002), Hornby *et al.* (2000), and Zykov *et al.* (2004) have

produced reliable and robust systems, but have not overcome the problems of potential damage and slow, impractical convergence times. Evolving in parallel with a number of robots, (for example Watson *et al.* 1999) reduces the time required, but can still be extremely prohibitive in terms of time and logistics. Simulated robots provide a definite advantage in terms of speed of convergence, but the trade-off is the huge difference between the simulated and real domains (Brooks, 1992).

Systems that employ an evolutionary training period (long-term leaning phase) and some form of lifelong adaptation (short-term learning phase) have been used to try to address the problem of domain differences, for example by Nehmzow (2002). However, the long-term learning phase in Nehmzow's work uses physical robots evolved in parallel, which means that the method is slow and restricted to multi-agent tasks. Floreano and Urzelai (2000) evolve an adaptable neural controller that transfers to different environments and platforms, but use a single physical robot for the long-term phase. Keymeulen *et al.* (1998) run their long-term and short-term learning phases simultaneously, as the physical robot maps its environment at the same time as carrying out its goal-seeking task, thus creating the simulated world. They report the rapid evolution of adaptable and fit controllers, but these results apply only to simple, structured environments where the robot can always detect the coloured target, and the obstacles are few. For example, they observe the development of obstacle avoidance in five minutes, but this applies to an environment with only one obstacle, and the results imply that the real robot was unable to avoid the obstacle prior to this. Furthermore, only eight different types of motion are possible in their system. Walker *et al.* (2006) use a GA in the simulated long-term phase and an evolutionary strategy (ES) on the physical robot. They note improved performance when the long-term phase is implemented, and remark that the ES provides continued adaptation to the environment, but they deal with only five or 21 behaviour parameters in the GA, and do not state the duration of the long-term phase.

The method described here aims to capitalize on the fast convergence speeds that a simulator can achieve, but will also address the domain compatibility issues by validating and, if necessary, modifying all simulation-derived behaviours in the real world. This will be achieved by transferring the behaviours to an adaptive AIS that runs on a real robot. The method is hence entirely practical for real world situations, in terms of delivering a short

training-period, safe starting-behaviours, and a fully-dynamic and adaptable system.

3 TEST ENVIRONMENT AND PROBLEM

The long-term phase requires accelerated simulations in order to produce the initial sets of antibodies as rapidly as possible. For this reason the Webots simulator (Michel, 2004) is selected as it is able to run simulations up to 600 times faster than real time, depending on computer power, graphics card, world design and the number and complexity of the robots used. The chosen robot is the e-puck (see Figure 1), since the Webots c++ environment natively supports it. It is a miniature mobile-robot equipped with a ring of eight noisy, nonlinear, infrared (IR) sensors that can detect the presence of objects up to a distance of about 0.1 m. It also has a small frontal camera that receives the raw RGB values of the images in its field-of-view. Blob-finding software is created to translate this data into groups of like-coloured pixels (blobs).

The test problem used here consists of a virtual e-puck that must navigate around a building with three rooms (see Figures 2 and 3) by tracking blue markers painted on the walls. These markers are intended to guide the robot through the doors, which close automatically once the robot has passed through. The course is completed once the robot has crossed the finish-line in the third room, and its performance is measured according to how quickly it can achieve this goal, and how many times it collides with the walls or obstacles placed in the rooms. Two different test environments are used; World 1 (see Figure 2) has fewer obstacles and no other robots. World 2 (see Figure 3) contains more obstacles, and there is also a dummy wandering-robot in each room.

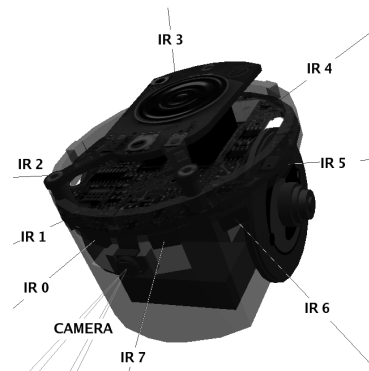


Figure 1: A simulated e-puck robot.

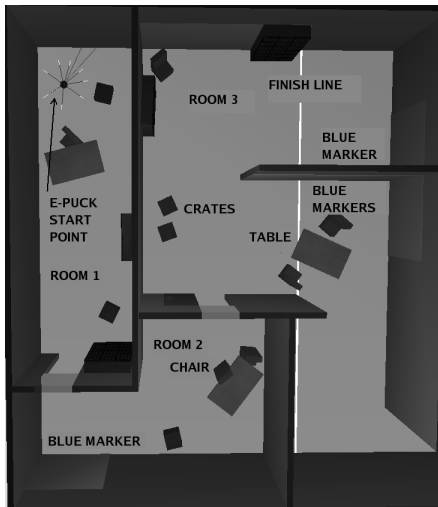


Figure 2: World 1 showing e-puck start point.

The simulations are run in fast mode (no graphics) with Webots version 5.1.10 using GNU/Linux 2.6.9 (CentOS distribution) with a Pentium 4 processor (clock speed 3.6 GHz). The graphics card used is an NVIDIA GeForce 7600GS, which affords average simulation speeds of approximately 200-times real-time for World 1 and 100-times real-time for World 2. The camera field-of-view is set at 0.3 radians, the pixel width and height at 15 and 3 pixels respectively and the speed unit for the wheels is set to 0.00683 radians/s.

4 SYSTEM ARCHITECTURE

4.1 Antigens and Antibodies

The antigens model the environmental information as perceived by the sensors. In this problem there are only two basic types of antigen, whether a door-marker is visible (a “marker” type) and whether an obstacle is near (an “obstacle” type), the latter taking priority over the former. An obstacle is detected if the IR sensor with the maximum reading I_{max} has value V_{max} equal to 250 or more. The IR sensors correspond to the quantity of reflected light, so higher readings mean closer obstacles. If no obstacles are detected then the perceived antigen is of type “marker” and there are two varieties, “marker seen” and “marker unseen”, depending on whether appropriate-coloured pixel-clusters have been recognized by the blob-finding software. If an obstacle is detected then the antigen is of type “obstacle”, i.e. the robot is no longer concerned with the status of the door-marker. The obstacle is classi-

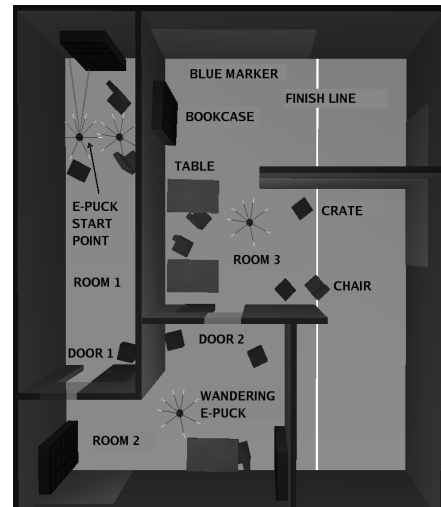


Figure 3: World 2 showing e-puck start-point, dummy-robot start-point and dummy-robot repositioning points.

fied in terms of both its distance from and its orientation toward the robot. The distance is “near” if V_{max} is between 250 (about 0.03 m) and 2400 (about 0.01 m), and “collision” if V_{max} is 2400 or more. The orientation is “right” if I_{max} is sensor 0, 1 or 2, “rear” if it is 3 or 4 and “left” if it is 5, 6 or 7 (see Figure 1). There are thus eight possible antigens, which are coded 0–7, see Table 1.

Table 1: System antigens.

Antigen Code	Antigen Type	Name
0	Marker	Marker unseen
1	Marker	Marker seen
2	Obstacle	Obstacle near right
3	Obstacle	Obstacle near rear
4	Obstacle	Obstacle near left
5	Obstacle	Collision right
6	Obstacle	Collision rear
7	Obstacle	Collision left

The behaviours that form the core of the antibodies are encoded using a structure that has the attributes, type T , speed S , frequency of turn F , angle of turn A , direction of turn D , frequency of right turn R_f , angle of right turn R_a , and cumulative reinforcement-learning score L . There are six types of behaviour; wandering using either a left or right turn, wandering using both left and right turns, turning forwards, turning on the spot, turning backwards, and tracking the door-markers. The fusion of these basic behaviour-types with a number of different attributes that can take many values means that millions of different behaviours are

possible. However, some behaviour types do not use a particular attribute and there are limits to the values that the attributes can take. These limits are carefully selected in order to strike a balance between reducing the size of the search space, which increases speed of convergence, and maintaining diversity, see Table 2.

4.2 System Structure

The control program uses the two-dimensional array of behaviours B_{ij} , $i = 0, \dots, x-1, j = 0, \dots, y-1$, where x is the number of robots in the population ($x \geq 5$) and y is the number of antigens, i.e. eight. When the program begins i is equal to zero, and the array is initialized to null. The infra-red sensors are read every 192 milliseconds and the camera is read every 384 milliseconds, but only if no obstacles are found, as this increases computational efficiency.

Once an antigen code is determined, a behaviour or antibody is created to combat it by randomly choosing a behaviour type and its attribute values. For example, the behaviour WANDER_SINGLE (605, 50, 90, LEFT, NULL, NULL) may be created. This behaviour consists of travelling forwards with a speed of 605 Speed Units/s, but turning left 50% of the time by reducing the speed of the left wheel by 90%. If the antigen code is 7 then the S , F , A , D , R_f and R_a attributes of B_{07} take the values 605, 50, 90, LEFT, NULL, NULL respectively. The action is executed and the sensor values are read again to determine the next antigen code. If the antigen has been encountered before, then the behaviour assigned previously is used, otherwise a new behaviour is created. The algorithm proceeds in this manner, creating new behaviours for antigens that have not been seen before and reusing the behaviours allotted to those that have.

However, the performance of the behaviours in dealing with the antigen they have been allocated to

is constantly assessed using reinforcement learning (see section 4.4), so that poorly-matched behaviours can be replaced with newly-created ones when the need arises. Behaviours are also replaced if the antigen has not changed in any 60-second period, as this most likely means that the robot has not undergone any translational movement. The cumulative reinforcement-score of the previously used behaviour L is adjusted after every sensor reading, and if it falls below the threshold value of -14 then replacement of the behaviour occurs. The control code also records the number of collisions c_i for each robot in the population.

A separate supervisor-program is responsible for returning the virtual robot back to its start-point once it has passed the finish-line, for opening and closing the doors as necessary, and for repositioning the wandering dummy-robot, so that it is always in the same room as the mission robot. Another of the supervisor's functions is to assess the time taken t_i to complete the task. Each robot is given 1250 seconds to reach the end-point; those that fail receive a 1000-second penalty if they did not pass through any doors. Reduced penalties of 750 or 500 seconds are awarded to failing robots that pass through one door or two doors respectively. When the whole population has completed the course, the relative-fitness μ_i of each individual is calculated. Since high values in terms of both t_i and c_i should yield a low relative-fitness, the following formula is used:

$$\mu_i = \frac{1}{(t_i + c_i) \sum_{k=0}^{x-1} (t_k + c_k)^{-1}}. \quad (1)$$

The five fittest robots in the population are selected, and their mean t_n , c_n and absolute-fitness f_n are calculated, where n represents the generation number, and $f_n = t_n + c_n$. In addition, the value of f_n is compared with that of the previous generation f_{n-1}

Table 2: System antibody types.

No.	Description	S Speed Units / s		F % of time		A % reduction in speed of one wheel		D Either left or right		R_f % of time		R_a % reduction in right wheel- speed	
		MIN	MAX	MIN	MAX	MIN	MAX	1	2	MIN	MAX	MIN	MAX
0	vander single	50	800	10	90	10	110	L	R	-	-	-	-
1	vander both	50	800	10	90	10	110	-	-	10	90	10	110
2	orward turn	50	800	-	-	20	200	L	R	-	-	-	-
3	tatic turn	50	800	-	-	100	100	L	R	-	-	-	-
4	everse turn	500	800	-	-	20	200	L	R	-	-	-	-
5	rack markers	50	800	-	-	0	30	-	-	-	-	-	-

to assess rate-of-convergence. The genetic algorithm is complete when any of the four conditions shown in Table 3 are reached. These are selected in order to achieve fast convergence, but also to maintain a high solution quality. Once convergence is achieved the attribute values representing the behaviours of the five fittest robots are saved for seeding the AIS system. If there is no convergence then the GA proceeds as described in section 4.3.

Table 3: Stopping criteria.

	Criteria - World 1	Criteria - World 2
1	$n > 0$ AND $t_n < 400$ AND $c_n < 60$ AND $ f_n - f_{n-1} < 0.1$	$n > 0$ AND $t_n < 600$ AND $c_n < 90$ AND $ f_n - f_{n-1} < 0.2$
2	$n > 30$	$n > 30$
3	$t_n < 225$ AND $c_n < 35$	$t_n < 400$ AND $c_n < 45$
4	$n > 15$ AND $ f_n - f_{n-1} < 0.1$	$n > 15$ AND $ f_n - f_{n-1} < 0.2$

Note that when adopting the scenario of five separate populations that never interbreed, the five robots that are assessed for convergence are the single fittest from each of the autonomous populations. In this case, convergence is dependent upon the single best t_n , c_n and f_n values. The final five robots that pass their behaviours to the AIS system are the single fittest from each population after convergence.

4.3 The Genetic Algorithm

Two different parent robots are selected through the roulette-wheel method and each of the x pairs interbreeds to create x child robots. This process is concerned with assigning behaviour attribute-values to each of the x new robots for each of the y antigens in the system. It can take the form of complete antibody replacement, attribute-value mutation, adoption of the attribute values of only one parent or crossover from both parents.

Complete antibody replacement occurs according to the prescribed mutation rate ϵ . Here, a completely new random behaviour is assigned to the child robot for the particular antigen, i.e. both the parent behaviours are ignored.

Crossover is used when there has been no complete replacement, and the method used depends on whether the parent behaviours are of the same type. If the types are different then the child adopts the complete set of attribute values of one parent only, which is selected at random. If the types are the same, then crossover can occur by taking the averages of the two parent values, by randomly

selecting a parent value, or by taking an equal number from each parent according to a number of set patterns. In these cases, the type of crossover is determined randomly with equal probability. The purpose behind this approach is to attempt to replicate nature, where the offspring of the same two parents may differ considerably each time they reproduce.

Mutation of an attribute value may also take place according to the mutation rate ϵ , provided that complete replacement has not already occurred. Here, the individual attribute-values (all except D) of a child robot may be increased or decreased by between 20% and 50%, but must remain within the prescribed limits shown in Table 2.

4.4 Reinforcement Learning

Reinforcement learning is used in order to accelerate the speed of the GA's convergence. It can be thought of as microcosmic short-term learning within the long-term learning cycle. The reinforcement works by comparing the current and previous antibody codes, see Table 4. Ten points are awarded for every positive change in the environment, and ten are deducted for each negative change. For example, 20 points are awarded if the antigen code changes from an "obstacle" type to "marker seen", because the robot has moved away from an obstacle as well as gaining or keeping sight of a door-marker.

Table 4: Reinforcement scores.

Antigen code		Reinforcement status (score)
Old	New	
0	0	Neutral (0)
1	0	Penalize - Lost sight of marker (-10)
2-7	0	Reward - Avoided obstacle (10)
0	1	Reward - Found marker (10)
1	1	Reward - Kept sight of marker (Score depends on orientation of marker with respect to robot)
2-7	1	Reward - Avoided obstacle and gained or kept sight of marker (20)
0	2-7	Neutral (0)
1	2-7	Neutral (0)
2-7	2-7	Reward or Penalize (Score depends on several factors)

In the case where the antigen code remains at 1 (a door-marker is kept in sight), the score awarded depends upon how the orientation of the marker has moved with respect to the robot. In addition, when an obstacle is detected both in the current and

previous iteration, then the score awarded depends upon several factors, including changes in the position of I_{max} and in the reading V_{max} , the current and previous distance-type (“collision” or “near”) and the tallies of consecutive “nears” and “collisions”.

5 EXPERIMENTAL PROCEDURES

5.1 General Procedures

The GA is run in Worlds 1 and 2 using single populations of 25, 40, and 50 robots, and using five autonomous populations of five, eight, and ten. A mutation rate ϵ of 5% is used throughout, as previous trials have shown that this provides a good compromise between fast convergence, high diversity and good solution-quality. Solution quality is measured as $q = (t + 8c)/2$, as this allows equal weighting for the number of collisions. For each scenario, ten repeats are performed and the means of the program execution time τ , solution quality q , and diversity in type Z_i and speed Z_s are recorded. The mean solution-quality is also noted when 240 repeats are performed in each world using a hand-designed controller. This shows how well the GA-derived solutions compare with an engineered system and provides an indication of problem difficulty. Two-tailed standard t-tests are conducted on the result sets, and differences are accepted as significant at the 99% level only.

In World 2 the stopping criteria is relaxed in order to improve convergence speed, see Table 3. This is necessary since there are more obstacles and moving robots to navigate around, which means that completion time is affected.

5.2 Measuring Diversity

Diversity is measured using the type T and the speed S attributes of each of the final antibodies passed to the AIS system, since these are the only action-controlling attributes that are common to all antibodies. The antibodies are arranged into y groups of five (y is the number of antigens) and each group is assessed by comparison of each member with the others, i.e. ten pair-wise comparisons are made in each group. A point is awarded for each comparison if the attribute values are different; if they are the same no points are awarded. For example, the set of behaviour types [1 3 4 4 1] has two pair-wise

comparisons with the same value, so eight points are given. Table 5 summarizes possible attribute-value combinations and the result of conducting the pair-wise comparisons on them.

Table 5: Diversity scores.

Attribute-value status	Points	Expected:	
		Frequency for T	Score for T
All five different	10	9.26	0.926
One repeat of two	9	46.30	4.167
Two repeats of two	8	23.15	1.852
One repeat of three	7	15.43	1.080
Two repeats, one of two, one of three	6	3.86	0.231
One repeat of four	4	1.93	0.077
All five the same	0	0.08	0.000
Total		100.00	8.333

The y individual diversity-scores for each of T and S are summed and divided by σy to yield a diversity score for each attribute. Here σ is the expected diversity-score for a large number of randomly-selected sets of five antibodies. This is approximately 8.333 for T (see Table 5) and 10.000 for S . It is lower for T since there are only six behaviours to select from, whereas the speed is selected from 751 possible values, so one would expect a random selection of five to yield a different value each time. The adjustment effectively means that a random selection yields a diversity of 1 for both S and T . The diversity calculation is given by:

$$Z = \frac{\sum_{i=1}^y z_i}{\sigma y}, \quad (2)$$

where Z represents the overall diversity-score and z represents the individual score awarded to each antigen.

6 RESULTS AND DISCUSSION

Table 6 presents mean τ , q , Z_i , and Z_s values in World 1, and Table 7 summarises the significant difference levels when comparing single and multiple populations. The schemes that are compared use the same number of robots, for example a single population of 25 is compared with five populations of five. In addition, the smallest and largest population sizes are compared for both single and multiple populations.

Table 6: World 1 means.

Pop. size	τ (s)	q	Z_t (%)	Z_s (%)
25	417	220	40	86
40	530	216	53	95
50	811	191	49	90
5 x 5	508	155	55	100
5 x 8	590	146	54	100
5 x 10	628	144	58	100

Table 7: World 1 significant differences.

Comparison		τ (s)	q	Z_t (%)	Z_s (%)
25	5 x 5	77.40	99.94	99.90	99.99
40	5 x 8	72.58	99.97	43.07	99.97
50	5 x 10	97.13	99.80	98.36	99.96
25	50	99.99	91.76	96.10	75.81
5 x 5	5 x 10	88.41	58.13	60.40	00.00

The tables show that there are no significant differences between controller run-times when comparing the single and multiple populations. Type diversity is consistently higher for the multiple populations, but only significantly higher when comparing a single population of 25 with five populations of five. However, solution quality and speed diversity are significantly better for the multiple populations in all three cases. Multiple populations always demonstrate a speed diversity of 100%, indicating that the final-selected genes are completely unrelated to each other, as expected. In contrast, single-population speed-diversity never reaches 100% as there are always repeated genes in the final-selected robots. Evidence from previous experiments with single populations of five, ten and 20 suggests that the level of gene duplication decreases as the single population size increases. This explains the lower Z_t and Z_s values for a population of 25 robots. However, when comparing the results from single populations of 25 with 50, the only significant difference is in the run-time, with 25-robot populations running much faster. This is intuitive, since fewer robots must complete the course for every generation. There are no significant differences when comparing five-robot and ten-robot multiple populations. Run-times may be comparable here because the course has to be completed fewer times for the smaller population, but it requires more generations for convergence since there seems to be a reduced probability of producing successful robots.

In all cases, mean type-diversity ratings never reach 100%, yet mean speed-diversity is always 100% in the multiple populations, which shows

there are no repeated genes. The reduced type-diversity ratings must therefore occur because the types are not randomly selected but chosen in a more intelligent way. The relatively small number of types (six) means that intelligent selection reduces the type diversity, whereas speed diversity is unaffected because there are many potentially-good speeds to choose from and convergence is rapid. It is likely that both intelligent selection and repeated genes decrease the type-diversity scores for the single populations, but in the multiple populations, the phenomenon is caused by intelligent selection only.

The hand-designed controller demonstrates a mean solution-quality of 336. (The scores from the 49 robots that failed to complete the course are not counted.) This is significantly worse than all of the multiple populations, but not significantly different to the single populations, although single-population quality scores are considerably better. The multiple populations may have an advantage over the single populations in terms of solution quality because, for each population, they require a fast time and few collisions for only one member in order to meet the convergence criteria. The single-population case demands good mean-scores from five robots.

Table 8 presents the significant difference levels when comparing the results from World 1 with those from World 2. There is a significant difference in run-time in every case, which is not surprising because the GAs in World 2 take, on average, 2.25 times as long to converge. This is partly due to the World 2 simulations running only half as fast as those in World 1 (because there are two robots to control) and partly because the problem is harder to solve. There are also significant differences in solution quality for 50-robot single populations and all the multiple populations, with World 2 producing the lower-quality solutions. (The 25-robot and 30-robot populations are almost significant.) This difference is due to the less-stringent convergence criteria in World 2. There are no significant differences in type diversity or speed diversity between the two worlds.

Tables 9 and 10 summarise the same data as Tables 6 and 7, but for World 2. When comparing single with multiple populations, the results reveal a similar pattern to World 1 in terms of run-time, type diversity and speed diversity, i.e., there are no significant differences between run-times, although they are slightly higher for the multiple populations. Type diversity is consistently better for the multiple populations, but only significantly higher when comparing a single population of 25 with a five-robot multiple population. Speed diversity is

consistently significantly higher for the multiple populations, with all multiple populations producing 100% diversity. However, unlike World 1, there are no significant differences in solution quality, although the figures for the multiple populations are better in each case.

Table 8: World 1 compared with World 2.

Pop. size	τ (s)	q	Z_t (%)	Z_s (%)
25	99.99	97.61	50.34	11.19
40	99.99	96.61	24.70	84.41
50	99.93	99.97	80.09	87.67
5 x 5	99.99	99.99	57.16	66.94
5 x 8	100.00	99.99	14.38	0.00
5 x 10	100.00	99.86	27.51	66.94

Table 9: World 2 means.

Pop. size	τ (s)	q	Z_t (%)	Z_s (%)
25	972	314	37	85
40	1292	266	51	89
50	1414	250	56	94
5 x 5	1211	258	58	100
5 x 8	1325	225	55	100
5 x 10	1498	208	57	100

Table 10: World 2 significant differences.

Comparison		τ (s)	q	Z_t (%)	Z_s (%)
25	5 x 5	88.47	84.51	99.96	99.63
40	5 x 8	20.91	94.09	61.19	99.28
50	5 x 10	40.78	97.31	18.34	99.87
25	50	98.79	90.17	99.50	93.43
5 x 5	5 x 10	94.36	97.97	22.16	00.00

When comparing the results from single populations of 25 robots with 50 robots, the only significant difference is in type diversity, with 50-robot populations producing more diverse sets of behaviour type. Since type diversity is also significantly higher in the five-robot multiple populations, this suggests there may be a threshold single population size, below which single populations are significantly less diverse in behaviour-type than their multiple-population counterparts. There are no significant differences when comparing five-robot and ten-robot multiple populations, although the higher solution-quality for ten robots almost reaches significance.

In World 2 the hand-designed controller produces a mean solution-quality of 623 (not counting the results from the 109 robots that failed

to complete the course). The performance is significantly worse than the GA-derived solutions from both the single and multiple populations in World 2.

7 CONCLUSIONS AND FUTURE WORK

This paper has described a GA method for intelligently seeding an idiotypic-AIS robot control-system, i.e. it has shown how to prepare an initial set of antibodies for each antigen in the environment. Experiments with static and dynamic worlds have produced solution-sets with significantly better mean solution-quality than a hand-designed controller, and the system has been able to deliver the starting antibodies within about ten minutes in the static world, and within about 25 minutes in the dynamic world. These are fast results compared with GAs that have used physical robots and reported convergence in terms of number of days rather than minutes. The method hence provides a practical training-period when considering real-world tasks.

The resulting antibody sets have also been tested for quality and diversity, and it has been shown that significantly higher antibody diversity can be obtained when a number of autonomous populations are used, rather than a single one. For sets of five populations, the mean diversity of antibody speed is 100%, and one can run the genetic algorithm without significantly increasing the convergence time or reducing solution quality. In fact, for simpler problems, multiple populations may help to improve solution quality. Results have also shown that the diversity ratings are not affected by the difficulty of the problem.

The potential of the method to create high behaviour-diversity augurs well for the next stage of the research, which is transference to a real robot running an AIS. This part of the work will investigate how the idiotypic-selection process should choose between the available solutions, and how antibodies should be replaced within the system when they have not proved useful. The work will also examine how closely the simulated-world needs to resemble the real-world in order that the initial solutions are of benefit.

REFERENCES

- Brooks, R. A., 1992. Artificial life and real robots. In F. J. Varela and P. Bourguin (Eds.) *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. Cambridge, MA: MIT Press.
- Farmer, J. D., Packard, N. H., Perelson, A. S., 1986. The immune system, adaptation, and machine learning. *Physica, D, Vol. 2, Issue 1-3*, 187-204.
- Floreano, D., Mondada, F., 1996. Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics, Vol. 26, No. 3*, 396-407
- Floreano, D., Urzelai, J., 2000. Evolutionary robots with on-line self-organization and behavioural fitness. *Neural Networks, Vol. 13*, 431-443.
- Hart, E., Ross, P., Webb, A., Lawson, A., 2003. A role for immunology in 'next generation' robot controllers. In *Proceedings of the 2nd International Conference on Artificial Immune Systems*, 46-56.
- Hornby, G., Takamura, S., Yokono, J., Hanagata, O., Yamamoto, T., Fujita, M., 2000. Evolving robust gaits with AIBO. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 3040-3045.
- Jerne, N. K., 1974. Towards a network theory of the immune system. *Ann. Immunol. (Inst Pasteur)*, 125 C, 373-389.
- Keymeulen, D., Iwata, M., Kuniyoshi, Y., Higuchi, T., 1998. Comparison between an off-line model-free and an on-line model-based evolution applied to a robotics navigation system using evolvable hardware. In *Artificial Life VI: Proceedings of the 6th International Conference on Artificial Life*, 199-209.
- Krautmacher, M., Dilger, W., 2004. AIS based robot navigation in a rescue scenario. *Lecture Notes Computer Science, 3239*, 106-118.
- Luh, G. C., Liu, W. W., 2004. Reactive immune network based mobile robot navigation. *Lecture Notes Computer Science, 3239*, 119-132.
- Marocco, D., Floreano, D., 2002. Active vision and feature selection in evolutionary behavioural systems. In *From Animals To Animats: Proceedings of the 7th International Conference on Simulation of Adaptive Behaviour (SAB-02)*, 247-255.
- Michel, O., 2004. Cyberbotics Ltd – WebotsTM: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems, Volume 1, Number 1*, 39-42.
- Michelan, R., Von Zuben, F. J., 2002. Decentralized control system for autonomous navigation based on an evolved artificial immune network. In *Proceedings of the 2002 Congress on Evolutionary Computation, Vol. 2*, 1021-1026.
- Nehmzow, U., 2002. Physically embedded genetic algorithm learning in multi-robot scenarios: the PEGA algorithm. In *Proceedings of the 2nd International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*.
- Vargas, P. A., de Castro, L. N., Michelan, R., 2003. An immune learning classifier network for autonomous navigation. *Lecture Notes Computer Science, 2787*, 69-80.
- Walker, J. H., Garrett, S. M., Wilson, M. S., 2006. The balance between initial training and lifelong adaptation in evolving robot controllers. *IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics, Vol. 36, No. 2*, 423-432.
- Watanabe, Y., Ishiguro, A., Shirai, Y., Uchikawa, Y., 1998a. Emergent construction of behavior arbitration mechanism based on the immune system. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC)*, 481-486.
- Watanabe, Y., Kondo, T., Ishiguro, A., Shirai, Y., Uchikawa, Y., 1998b. Evolutionary construction of an immune network-based behavior arbitration mechanism for autonomous mobile robots. *Electrical Engineering in Japan, Vol. 123, No. 3*, 1-10.
- Watson, R. A., Ficici, S. G., Pollack, J. B., 1999. Embodied evolution: A response to challenges in evolutionary robotics. In J. L. Wyatt and J. Demiris (Eds.) *Proceedings of the Eighth European Workshop on Learning Robots*, 14-22.
- Whitbrook, A. M., Aickelin, U., Garibaldi, J. M., 2007. Idiotypic Immune Networks in Mobile Robot Control. *IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics, Vol. 37, No. 6*, 1581-1598.
- Zykov, V., Bongard, J., Lipson, H., 2004. Evolving dynamic gaits on a physical robot. In *Proceedings of The Genetic and Evolutionary Computation Conference (GECCO), Late Breaking Papers*.

DIAGNOSIS OF DISCRETE EVENT SYSTEMS WITH PETRI NETS AND CODING THEORY

Dimitri Lefebvre

GREAH – University Le Havre, France

dimitri.lefebvre@univ-lehavre.fr

Keywords: Diagnosis, discrete event systems, Petri net models, events estimation.

Abstract: Event sequences estimation is an important issue for fault diagnosis of DES, so far as fault events cannot be directly measured. This work is about event sequences estimation with Petri net models. Events are assumed to be represented with transitions and firing sequences are estimated from measurements of the marking variation. Estimation with and without measurement errors are discussed in n – dimensional vector space over alphabet $Z_3 = \{-1, 0, 1\}$. Sufficient conditions and estimation algorithms are provided. Performance is evaluated and the efficiency of the approach is illustrated on two examples from manufacturing engineering.

1 INTRODUCTION

Modern technological processes include complex and large-scale systems, where faults in a single component have major effects on the availability and performances of the system as a whole. For example manufacturing systems consists of many different machines, robots and transportation tools all of which have to correctly satisfy their purpose in order to ensure and fulfil global objectives. In this context, a failure is any event that changes the behaviour of the system such that it does no longer satisfy its purpose (Rausand et al., 2004). Faults can be due to internal causes as to external ones, and are often classified into three subclasses: plant faults that change the dynamical input – output properties of the system, sensor faults that results in substantial errors during sensors reading, and actuator faults when the influence of the controller to the plant is disturbed. In order to limit the effects of the faults on the system, diagnosis is used to detect and isolate the failures. Diagnosis includes distinct stages: the fault detection decides whether or not a failure event has occurred; the fault isolation find the component that is faulty; the fault identification identifies the fault and estimates also its magnitude. Model-based and data-based methods have been investigated for diagnosis (Blanke et al., 2003).

The motivations for the diagnosis of discrete event system (DES) are obvious as long as DES occur naturally in the engineering practice. Many actuators like switches, valves and so on, only jump between discrete states. Binary signals are mainly

used with numerical systems and logical values “true” and “false” are often used as input and output signals. Alarm sensors that indicate that a physical quantity exceeds a prescribed bound are typical systems with only two logical states. Moreover, in several systems also the internal state is discrete valued. As an example, robot encoders are discrete valued even if the number of discrete state is large enough to produce smooth trajectories. At last, one must keep in mind that a given dynamical system can always be considered as a DES system or as a continuous variable system according to the purpose of the investigation. As long as supervision problems are considered, a rather broad view on the system behaviour can be adopted that is based on discrete signals. On the contrary, if signals have to remain in a narrow tolerance band, the following approaches do no longer fit and one has to adopt a continuous point of view (Blanke et al., 2003).

The behaviour of DES is described by sequences of input and output events. In contrast to the continuous systems only abrupt changes of the signal values are considered with DES. In that case, the problem has been originally investigated with observation methods for automata developed in connection with the supervisory control theory (Ramadge et al., 1987). Concerning model-based methods automata (Sampath et al., 1995) or Petri nets (Ushio et al. 1998) models can be used. This article focus on diagnosis of DES modelled with Petri nets (PN) where failures are represented with some particular transitions. The problem is to detect and isolate the firing of the failure transitions in a

given firing sequence. The firings of the failure transitions are assumed to be unobservable and must be estimated according to complete or partial marking measurements that are eventually disturbed by measurement errors. As a consequence a method based on coding theory is proved to be suitable for sensor faults diagnosis. The article is divided into six sections. Section two is about Petri nets states. Section three states the diagnosis problem for DES and is about the usual state space methods for PN. Section four details the event estimation with coding theory that can be combined with state space approach. Both methods are presented in a framework in the conclusion.

2 ORDINARY PETRI NETS

An ordinary PN with n places and q transitions is defined as $\langle P, T, \text{Pre}, \text{Post} \rangle$ where $P = \{P_i\}$ is a non-empty finite set of n places, $T = \{T_j\}$ is a non-empty finite set of q transitions, such that $P \cap T = \emptyset$. $\text{Pre}: P \times T \rightarrow \{0, 1\}$ is the pre-incidence application and $W_{\text{PR}} = (w_{ij}^{\text{PR}}) \in \{0, 1\}^n \times q$ with $w_{ij}^{\text{PR}} = \text{Pre}(P_i, T_j)$ is the pre-incidence matrix. $\text{Post}: P \times T \rightarrow \{0, 1\}$ is the post-incidence application and $W_{\text{PO}} = (w_{ij}^{\text{PO}}) \in \{0, 1\}^n \times q$ with $w_{ij}^{\text{PO}} = \text{Post}(P_i, T_j)$ is the post-incidence matrix. The PN incidence matrix W is defined as $W = W_{\text{PO}} - W_{\text{PR}} \in Z_3^{n \times q}$ with $Z_3 \in \{-1, 0, 1\}$ and w_i stands for the i^{th} column of W (Askin et al., 1993; Cassandras et al., 1999; David et al., 1992). $M = (m_i) \in (Z^+)^n$ is defined as the marking vector and $M_1 \in (Z^+)^n$ as the initial marking vector, with Z^+ the set of non negative integer numbers. A firing sequence $\sigma = T_1 T_2 \dots T_k$ is defined as an ordered series of transitions that are successively fired from marking M to marking M' (i.e. $M[\sigma > M']$) such that equation (1) is satisfied:

$$\sigma: M \xrightarrow{T_1} M_1 \xrightarrow{T_2} M_2 \rightarrow \dots \xrightarrow{T_k} M' \quad (1)$$

A sequence σ can be represented by its characteristic vector (i.e. Parikh vector) $X = (x_j) \in (Z^+)^q$ where x_j stands for the number of times T_j has occurred in sequence σ (David et al., 1992). Marking M' resulting from marking M with the execution of sequence σ is given by (2) where X is the characteristic vector for sequence σ :

$$\Delta M = M' - M = W.X \quad (2)$$

The reachability graph $R(\text{PN}, M_1)$ is the set of markings M such that a firing sequence σ exists from M_1 to M . A sequence σ is said to be executable

for marking M_1 if there exists a couple of markings $(M, M') \in R(\text{PN}, M_1)$ such that $M[\sigma > M']$.

3 DIAGNOSABILITY AND DIAGNOSER DESIGN FOR DES

3.1 Problem Statement

In the context of diagnosis, it is commonly assumed that no inspection of the process is possible. As a consequence the diagnosis is only based on available measurement data. Basically, the diagnosis problem for a dynamical system with input u , output y and subject to some faults f , is to detect and isolate the faults from a given sequence of input – output couples (U, Y) with:

$$U = (u(0), u(1), \dots, u(k)) \quad (3)$$

$$Y = (y(0), y(1), \dots, y(k))$$

where k stands for time $t = k.\Delta t$, and Δt represents the sampling period of sensors. The main issues are (1) to decide the diagnosability of the faults; (2) to detect, isolate and identify the faults that are diagnosable. In case of model - based diagnosis, the input – output couples (U, Y) are usually compared with the behaviour of a reference model. Fault indicators like residuals are worked out from this comparison. It is often convenient to separate actuator, system and sensor faults.

As long as DES are considered the inputs and faults are usually considered as events and the outputs are related to the states of the DES. A reference model (automata, finite state machines, Petri nets, and so on) can be used for diagnosis purpose and sequences of estimated outputs obtained thanks to the model are compared with the measured outputs of the system. Indicators of the faults result from this comparison. According to the traces generated by the system, faults are :

- (1) strongly diagnosable if they result in immediate abnormal behaviours (no intermediate event is required for diagnosis);
- (2) weakly diagnosable if they result in abnormal behaviours after a finite number of intermediate events;
- (3) non diagnosable if no abnormal behaviour occurs whatever the future evolution of the system.

Let us notice that the notion of strong or weak diagnosability for DES is related to the question of persistent excitation in temporal systems.

The figure 1 is an example of diagnosis with finite state machine. The system has 5 states $\{A, B, C, D, E\}$, 4 outputs $\{1, 2, 3, 4\}$, 5 inputs $\{a, b, c, f_1, f_2\}$ (3 normal events $\{a, b, c\}$ and 2 fault events $\{f_1, f_2\}$). The reference model (full lines only) and the system (full and dashed lines) evolve according to the figure 1. Diagnosability analysis and diagnosers design result from the simulation with automata in figure 1.

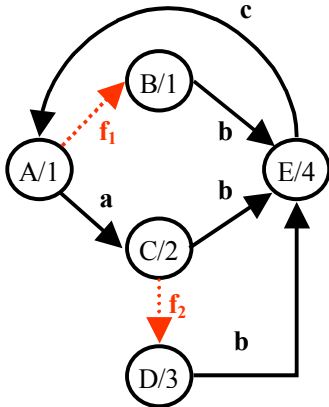


Figure 1: Example of diagnosis with finite state machine.

If the state of the system is measured, then the faults f_1 and f_2 are both strongly diagnosable as long as the fault events lead to an immediate difference between system state (S) and estimated one (S_{est}) (table 1, grey cells). If only the output is measured then the fault f_2 is strongly diagnosable but the fault f_1 is weakly diagnosable in the sense that intermediate event “b” must occur so that the system output (O) and estimated output (O_{est}) become different. If state “E” results in output “1” instead of “4” then fault f_2 is non diagnosable.

3.2 Diagnosis with Petri Nets

The previous approach can be applied to Petri net models with finite reachability graph to prove the diagnosability of the faults and to design diagnosers based on Petri net models. The basis idea is to investigate the indeterminate cycles in partial expansion of the reachability graph (Ushio et al., 1998). The considered PN are live (i.e. for any $T_j \in T$, and for all $M \in R(PN, M_j)$ there exists a sequence σ executable from M that includes transition T_j) and safe (i.e. for all $M \in R(PN, M_j)$, $M \in \{0, 1\}^n$). Some places are assumed to be observable and other not, and transitions, that are associated with events, are usually assumed to be unobservable. A cycle is called “determined” if it contains at least one observable state that results with no ambiguity from

a normal firing sequence, or from a firing sequence with a fault. The fault is diagnosable if and only if there is no indeterminate cycle in partial expansion of the reachability graph that correspond to the observable part of the system. For a diagnosable fault, the detection and isolation can be obtained according to the finite state machine that corresponds to partial expansion of the reachability graph. Let us notice that the method is different from the diagnosis with finite state machines in the sense that knowledge of inputs is not required and that definition of outputs is restricted to marking projection.

Let consider the system PN1 in figure 2 as an example. The reachability graph of PN1 is the finite state machine of figure 1. If the set of observable places is given by $P_{O1} = \{P_1, P_4, P_5\}$, the observable part of the labelled reachability graph $R(PN1, \{T_1\}, (1, 0, 0, 0, 0)^T, P_{O1})$ is worked out as in figure 3a. This diagnoser has an indetermined cycle so the system is not diagnosable (figure 3a, left cycle). If $P_{O2} = \{P_1, P_3\}$, the observable part of the labelled reachability graph $R(PN1, \{T_1\}, (1, 0, 0, 0, 0)^T, P_{O2})$ is worked out as in figure 3b. This diagnoser has no indetermined cycle so the system is diagnosable.

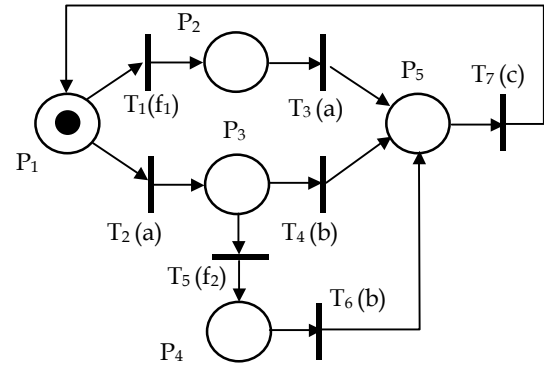
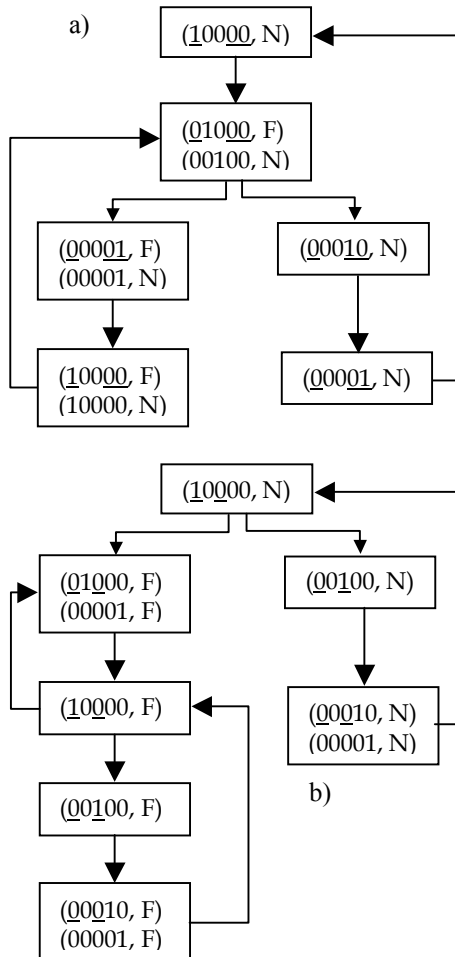


Figure 2: Example PN1 of Petri net.

Let us mention that other approaches have been developed for diagnosis based on event detectability (Ramirez – Trevino et al., 2007) and structural properties (Lefebvre et al., 2007). All above mentioned approaches require complete or partial measurements of the marking vector. Thus, they are sensitive to measurement errors. As a consequence, it is important to detect and eventually correct the errors that disturb the measurements of marking variation in order to obtain an exact estimation of the occurrence of events. The next section concerns events estimation and can be introduced as a diagnosis method for sensor faults.

Table 1: Example of input sequence (I), state sequence (S), output sequence (O), estimated state sequence (S_{est}) and estimated output sequence (O_{est}) for the final state machine in figure 1

I	a	b	c	a	f_2	b	C	f_1	b	c	a	b	...
S	C	E	A	C	D	E	A	B	E	A	C	E	...
O	2	4	1	2	3	4	1	1	4	1	2	4	...
S_{est}	C	E	A	C	C	E	A	A	A	A	C	E	...
O_{est}	2	4	1	2	2	4	1	1	1	1	2	4	...


 Figure 3 : Two partial expansions of the reachability graph for PN1 a) $R(PN1, \{T_1\}, (1, 0, 0, 0, 0)^T, P_{O1})$; b) $R(PN1, \{T_1\}, (1, 0, 0, 0, 0)^T, P_{O2})$.

4 SENSOR FAULTS DIAGNOSIS BASED ON CODING THEORY

Event sequences estimation is an important issue for fault diagnosis of DES, so far as fault events cannot be directly measured. This section is about event sequences estimation with PN models. Events are assumed to be represented with transitions and firing

sequences are estimated from measurements of the marking variation. Estimation with and without measurement errors can be discussed in n – dimensional vector space over alphabet $Z_3 = \{-1, 0, 1\}$ (Lefebvre, 2008). The basis idea to correct measurement errors by projecting measurements in orthogonal subspace of $\text{Vect}(W)$ where $\text{Vect}(W)$ stands for the subspace generated by the columns of W . This method is inspired from linear coding theory (Van Lint, 1999) and extends the results presented for continuous PN in (Lefebvre et al., 2001).

Our contribution can be compared to another method that incorporates redundancy into Petri nets to detect and identify faults (Li et al., 2004; Wu et al., 2002, 2005) and uses algebraic decoding techniques as the Berlekamp – Massey decoding (Berlekamp, 1984). The marking of the original PN is embedded into a redundant one and the diagnosis of faults is performed by mean of linear parity checks. In comparison with the method developed in (Wu et al., 2005), our approach does not require additive places, but is less efficient for faults correction.

Let us assume that measurement $\Delta \hat{M}$ of marking variation $\Delta M \in (Z_3)^n$ may be affected by additive error vector $E \in (Z_3)^n$: $\Delta \hat{M} = \Delta M + E$ where “+” stand for the sum endowed over Z_3 . Error vector will be characterized according to the Hamming distance $d(W)$ of the considered PN that is defined with the Hamming distance of the columns of incidence matrix :

$$d(W) = \min\{\min\{d(w_i, w_j), i \neq j\}, \min\{d_0(w_i)\}\} \quad (4)$$

where $d(w_i, w_j)$ stands for the Hamming distance between columns w_i and w_j of matrix W and $d_0(w_i) = d(w_i, 0)$ stands for the weight of vector w_i .

It is assumed that error vector E verifies the following conditions:

- $\Pr(d_0(E) = 0) > \Pr(d_0(E) = 1) > \dots > \Pr(d_0(E) = n)$ where $\Pr(d_0(E) = i)$ is the probability that weight of E equals i ;
- An error in position i does not influence other positions;
- A symbol in error can be each of the remaining symbols with equal probability.

A short estimation algorithm easy to use and to implement when state measurement is complete (i.e. all entries of $\Delta\hat{M}$ are measured), and error free (i.e. measurement equals actual marking variation ΔM), is based on the comparison of measurement with respect to columns of W and zero vector (this corresponds to the condition of event-detectability in case that all places are observable). When this measurement equals a single column of W , the algorithm decides that the corresponding transition fired. When it equals the zero vector, the algorithm decides that no transition fired.

When measurement is perturbed by non zero error E , two problems must be mentioned :

- a) A miss estimation may occur when $\Delta\hat{M}$ is non zero and different from any columns of W . The estimation algorithm is not able to decide if a transition fired or not and which transition fired. As consequence the algorithm does not give any decision.
- b) A wrong estimation may occur when $\Delta\hat{M}$ does not equal actual marking variation ΔM but equals zero vector or another column of W . The estimation algorithm decides if a transition fired or not and which transition fired, but the decision is wrong due to the measurement error.

To overcome these difficulties and to improve estimation, diagnosis can be reformulated as a linear problem in $((Z_3)^n, +, *)$, with the Smith transformation of W , where “+” and “*” stand for the sum and product endowed over Z_3 . The Smith transformation results from elementary operations (i.e. row or column permutations, linear combinations and external products), summed up in matrices $P \in (Z_3)^{n \times n}$ and $Q \in (Z_3)^{q \times q}$ such that:

$$P * W * Q = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix} \quad (5)$$

I_r is the identity matrix of dimension $r \times r$, and r is the rank of matrix W . The Smith transformation leads to reduced incidence matrix W' :

$$W' = (I_r \ 0) * Q^{-1} = (I_r \ 0) * P * W \\ = F * W \in (Z_3)^{r \times q} \quad (6)$$

Necessary and sufficient conditions for firing sequences estimation can be stated when measurement is error free and basic assumption in section 2.b is satisfied : columns of incidence matrix W' defined by equation (6) are distinct and non zero (Lefebvre, 2008). In case of measurement errors that

satisfy assumptions a to c, sufficient conditions inspired from coding theory can be stated. These conditions are based on Hamming distance, cosets investigation, parity check matrices, and syndromes (Van Lint, 1999). Cosets characterise the structure of $(Z_3)^n$ according to the sum and product over Z_3 (the coset $C(u)$ of u is defined as $C(u) = \{x \in (Z_3)^n \text{ such that } x = u + y \text{ with } y \in \text{Vect}(W)\}$, for any vector $u \in (Z_3)^n$). Parity check matrices are introduced to work out syndromes that can be considered as the signatures of the faults in $(Z_3)^n$. Two conditions for firing sequences estimation are proposed (Lefebvre, 2008):

- a) Columns of incidence matrix W are distinct, non zero and errors E that disturb satisfy $d_0(E) \leq (d(W) - 1) / 2$ (i.e. the number of disturbed entries of measurement is no larger than $(d(W) - 1) / 2$).
- b) Columns of reduced incidence matrix W' are distinct and non zero, and considered errors E belong to distinct cosets different from $C(0)$.

Moreover, the use of the Smith transformation of incidence matrix is also helpful to define the parity check matrix $H^T = (0 \ I_{n-r}) * P \in (Z_3)^{(n-r) \times n}$, and to work out the syndrome of marking variation measurements $S(\Delta\hat{M}) = H^T * \Delta\hat{M}$ and to compare it with the syndrome of errors $S(E) = H^T * E$. As a consequence the method leads to a less complex and more efficient diagnosis algorithm (algorithm b) in comparison with usual method based on Hamming distance (algorithm a) (Lefebvre, 2008).

Algorithm a

1. For each time k , measure $\hat{M}(k)$ the current state of DES
2. Compute $\Delta\hat{M}(k) = \hat{M}(k) - \hat{M}(k-1)$
3. Compute weight $d_0(\Delta\hat{M}(k))$. If $d_0(\Delta\hat{M}(k)) \leq (d(W) - 1) / 2$, then no event occurs between two consecutive state measurements. Go to step 6.
4. Compute Hamming distance $d(\Delta\hat{M}(k), w_j)$ for each column w_j of W . If $d(\Delta\hat{M}(k), w_j) \leq (d(W) - 1) / 2$ then T_j fired. Go to step 6.
5. If for all $j = 1, \dots, q$, $d(\Delta\hat{M}(k), w_j) > (d(W) - 1) / 2$ then measurement is too much disturbed by errors (i.e. $d_0(E) > (d(W) - 1) / 2$) and no decision is provided (i.e. a miss estimation occurs).
6. Wait until time $k + 1$. Go to step 1.

Algorithm b

1. For each time k , measure $\hat{M}(k)$ the current state of DES
2. Compute $\Delta\hat{M}(k) = \hat{M}(k) - \hat{M}(k-1)$
3. Compute $H^T * \Delta\hat{M}(k)$. If $H^T * \Delta\hat{M}(k) = 0$ then measurement is not disturbed by errors: $\Delta M(k) = \Delta\hat{M}(k)$. Go to step 5.
4. If syndrome $H^T * \Delta\hat{M}(k) \neq 0$, compute coset leader $E(k)$ and $\Delta M(k) = \Delta\hat{M}(k) - E(k)$. Go to step 5.
5. Compute $\Delta M'(k) = F * \Delta M(k)$.
6. If $\Delta M'(k) = 0$ then no event occurs between 2 consecutive state measurements. Go to step 8.
7. If $\Delta M'(k) = w_j$ then T_j fired. Go to step 8.
8. Wait until time $k + 1$. Go to step 1.

The correction capacity (i.e. number of error vectors that are corrected) of algorithm a is given by equation (7):

$$\sum_{i=1}^{(d(W)-1)/2} 2^i \cdot \left(\frac{n!}{i!(n-i)!} \right) \quad (7)$$

and its complexity results from $2n \cdot (q+1)$ scalar comparisons or operations whereas correction capacity of algorithm b equals $3^{n-r} - 1$, and its complexity results from $r \cdot (2n+q) + (n-r) \cdot (2n-1+3^{n-r})$ scalar comparisons or operations (Lefebvre, 2008). As a conclusion, algorithm b (with matrix W') is more efficient than algorithm a (with matrix W) for PN with small rank r in comparison with the number of places, and for PN with few transitions in comparison with the number of places. Algorithm b will be also preferred for PN with a small Hamming distance. This result is not surprising as long as the correction capacity of algorithm a is directly related to the value of Hamming distance. The determination of reduced incidence matrix does not increase the complexity of algorithm b as long as this determination is work out off-line.

5 APPLICATION

Algebraic methods have been used for the diagnosis of manufacturing and robotic systems. In order to illustrate algebraic methods, let us consider PN2 in figure 4 with incidence matrix (8), that is a simplified model of a manufacturing workshop (Silva et al., 2004). The final product is composed of two different parts that are processed in two separate machines modelled by transitions T_1 and T_2 , and stored in buffers P_4 and P_6 , respectively. Then, they are assembled by the machine T_3 , and processed by

T_4 and T_5 . During the processing, several tools are needed, modelled by places P_3 , P_5 and P_7 .

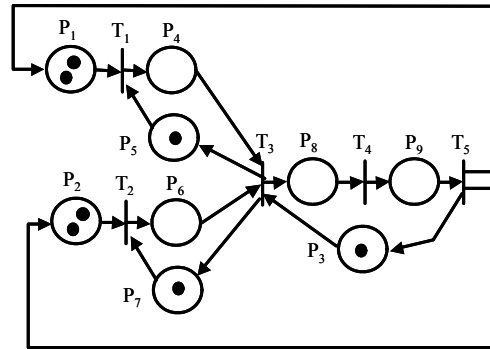


Figure 4: Model PN2 of a manufacturing system.

$$W = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \quad (8)$$

PN2 has $n = 9$ places, $q = 5$ transitions, is of rank $r = 4$ and incidence matrix W has a Hamming distance $d = 2$. Matrices F and H^T , worked out as in section 4, are given according to equations (9) and (10):

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (9)$$

$$H^T = \begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \quad (10)$$

PN2 has 243 cosets and each coset has 81 vectors. The table 2 gives the relationships between syndromes and coset leaders. Let us notice that the two last syndromes correspond to two different coset leaders. As a consequence not all errors of weight 1 will be corrected by algorithms a and b (errors $(0\ 0\ 0\ 0\ 0\ 0\ 1\ 0)^T$ and $(0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1)^T$ cannot be separated as errors $(0\ 0\ 0\ 0\ 0\ 0\ 0\ -1\ 0)^T$ and $(0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ -1)^T$).

Table 2: Correspondence between syndromes and coset leaders for PN2.

Syndromes	Errors of weight 1	Syndromes	Errors of weight 1
$(-1\ 0\ 0\ 1\ 0)^T$	$(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$	$(1\ 0\ 0\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0)^T$
$(1\ 0\ 0\ -1\ 0)^T$	$(-1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$	$(-1\ 0\ 0\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ -1\ 0\ 0\ 0\ 0)^T$
$(0\ 1\ -1\ 0\ 0)^T$	$(0\ 1\ 0\ 0\ 0\ 0\ 0\ 0)^T$	$(0\ 1\ 0\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0)^T$
$(0\ -1\ 1\ 0\ 0)^T$	$(0\ -1\ 0\ 0\ 0\ 0\ 0\ 0)^T$	$(0\ -1\ 0\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ 0\ -1\ 0\ 0\ 0)^T$
$(1\ -1\ 1\ -1\ 1)^T$	$(0\ 0\ 1\ 0\ 0\ 0\ 0\ 0)^T$	$(0\ 0\ 1\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0)^T$
$(-1\ 1\ -1\ 1\ -1)^T$	$(0\ 0\ -1\ 0\ 0\ 0\ 0\ 0)^T$	$(0\ 0\ -1\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ 0\ 0\ -1\ 0\ 0)^T$
$(0\ 0\ 0\ 1\ 0)^T$	$(0\ 0\ 0\ 1\ 0\ 0\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ 1)^T$	$(0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0)^T$ $(0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1)^T$
$(0\ 0\ 0\ -1\ 0)^T$	$(0\ 0\ 0\ -1\ 0\ 0\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ -1)^T$	$(0\ 0\ 0\ 0\ 0\ 0\ 0\ -1\ 0)^T$ $(0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ -1)^T$

Simulations for on – line estimation of the transitions firing are provided with figure 5. In these simulations, a measurement error ratio of 0.1 is supposed to be associated to each place (i.e. a probability of 0.1 that the marking variation of each place is biased). Transitions are assumed to fire with stochastic firing periods (exponential distribution) of mean value equal to 1 TU. All simulations indicate that complexity of algorithm b is not a limitation for real time applications. For the example PN2, the total CPU time for algorithm b is less than 5 TU for a simulation of 100 TU with a sampling period of 0.1 TU. This means that the average duration for each cycle of algorithm is approximatively 20 times less than the sampling period. The miss estimation rate for b is about 32% in comparison with a that has a rate of 60% and the wrong estimation rate is about 8% for b in comparison with a that has a rate less than 1%. Let us mention that the large number of miss estimation (even if measurement is unbiased) is due to the small Hamming distance of W ($d = 2$). For this reason numerous unbiased measurements of the marking variation are considered as suspicious and not used for estimation.

6 CONCLUSIONS

The investigation of diagnosis methods for discrete event systems shows that Petri nets is efficient not only to model the considered systems but also to support the diagnosis methods. Several approaches can be used in order to check diagnosability, to select sensors and to work out diagnosers. As a conclusion it is important to notice the great effort,

observed this last years to develop and improve diagnosis methods for DES. The use of the coding theory plays an important role in that development. As long as it is suitable to detect and correct measurement errors in the marking error variation. The main drawback is the strong dependence of the method to the algebraic properties of the incidence matrix.

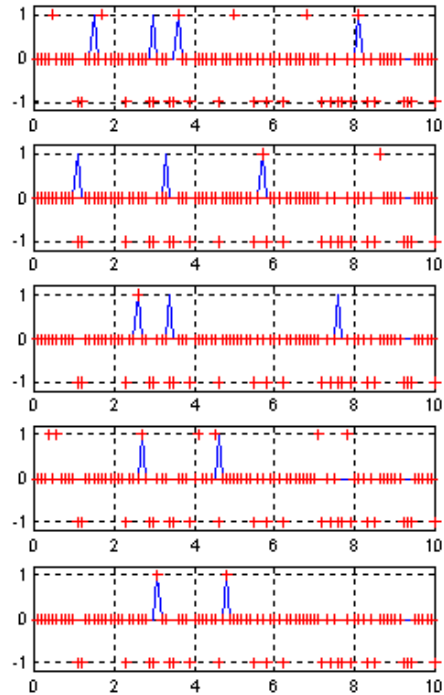


Figure 5: On – line firings estimation with algorithm b for the transitions T_1 to T_5 of PN2 (number of firings, full line: correct value; cross: estimated value; estimated value = -1 means miss estimation) in function of time (TU).

The method can be improved by incorporating additive places into Petri nets models. Taken into account the past sequence of events is another perspective to improve the efficiency of the method. But, the main challenge is, from our point of view, to take advantages from many important contributions that have been proposed for continuous systems. To build a bridge from continuous variable systems to DES theories remains one of the most promising issues for the next years.

REFERENCES

- Askin R.G., Standridge C. R. (1993). Modeling and analysis of Petri nets, John Wiley and sons Inc.
- Berlekamp R.E. (1984). Algebraic coding theory, Laguna Hills, CA, Aegean Park.
- Blanke M., Kinnaert M., Lunze J., Staroswiecki M. (2003). Diagnosis and fault tolerant control, Springer Verlag, New York.
- Cassandras C.G., Lafortune S. (1999). Introduction to discrete event systems, Kluwer Academic Pub.
- David R., Alla H. (1992). Petri nets and grafcet – tools for modelling discrete events systems, Prentice Hall, London.
- Lefebvre D., El Moudni A. (2001). Firing and enabling sequences estimation for timed Petri nets, Trans. IEEE - SMCA, vol. 31, no.3, pp. 153- 162.
- Lefebvre D., Delherm C. (2007). Fault detection and isolation of discrete event systems with Petri net models, Trans. IEEE – TASE, vol. 4, no. 1, pp. 114 – 118.
- Lefebvre D. (2008). Firing sequences estimation in vector space over Z_3 for ordinary Petri nets, , accepted for publication in Trans. IEEE – SMCA.
- Li L., Hadjicostis C. N. Sreenivas R. S. (2004). Fault Detection and Identification in Petri Net Controllers, Proc. IEEE-CDC04, pp. 5248 – 5253, Atlantis, Paradise Island, Bahamas.
- Ramirez-Trevino A., Ruiz-Bletran E., Rivera-Rangel I., Lopez-Mellado E. (2007). Online Fault Diagnosis of Discrete Event Systems. A Petri Net-Based Approach, Trans. IEEE – TASE, vol. 4, no. 1, pp. 31-39.
- Rausand M., Hoyland A. (2004). System reliability theory : models, statistical methods, and applications, Wiley, Hoboken, New Jersey.
- Ren H., Mi Z. (2006). Power system fault diagnosis modeling techniques based on encoded Petri nets, Proc. IEEE Power Engineering Society General Meeting.
- Sampath M., Sengupta R., Lafortune S., Sinnamohideen K., Teneketzis D. (1995). Diagnosibility of discrete event systems, Trans. IEEE-TAC, vol. 40, no.9, pp. 1555- 1575.
- Silva M., Recalde L. (2004). On fluidification of Petri Nets: from discrete to hybrid and continuous models, Annual Reviews in Control, vol. 28, no. 2, pp. 253-266.
- Ushio T., Onishi I., Okuda K., (1998). Fault detection based on Petri net models with faulty behaviours, Proc. IEEE – SMC98, pp 113-118.
- Van Lint J.H. (1999). Introduction to Coding Theory, Graduate Texts in Mathematics, vol. 86, Springer Verlag.
- Wu Y., Hadjicostis N. (2002). Non-concurrent fault identification in discrete event systems using encoded Petri net states, Proc. IEEE – CDC02, vol. 4, pp4018-4023.
- Wu Y., Hadjicostis N. (2005). Algebraic approaches for fault identification in discrete event systems, Trans. IEEE - TAC, vol. 50, no. 12, pp. 2048 – 2053.

DYNAMIC-BASED SIMULATION FOR HUMANOID ROBOT WALKING USING WALKING SUPPORT SYSTEM

Aiman Musa M. Omer, Yu Ogura, Hideki Kondo
Graduate School of Science and Engineering, Waseda University, Tokyo, Japan
aiman@fuji.waseda.jp

Hun-ok Lim
Department of Mechanical Engineering, Kanagawa University, Yokohama, Japan
holim@ieee.org

Atsuo Takanishi
Department of Mechanical Engineering/ Humanoid Robotics Institute, Waseda University, Tokyo, Japan
takanisi@waseda.jp

Keywords: Dynamic Simulation, Humanoid Robot, Biped robot, Walking Assist Machine.

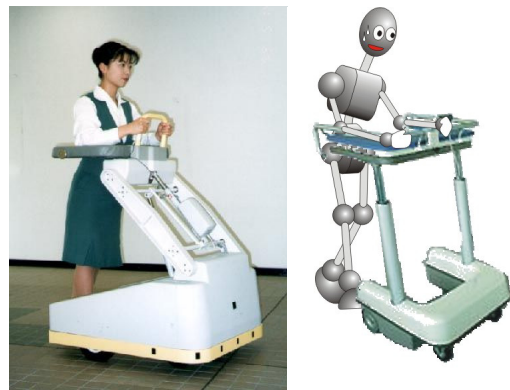
Abstract: A new humanoid bipedal robot WABIAN-2R was developed to simulate human motion. WABIAN-2R is able to perform similar human-like walking motion. Moreover, the robot is able to perform walking motions with a passive walk-assist machine. However, walking with an active walk-assist machine is expected to be unstable. Conducting this experiment is highly risky and costly. Therefore, we had developed a dynamic simulator in order to test walking robot with walk-assist machine before conducting it in real simulation.

1 INTRODUCTION

With the rapid aging of society in recent times, the number of people with limb disabilities is increasing. According to the research by the Health, Labour and Welfare Ministry, Japan, there are around 1,749,000 people with limb disabilities; this accounts for more than half of the total number of disabled people (3,245,000 handicapped people) (Health). The majority of these people suffer from lower-limb disabilities. Therefore, the demands for establishing a human walking model that can be adapted to clinical medical treatment are increasing. Moreover, this model is required for facilitating the development of rehabilitation and medical welfare instruments such as walking machines for assistance or training (Figure 1(a)). However, experiments that are carried out to estimate the effectiveness of such machines by the elderly or handicapped could result in serious bodily injury.

Many research groups have been studying biped humanoid robots in order to realize the robots that can coexist with humans and perform a variety of tasks. For examples, a research group of HONDA

has developed the humanoid robots—P2, P3, and ASIMO (Sakagami et. al, 2002).



(a) by human (b) by robot
Figure 1: Walk-assist machine.

The Japanese National Institute of Advanced Industrial Science and Technology (AIST) and Kawada Industries, Inc. have developed HRP-2P. The University of Tokyo developed H6 and H7, and the Technical University of Munich developed Johnnie. Waseda University developed the

WABIAN series that realized various walking motions by using moment compensation. Korea Advanced Institute of Science and Technology (KAIST) also developed a 41-DOF humanoid robot—KHR-2 (Omer et. al, 2005).

The above mentioned human-size biped robots achieved dynamic walking. If these humanoid robots can use rehabilitation or welfare instruments as shown in Figure 1(b), they will be able to help in testing such instruments quantitatively. The main advantages of the human simulator can be considered to be as follows: (1) The measurement of the angle and the torque required at each joint can be measured easily and quantitatively as compared to the corresponding values in the case of a human measurement. (2) Experiments using such robots can help identify leg defects of a human from an engineering point of view. (3) A robot can replace humans as experimental subjects in various dangerous situations: experiments involving the possibility of falling, tests with incomplete prototype instruments, simulations of paralytic walks with temporarily locked joints.

Such experiments require a humanoid robot that enables it to closely replicate a human. However, humans have more redundant DOFs than conventional biped humanoid robots; this feature enables them to achieve various motions. Therefore, a DOF configuration that is necessary to reproduce such motions is one of the very important issues in the development of a humanoid robot (Ogura et. al, 2006).

The Waseda Bipedal Humanoid Robot WABIAN-2R has been developed to simulate human motion. WABIAN-2R performed human-like walking motions (Figure 2). Moreover, WABIAN-2R achieved to perform walking motion using walk-assist machine. However, the walk-assist machine was freely rolling without activating its wheels motors. In this case, the robot faced the minimum resistance or disturbance case by the walk-assist machine. On the other hand, activating the walk-assist machine may create a large disturbance for robot due to separate control for each of them. Conducting this experiment may be highly risky.

As we develop humanoid robot to coexist in the human environment, we need to conduct many experiments such as robot walking on uneven surface, climbing the stairs, and robot interact with other machine and instruments. Doing any new type of experiment using WABIAN-2 might be risky.

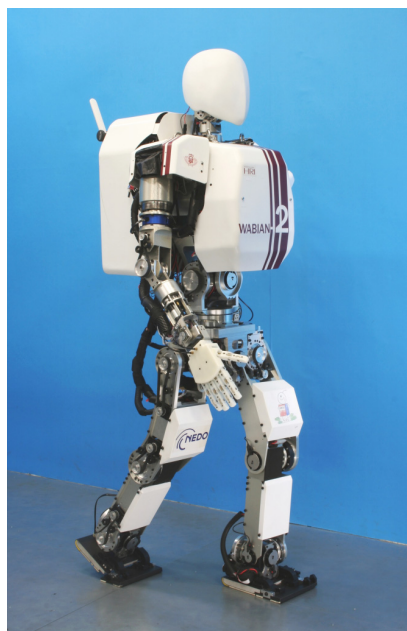


Figure 2: WABIAN-2R.

Therefore, we need find a safer method for initial experimental testing. Using a dynamic simulation is useful method due to some reasons such as: (1) It is safer in terms of cost and risk. (2) It is easy to monitor and view motion outputs. (3) It can show the variation caused by any external disturbances. In this paper, a dynamic simulator is described, which is able to easily simulate any new type of walking. Using the dynamic simulator, we can monitor the motion performance and output all needed data that is useful for further development. This paper is aimed to simulate the walking motions of WABIAN-2 using walk-assist machine.

2 DYNAMIC SIMULATION

Dynamic simulation could be used to simulate the dynamic motion of a mechanical structured model. It can analyze the effects of the surrounding environment on the mechanisms and objects. In robotics researches, simulation software are used for robotic simulation. There are many software used for robotics simulation in different applications. Most of those software are for industrial robot applications. However, there are some software used for mobile robot simulation. For examples, RoboWorks, SD/FAST, OpenHRP, and Yobotics are used for mobile and legged robot simulation. Webots is high and advanced simulation software used in Robotics simulation. It is use for prototyping

and simulation of mobile robots. It has many advanced functions and techniques. Webots is very easy to use and implement. Therefore, we choose it as simulation software (Webots).

2.1 Modeling

In order to develop a dynamic simulation, we need to go through several steps. First is modeling where we set up the simulation environment and initial parameters. We set up a full structure of WABIAN-2, based on the specifications (size, shape, mass distribution, friction, .etc) of components of WABIAN-2 (Figure 3).

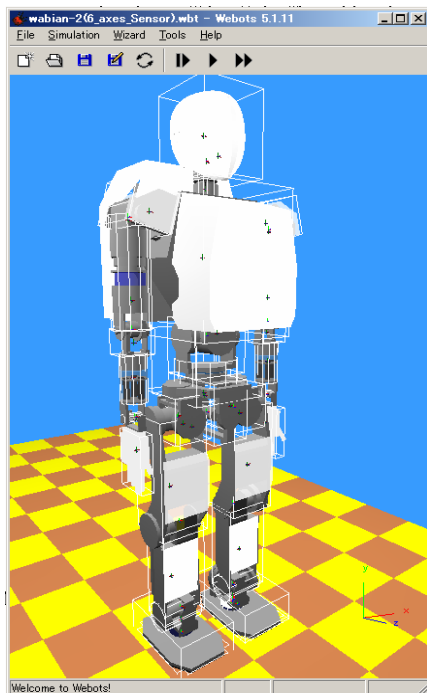


Figure 3: Modeled WABIAN-2R in the simulation world.

2.2 Controlling

Second is controlling, which identifies simulation objects and controls the simulation procedures. The controller is some how similar to the WABIAN-2R control. It gets the input data from the CSV pattern file, and sets the position angle of each joint through inverse kinematics techniques. Moreover, the controller sets the simulate time step and the measurement of data.

2.3 Running

Lastly is the running of the simulation and checking the dynamic motion. We can view the simulation

from different view sides which gives us a clear idea about the simulation performance. Moreover, most of the needed data could be measured through several functions.

3 WALKING WITH WALKING ASSIST MACHINE

WABIAN-2 performed some walking experiments using walking assist machine. The performance was conducted by leaning its arms on the walking assist machine holder. The walking assist machine moves passively without generating its own motion. The robot was able to walk and push the walking assist machine forward. The experiments were conducted with different walking styles and different heights of arm rest.

The walking performance of WABIAN-2 using an active walking assist machine, expected to be unstable. The walk-assist machine has its own control system, not connected to WABIAN-2 control system. The walking assist machine moves with constant velocity in a forward direction, while the robot moves by setting its position. The robot arms may displace from its position on the arm rest of the machine which will cause external forces on WABIAN-2. In order to stabilize the walking, the external force has to be minimized.

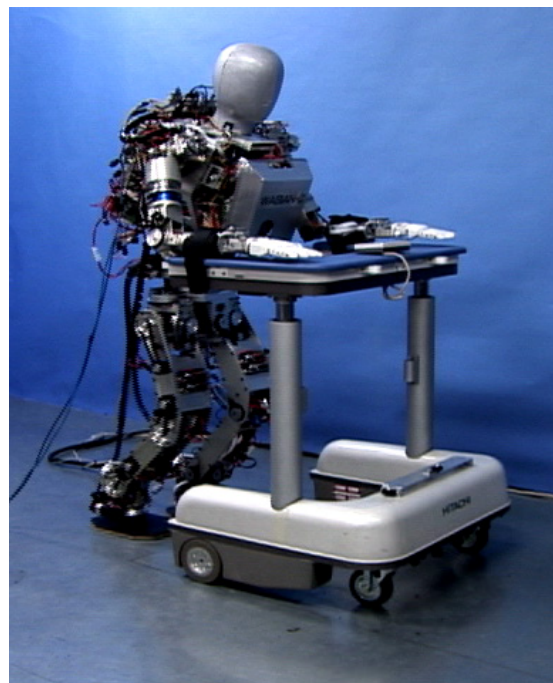


Figure 4: WABIAN-2 with Walking Assist Machine.

3.1 Force Sensor

The real walking assist machine is developed to sense the force applied by the load on the arm rest. A force sensor is attached on the top of the arm rest consisting of four displacement sensors. The displacement sensor is simply a spring mechanism. It senses forward and vertical forces and turns torque by determining relative displacements between the upper frame and the lower one (Figure 5). We can develop the system that can adjust the velocity of the walking assist machine in order to minimize the displacement.

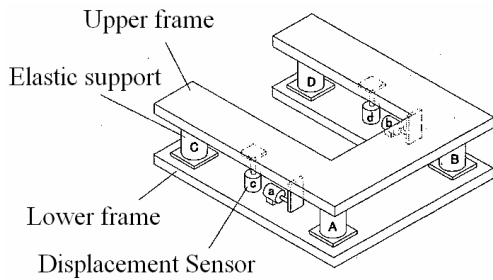


Figure 5: Force Sensor.

3.2 Velocity Control

There were some developments made on the walking assist machine control system to adjust its speed according to the force applied on the arm rest (Egawa et. al, 1999). The arm rest is designed to measure the force and torques applied by the user of the machine (Figure 6). The controller uses those measure data as an input data to set the velocity of each motor of the machine (Figure 7). The force f_y and the turning moment m which applied by the arm of the user is calculated in the sensor by the following equations:

$$m_z = m + s_x f_y \tag{1}$$

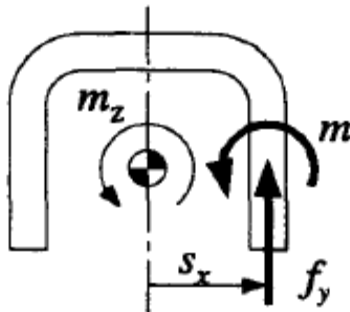


Figure 6: Force and Moment Applied to Arm rest.

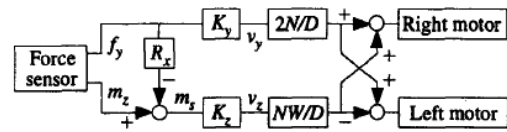


Figure 7: Block Diagram for Control System.

where m_z is the moment measured by the sensor, s_x the distance shifted from the arm position to the sensor position. The values for m_z and f_y are the input data for the controller that set the velocity of each wheel motor (Egawa et. al, 1999).

In this study, we have introduced a new control system model that controls the velocity of the walking assist machine. The system adjusts the velocity according to the force measured by the force sensor. The new adjusted velocity is based on current velocity and the displacement with WABIAN-2.

Developing the equations of the modeled system, we can have the following equation:

$$F_y = ma \tag{2}$$

where m is the total mass of the walking assist machine, a is the acceleration, and F_y is the force measured by the spring. The force is the result of displacement of the spring mechanism, which can be expressed as

$$F_y = Cx \tag{3}$$

where C is the spring constant and x is the amount of displacement. Substitute equation (3) in (2), we will have

$$a = (C/m) x \tag{4}$$

the acceleration is the derivative of velocity. Approximately, it is equal to the difference in velocity over step, which could be express as

$$a(t) = (v(t + \Delta t) - v(t)) / \Delta t \tag{5}$$

since we are dealing with discrete time, we can rearrange equation (5) to

$$a(k) = (v(k+1) - v(k)) / T \tag{6}$$

where $v(k)$ is the current velocity, $v(k+1)$ is the next velocity, and T is the step time. Substitute equation (4) in (6), we will have

$$v(k+1) = (C T / m) x(k) + v(k) \tag{7}$$

where $x(k)$ refer to the displacement measured by the spring of the sensor. This equation represents the velocity control process in the system.

4 SIMULATION RESULT

We test several types of motions performed by WABIAN-2. The simulator simulates the walking performance of conventional walking and stretch walking (Figure 8). Moreover, it simulates some other motions as the input pattern. The dynamic simulation has given us a simulation motion just like the real simulation. We monitor the simulation from different viewpoints. Moreover, we could measure some output data.

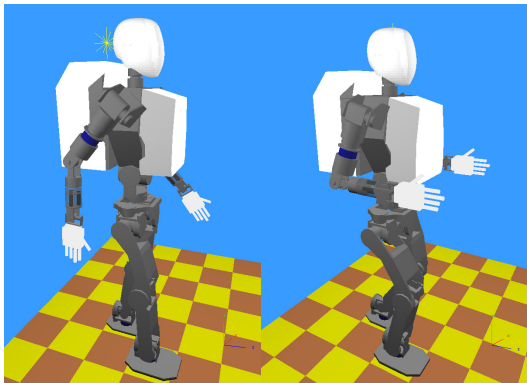


Figure 8: Simulation of different type of walking.

We conducted some simulation experiments of walking using the walking assist machine. The robot is able to walk stably with a passive walking assist machine just like the real experiment (Figure 9). But it was not possible to achieve the same result when we conducted the experiment using an active walking assist. As expected, the robot was affected by the external force produced by its contact with the walking assist machine. The robot became unstable during its walking, and in some experiments it fell down (Figure 10).

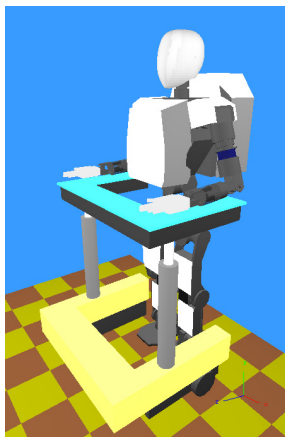


Figure 9: Simulation of walking with walk-assist machine.

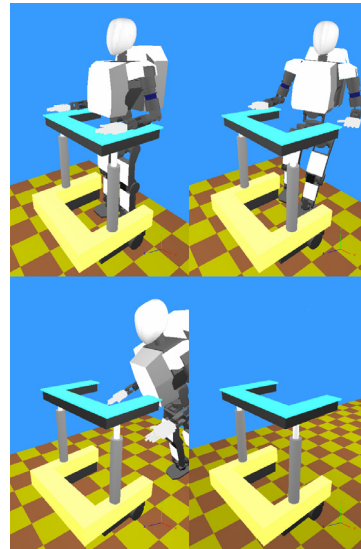


Figure 10: Simulation of walking with active walking machine.

By adding the force sensor to the simulated walking assist machine, we were able to measure the amount of external force acting on the robot. Using these measurements with the velocity control we had developed, the robot could walk with the active walking assist machine. The amount of holding torque we set to the walking assist machine wheels could increase from 0.5 N.m to 0.75 N.m by using this new velocity control in the control system of the dynamic simulator we had developed (Figure 11).

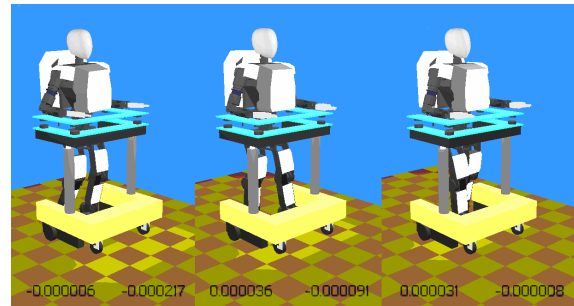


Figure 11: Simulation of walking with the walking machine using velocity control.

5 CONCLUSIONS AND FUTURE WORK

This paper describes the simulation of walking by WABIAN-2R with the walking assist machine. The dynamic simulation is very important to check the motion of any new pattern generated. Using the dynamic simulation we can see the effect of the

walking assist on WABIAN-2R. As expected, the walking was unstable due to the effect of external forces created from the arm rest. By using the velocity control in the control system of the simulation, the robot is able to walk stably with the walking assist machine.

In the near future, it is important to develop WABIAN-2R system to be stabilized during walking. The stabilization control will be based on Zero Moment Point. Moreover, it is necessary to develop the robot to interact with other objects and equipments. This will make the robot can interact with its surrounding environment.

IEEE International Conference on, Volume: 1, 11-15 May 2002 Robotics and Automation, 2002. Proceedings.

Philippe Sardain and Guy Bessonnet. *Force Acting on a Biped Robot Center of Pressure-Zero Moment Point*. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol. 34, NO. 5, September 2004.

REFERENCES

- Health, Labour and Welfare Ministry of Japan. <http://www.mhlw.go.jp/english/wp/wphw/vol1/p2c4s2.h-tml>. The current situation of people with disabilities.
- Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration," Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp.2478-2483, 2002.
- Aiman Musa M. Omer, Yu Ogura, Hideki Kondo, Akitoshi Morishima, Giuseppe Carbone, Marco Ceccarelli, Hun-ok Lim, and Atsuo Takanishi. *Development of a Humanoid Robot Having 2-DOF Waist and 2-DOF Trunk*. Humanoid2005 Conference, Tsukuba- December 2005.
- Yu Ogura, Hiroyuki Aikawa, Kazushi Shimomura, Hideki Kondo, Akitoshi Morishima, Hun-ok Lim, and Atsuo Takanishi. *Development of a New Humanoid Robot WABIAN-2*. Proceedings of the 2006 IEEE International Conference on Robotics and Automation Orlando, Florida - May 2006.
- Webots. <http://www.cyberbotics.com>. Commercial Mobile Robot Simulation Software.
- S. Mojon. *Realization of a Physic Simulation for a Biped Robot*. Semester Project at BIRG laboratory Swiss Federal Institute of Technology, Summer 2003.
- S. Egawa, Y. Nemoto, M. G. Fujie, A. Koseki, S. Hattori, T. Ishii S. Egawa, Y. Nemoto, M. G. Fujie. *Power-Assisted Walking Support System with Imbalance Compensation Control for Hemiplegics*. Proceedings of the Rrst Joint BMES/EMBS Conference Serving Humanity, Advancing Technology o& 1&16, 99, Athn\$, GA, USA.
- Saku Egawa, Ikuo Takeuchi, Atsushi Koseki, Takeshi ISHI. *Force-sensing Device for Power-assisted Walking Support System*. System Integration Conference, December 2002.
- P. E. Klopsteg and P. D. Wilson et al., *Human Limbs and Their Substitutes*, New York Hafner, 1963.
- F. Kanehiro, K. Fujiwara, S. Kajita, K. Yokoi, K. Kaneko, H. Hirukawa, Y. Nakamura, K. Yamane. *Open Architecture Humanoid Robotics Platform*. ICRA '02.

PEOPLE TRACKING USING LASER RANGE SCANNERS AND VISION

Andreas Kräußling, Bernd Brüggemann, Dirk Schulz

Department of Communication, Information Processing and Ergonomics (FKIE)

Research Establishment for Applied Sciences (FGAN), Neuenahrer Straße 20, 53343 Wachtberg, Germany

{a.kraeussling, brueggemann, schulz}@fgan.de

Armin B. Cremers

Institute of Computer Science III, University of Bonn, Römerstraße 164, 53117 Bonn, Germany

abc@iai.uni-bonn.de

Keywords: People tracking, crossing targets, laser range scanners, vision, HSV colour space.

Abstract: Tracking multiple crossing people is a great challenge, since common algorithms tend to lose some of the persons or to interchange their identities when they get close to each other and split up again. In several consecutive papers it was possible to develop an algorithm using data from laser range scanners which is able to track an arbitrary number of crossing people without any loss of track. In this paper we address the problem of rediscovering the identities of the persons after a crossing. Therefore, a camera system is applied. An infrared camera detects the people in the observation area and then a charge-coupled device camera is used to extract the colour information about those people. For the representation of the colour information the HSV colour space is applied using a histogram. Before the crossing the system learns the mean and the standard deviation of the colour distribution of each person. After the crossing the system relocates the identities by comparing the actually measured colour distributions with the distributions learnt before the crossing. Thereby, a Gaussian distribution of the colour values is assumed. The most probably assignment of the identities is then found using Munkres' Hungarian algorithm. It is proven with data from real world experiments that our approach can reassign the identities of the tracked persons stable after a crossing.

1 INTRODUCTION AND RELATED WORK

Multi-robot systems and service robots need to cooperate with each other and with humans in their environments. For this reason, they have to know about the locations and actions of the objects they want to interact with. Target tracking deals with the state estimation of one or more objects. It is a well studied topic in the field of aerial surveillance using radar devices (Bar-Shalom and Fortmann, 1988) and also in the area of mobile robotics. Here, mainly laser scanners are used for the purpose of people tracking (Prassler et al., 1999; Schulz et al., 2001; Fod et al., 2002; Romera et al., 2004; Zhao and Shibasaki, 2005; Bellotto and Hu, 2007). Due to the high resolution of laser scanners, which mostly cover a 180 degree field of view with 180 or 360 measurements, one target is usually the source of multiple returns within one laser scan. This conflicts with the assumption of punctiform targets used in the field of radar tracking.

There, each target is the origin of exactly one measurement. In contrast to that, using laser scanners, one needs to be able to assign the obtained measurements to extended targets.

A second important characteristic of tracking in the field of mobile robotics is the occurrence of crossing or interacting targets, for example two or more persons getting close to each other, so that they can no longer be distinguished by common tracking algorithms (Fortmann et al., 1983; Kräußling et al., 2005; Kräußling et al., 2007). In this article we present an approach to deal with this particular problem. The key idea of our approach is to adopt an algorithm for tracking punctiform objects in clutter, known from the radar community, for the purpose of reliably tracking extended objects with laser scanners. Several different methods for tracking punctiform crossing targets in clutter, i.e. tracking in the presence of false alarm measurements close to a target, have been developed over the last decades:

1. the MHT (Multi Hypothesis Tracker) introduced

by Reid in 1979 (Reid, 1979).

2. The JPDAF (Joint Probabilistic Data Association Filter) introduced by Fortmann, Bar-Shalom and Scheffe in 1983 (Fortmann et al., 1983).

These techniques can easily be extended to tracking extended objects as well, but there are several reasons, why such approaches are brittle:

- In most cases there are several measurements from the same target.
- Interacting objects might be indistinguishable over longer periods of time.
- Some of the objects might be occluded for some time.
- The objects can carry out abrupt manoeuvres, especially when they are crossing their paths.

These difficulties are well known in the mobile robotics community:

- Tracking moving objects whose trajectories cross each other is a very general problem ... Problems of this type cannot be eliminated even by more sophisticated methods ... (Prassler et al., 1999).
- Tracks are lost when people walk too closely together ... (Schumitch et al., 2006).

Due to these reasons, we have developed methods for tracking interacting people in laser data (Kräußling et al., 2004b; Kräußling et al., 2005; Kräußling et al., 2007). These methods have in common, that they employ a variant of the well known Viterbi algorithm (Viterbi, 1967; Forney Jr., 1973) in combination with geometrical properties of the people tracking problem, in order to achieve a high degree of robustness against track loss.

However, although the tracks are very rarely lost by these algorithms, they tend to confuse the assignment of the tracks to the individual persons being tracked after a crossing of paths has occurred. This happens because the distance measurements of the laser scanners do not provide direct information about the persons' identities. For this reason, additional cues are required, if we want to reliably distinguish between persons. Possible cues are:

1. Different colours and surface-textures of the pairs of trousers people wear might result in different intensities of the reflected laser beams.
2. Ultrasound or infrared signals uniquely identifying individuals, which are transmitted by special active badges the people wear (Schulz et al., 2003).
3. Different colours of the clothes people wear. This information can be exploited for the identification of the people using a camera network.

4. Differences in physiognomy like size and built of persons. These differences can again be detected using cameras (Schulz, 2006).

In this article we propose a technique to combine Viterbi-based tracking with person identification based on colour information. A calibrated setup consisting of an infrared and a CCD camera is used to learn colour histograms of the persons, while they are well separated during tracking. This information is then employed to correctly reassign person IDs to tracks after interactions have occurred. The new assignments are determined using the Hungarian algorithm, which computes the maximum likelihood assignment, based on the likelihood of colour observations. Our experiments show that this approach allows to track several interacting humans without loss of track and without accidental confusion of the track assignments.

The remainder of this paper is organised as follows. In Section 2 the combined method for tracking multiple interacting persons is described. It consists of the tracking method based on the Viterbi algorithm and an identity assignment method based on colour information. Section 3 presents experiments illustrating the robustness of our approach against loss of track as well as against errors in track assignment. We conclude in Section 4.

2 THE METHOD

In order to reliably keep track of several interacting persons, we have to solve two problems: the trajectories of the persons have to be estimated without losing track of the persons and we have to make sure, that we can always assign the individual trajectories to the correct person. We have developed the so called Cluster Sorting algorithm (CSA) to solve the first problem (Kräußling, 2006b). The CSA uses data from laser range scanners to estimate the trajectories of objects over time. The second problem is then solved by additionally using colour histograms extracted from camera images, in order to compute the most likely assignment of the trajectories to the persons being tracked. In the following, we will first explain the CSA in detail. Afterwards we will describe how the reassignment of tracks based on colour information can be integrated into the approach.

The Cluster Sorting algorithm estimates multiple trajectories using a hidden Gauß-Markov chain, where the tracking process is carried out using Kalman filters. Because laser range scanners return range measurements to any object in the surrounding of the robot, the measurements originating

from persons have to be discriminated from measurements of static objects. The CSA computes validation gates (Bar-Shalom and Fortmann, 1988) for this purpose, i.e. only measurements, which are close to the currently estimated positions of persons are being considered; we call those measurements the selected measurements. The distinction between measurements of different persons is possible based on the distance between selected measurements, as long as persons do not get close to each other; otherwise, persons share selected measurements. For this reason, the CSA deals with the selected measurements in two different ways:

(1) as long as the measurements of persons are well separated, it computes for each person the unweighted mean of all the selected measurements of that person. These means are then used to update the Kalman filters for the individual trajectories of each person; we call this procedure the Kalman Filter Algorithm (KFA). It has been shown in (Kräußling et al., 2005; Kräußling, 2006a) to be very fast and to provide good information about the position of the targets, but it cannot reproduce multi-modal probability distributions. Thus it is not able to handle multiple interacting people.

(2) When persons have selected measurements in common, the CSA no longer computes one single track for each person, but it starts to compute individual tracks for each selected measurement of each person using a variant of the Viterbi algorithm. The algorithm calculates for every old selected measurement a separate position estimate and validation gate. The new selected measurements are the ones which lie in at least one of those gates; we call this algorithm Viterbi-based algorithm (VBA); it has been introduced in (Kräußling et al., 2004a). The algorithm allows to represent multi-modal probability distributions to some extent, which is a major advantage when dealing with multiple interacting targets. The VBA is much more robust against track loss, when compared to the KFA, because the VBA maintains several hypotheses about a persons position, one hypothesis for each gating measurement. Common tracking algorithms like the KFA, in contrast, make a hard decision which measurement they use. In difficult situations, they tend to assign the same measurement to several objects. Algorithms with a random component like the SJPDAF (Schulz et al., 2001) occasionally choose for each track the path of a different person, so that no persons gets lost. But this behaviour is not stable (Kräußling and Schulz, 2006).

It remains to describe, how the CSA actually decides when to switch between the KFA and the VBA. The Cluster Sorting algorithm uses two classes of ob-

jects:

- single targets.
- clusters, which represent at least two interacting persons, i.e. humans that are moving very close to each other.

Single targets are tracked with the KFA, since there is no need for representing multi-modal probability distributions. Clusters are tracked with the VBA, since multi-modal distributions have to be represented. This approach guarantees that none of the objects that are associated with the cluster is lost. This fact is important especially when the objects split and start to move separately again.

Three different events have to be regarded when tracking multiple interacting people:

1. The merging of two single persons. This means that two single targets get very close to each other. This is the case, if at least one measurement is located in the validation gates of both targets. Then the algorithm stops to track the two single targets with the KFA and starts tracking a cluster, which contains both targets, using the VBA. Therefore, it uses the measurements located in the validation gates of at least one target.
2. The merging of a single human and a cluster. This means that a single person and a cluster get very close to each other. This happens, if at least one measurement is located in the validation gates of the person and the cluster. In this case the algorithm stops to track the single human and the cluster separately. Instead it starts tracking a combined cluster. Therefore, it uses the measurements located in the validation gates of either the single target or the previously considered cluster or both.
3. The merging of two clusters. This means that two clusters get very close to each other. This is the case, if at least one measurement is located in the validation gates of both clusters. If this is true, the algorithm stops to track the two clusters and starts tracking a combined cluster. Therefore, it uses the measurements located in the validation gates of at least one of the previously considered clusters.

Note, that whenever a merging takes place, the algorithm remembers the humans which correspond to the newly combined cluster.

For each tracked cluster, we also have to decide, if it has split into single person tracks again. Whether clusters are split depends on three conditions:

1. The position estimates corresponding to the measurements in the validation gates are separated into subclusters. For this purpose, we select the first estimate, which then is associated with the

first subcluster. For all other estimates associated with the cluster, the Euclidean distance to the first estimate is calculated. If this distance is below a certain threshold, the estimate is associated with the first subcluster. In our experiments, we set the threshold to 150 cm , which corresponds to the maximum distance between the legs of a walking person. We then have to consider the estimates, for which the Euclidean distance to the first subcluster exceeds this manually chosen threshold. Using the same procedure we applied for building the first subcluster, we now construct subclusters until all estimates are associated with one of these smaller clusters. If the number of subclusters equals the number of humans which were merged into this cluster, the first condition for the dispersion of the cluster is fulfilled. Then, we proceed with step 2.

2. We now check the pairwise distance between the subclusters. If the distance is above a manually chosen bound, we regard these clusters as separated. We choose the value of that bound to be 300 cm . The second condition is fulfilled, if the number of pairs of separated subclusters equals $\frac{n(n-1)}{2}$. Thereby, n is the number of single persons associated with the cluster. Hence, we are checking if all subclusters are pairwise separated.
3. Above this, we can separate single subclusters from the cluster to indicate them in the graphics. This follows the same logic as in step 1. Note, the algorithm is not able to determine, how many targets are represented by a single subcluster.

If conditions 1 and 2 are met, the n subclusters are associated with the n single targets therefrom tracked by the KFA. When separating targets from clusters, we cannot guarantee if the target association is the same as before merging the targets into the cluster. Thus, a possible solution to this problem, which uses colour information will now be proposed.

To obtain the colour information of the persons, a charge-coupled device camera (CCD camera) is used. In order to recognise which parts of the picture of the CCD camera belong to the persons, we employ an infrared camera. The two cameras are mounted in parallel on the robot, with only a small displacement; this allows us to easily correlate infrared and CCD images. The setup is shown in Figure 1. Since the temperature of the persons is in a small, well defined range, it is easy to identify the regions in the images of the CCD camera which originate from humans. Next, the persons which are detected by the camera system have to be assigned to the persons which are tracked by the laser scanners and the tracking algo-

Table 1: The values of the hue in the HSV colour space.

Hue	red	yellow	green	cyan	blue	magenta
Degrees	0	60	120	180	240	300

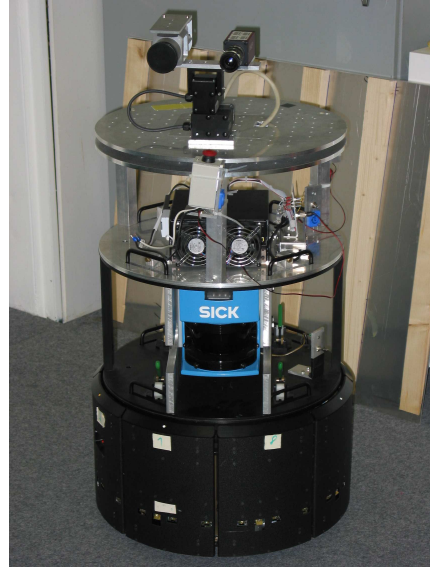


Figure 1: The robot equipped with the laser scanners and the cameras.

rithm. For this purpose, we exploit the fact that the camera system enumerates the persons in clockwise direction. Therefore, we arrange the persons tracked by the tracking algorithm in clockwise direction, too.

The colour information from the CCD camera is represented using the HSV colour space (Gonzalez and Woods, 1992), where H stands for hue, S for saturation and V for value. In our experiments we only used the hue, because it is fairly independent from the illumination and a stable characteristic of the persons. The hue values range from 0 to 360 degrees. The mapping between the values of the hue and six basic colours, we use for identification, is shown in Table 1.

The colour characteristics are learnt during the experiment before the crossings. We assume that the average relative frequencies of the hue for person i follow a Gaussian distribution with mean μ_i and variance Σ_i . Let $\mu_{i,k}$ and $\Sigma_{i,k}$ be the learnt mean and the learnt variance at time step k and let $y_{i,k}$ be the measurement corresponding to person i at time step k ,

$$\mu_{i,k} = \frac{\sum_{j=1}^k y_{i,j}}{k}, \quad (1)$$

$$\Sigma_{i,k} = \frac{\sum_{j=1}^k (y_{i,j} - \mu_{i,k})(y_{i,j} - \mu_{i,k})^T}{k-1}. \quad (2)$$

This mean and this variance can be learnt on-line

without storing previous values $y_{i,l}$, $l < k$, according to

$$\Delta_{k+1} = \mu_{i,k+1} - \mu_{i,k} \quad (3)$$

$$\mu_{i,k+1} = \frac{k}{k+1}\mu_{i,k} + \frac{y_{i,k+1}}{k+1} \quad (4)$$

$$\begin{aligned} \Sigma_{i,k+1} = & \frac{(y_{i,k+1} - \mu_{i,k+1})(y_{i,k+1} - \mu_{i,k+1})^\top}{k} + \\ & + \frac{k-1}{k}\Sigma_{i,k} + \Delta_{k+1}\Delta_{k+1}^\top. \end{aligned} \quad (5)$$

As soon as the tracking algorithm detects a crossing, the algorithm stops to learn the colour characteristics and the actual values $\mu_{i,k}$ and $\Sigma_{i,k}$ are assigned to the persons as the characteristics μ_i and Σ_i . As soon as the tracking algorithm detects the end of the crossing, the algorithm reassigns the identities to the persons. If $y_{j,k}$ is the measurement of the person that is assigned to track j at time step k , then the probability $p_{i,j,k}$, that the track j at time step k belongs to the person, which has been assigned the identity i before the crossing, is

$$\begin{aligned} p_{i,j,k} = & \frac{1}{(\det(2\pi\Sigma_i))^{1/2}} \cdot \\ & \exp\left\{-\frac{1}{2}(y_{j,k} - \mu_i)^\top \Sigma_i^{-1} (y_{j,k} - \mu_i)\right\}. \end{aligned} \quad (6)$$

Because the colour measurements of different points in time are independent, the probability $p_{i,j,k_1:k_2}$, that the track j from time step k_1 to time step k_2 originates from person i is

$$p_{i,j,k_1:k_2} = \prod_{k=k_1}^{k_2} p_{i,j,k}. \quad (7)$$

The tracks j , which are calculated by the tracking algorithm, are usually interchanged during a crossing. Thus, let m be the total number of persons associated with the cluster being split and let σ be a permutation of the person IDs $1, \dots, m$. Then, the probability that the IDs have been interchanged during the crossing according to the permutation σ given the measurements from time step k_1 to time step k_2 is

$$\Pi_\sigma = \prod_{l=1}^m p_{l,\sigma(l),k_1:k_2}. \quad (8)$$

The best reassignment $\hat{\sigma}$ of the tracks j to the learnt persons i is the one, that maximises the probability Π_σ . To compute $\hat{\sigma}$ we interpret the negative log-likelihoods $\log(p_{i,j,k_1:k_2})$ as the marginal assignment costs of assigning track j to person i after a crossing. The negative log-likelihoods $\log(\Pi_\sigma)$ then constitute the assignment cost of a complete assignment (permutation) σ . The minimum cost assignment $\hat{\sigma}$ is then calculated using the well known Hungarian algorithm (Munkres, 1957).

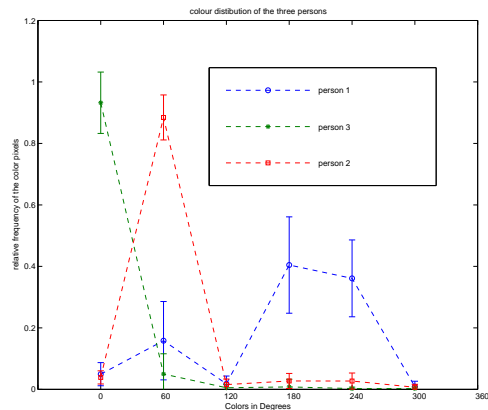


Figure 2: Colour distributions of the three subjects.

3 EXPERIMENTS

We conducted experiments with three persons in a real world scenario in our laboratory. The first person was wearing a blue cardigan and a blue pair of trousers. The second person was wearing a yellow cardigan and blue pair of trousers. The third person was wearing a red shirt and a red pair of trousers.

Figure 2 shows the corresponding colour distributions. The first subject has his maximum in the blue domain, the second has it in the yellow domain and the third in the red one. Thus, the measured colour distributions show a good coincidence with the real colours.

The experiments were accomplished with a B21 robot platform shown in Figure 1. On the top the camera system is mounted. The left camera is the infrared camera and the right camera is the CCD camera. There are two laser range scanners armed back to back at the robot, so that there is a 360 degree field of view.

The number of possible permutations of three objects is $3! = 6$. Thus, we conducted six experiments, for each permutation one experiment. We defined position 1 as the right upper part of the surveillance area, position 2 as the left upper part of the surveillance area and position 3 as the lower middle part. The persons are indexed in the order they appear in the surveillance area. In the six experiments the person indexed 1 occupied position 1 before the crossing, the person indexed 2 occupied position 2 and the person indexed 3 occupied position 3. After the crossing they occupied different positions corresponding to the six possible permutations. The experiment for the permutation $123 \mapsto 123$ is described in detail.

At first the three persons are occupying their start

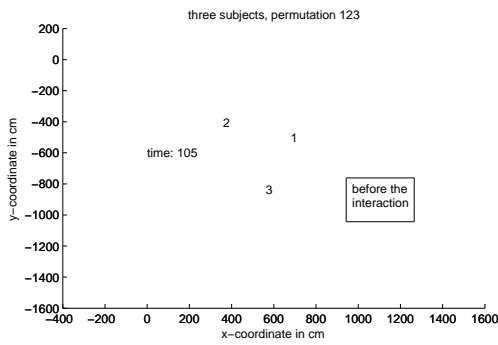


Figure 3: Three subjects before the crossing, permutation 123 \mapsto 123.

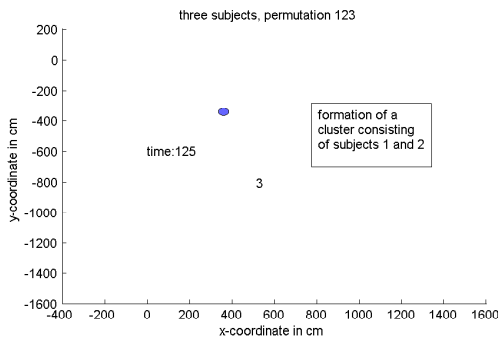


Figure 4: Formation of a cluster consisting of person 1 and 2, permutation 123 \mapsto 123.

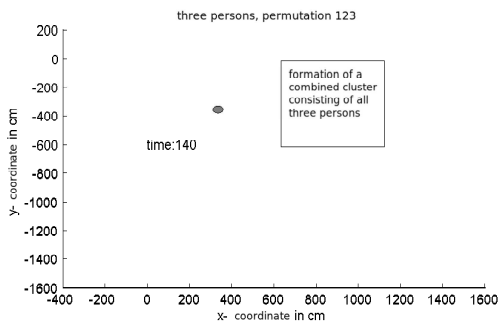


Figure 5: Formation of a cluster consisting of all three persons, permutation 123 \mapsto 123.

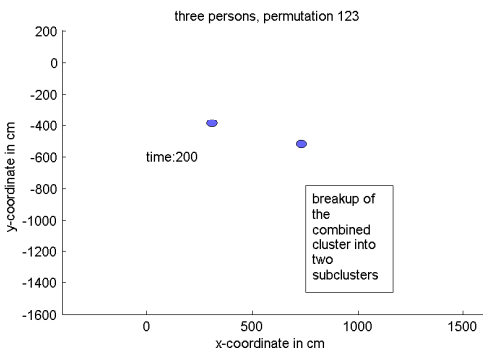


Figure 6: Disaggregation of the combined cluster into two subclusters, permutation 123 \mapsto 123.

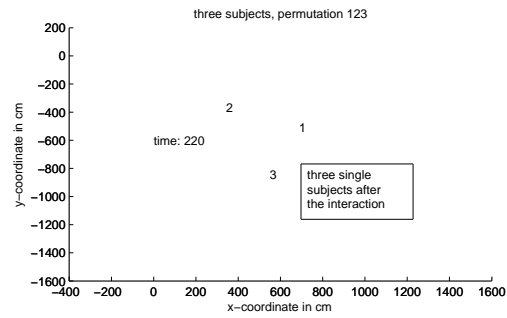


Figure 7: Three persons after the crossing, permutation 123 \mapsto 123.

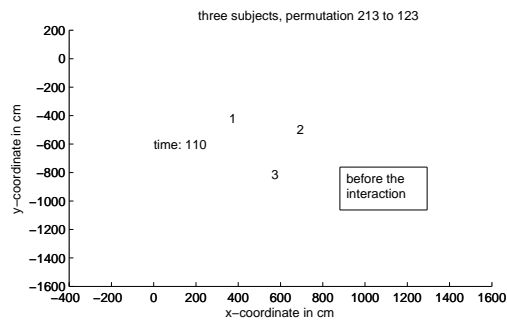


Figure 8: Three persons before the crossing, permutation 213 \mapsto 123.

positions and are indexed in the order of their appearance in the surveillance area; this is illustrated in Figure 3. Within this figure, the numbers corresponding to a person are drawn at the location computed by the tracking algorithm. In Figure 4 the persons 1 and 2 interact and merge into a cluster. The clusters are represented by ellipses within the figure. In the next step person 3 joins the other two and the algorithm merges them into a single cluster as shown in Figure 5. After some time, the group splits up into two subclusters (see Figure 6), and finally the three persons walk on their own again. This situation is illustrated in Figure 7. As can be seen, the algorithm tracks the three persons without loss of track and reassigns the identities correctly after the interaction.

Next, we investigated the question, whether the algorithm still works well, when the starting positions are interchanged. For this purpose we used the permutation 213 \mapsto 123, which means for instance, that the initial position of person number 2 is position number 1. Figures 8 and 9 show the starting and the end positions respectively with the assigned identities. Obviously the identities are in this case also reassigned correctly.

Finally, we examined the case, whether the algorithm can deal with several consecutive permutations. Therefore, we used the two consecutive permutations 123 \mapsto 213 and 213 \mapsto 231. Figure 10 shows the ini-

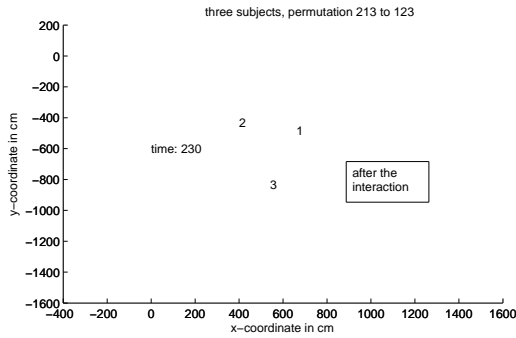


Figure 9: Three persons after the crossing, permutation 213 \mapsto 123.

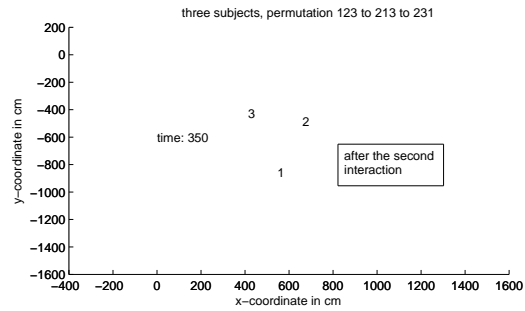


Figure 12: Three persons after the second crossing, permutation 123 \mapsto 213 \mapsto 231.

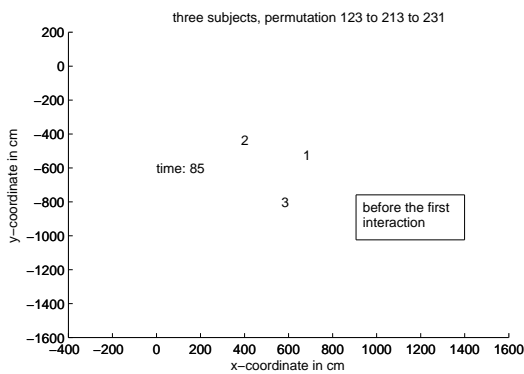


Figure 10: Three persons before the first crossing, permutation 123 \mapsto 213 \mapsto 231.

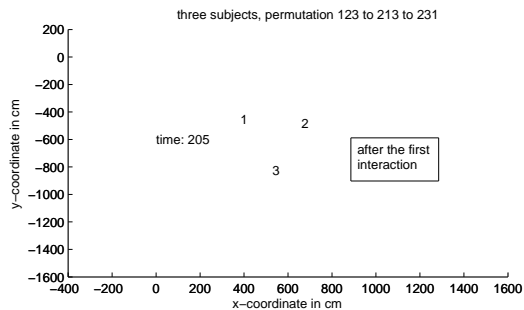


Figure 11: Three persons after the first crossing, permutation 123 \mapsto 213 \mapsto 231.

tial positions. Figure 11 shows the reassigned identities after the first crossing and Figure 12 shows the reassigned identities after the second crossing. It can easily be recognised that the identities are reassigned correctly after each crossing.

4 CONCLUSIONS

In this article we investigated the problem of tracking multiple interacting humans. There are two difficulties, which have to be addressed to solve this challenging problem:

1. The robot should not loose track of a person and
2. the robot should always assign the correct identity to the individual persons, especially after a group of interacting persons split.

We proposed a hybrid approach to solve these two problems, a laser-based tracking approach is applied to keep track of the trajectories of persons, and colour information about the persons' clothes is employed to disambiguate between the persons being tracked.

The proposed tracking algorithm reliably keeps track of several persons, even in very difficult situations, like the crossing of tracks and interactions of persons. Robustness against track loss is achieved by applying a switching approach, where a simple Kalman filter is used as long as tracks are well separated and a variant of the Viterbi algorithm, which tracks individual laser measurements independently, takes over as soon as persons get close to each other. A clustering technique is then employed to assign the measurement tracks to individual person tracks again, when the persons split up again.

However, during the Viterbi phase, the assignment of the persons identities to tracks is lost. For this reason, we employ the camera information to correctly reassign the persons IDs to the individual tracks after the crossing. The current implementation uses a combination of an infrared and a CCD camera for this purpose. The camera system provides the robot with colour information about the tracked persons. The most likely assignment of the identities is then found by using the Hungarian algorithm.

Our experiments show that this approach is able to reliably track multiple interacting persons without

interchanging the individual tracks even in challenging situations. But of course there are still possibilities for future research. The approach will run into problems, if the colour distributions of the peoples' clothes become too similar. This could be remedied by additionally taking size and shape information into account, like in (Schulz, 2006). In rare situations it is also still possible that the tracking algorithm loses track of an individual person, e.g. if a human moves away while it is in the shadow of the other persons during a crossing. This drawback could be overcome by coordinating a team of robots in order to keep full coverage of the scene.

REFERENCES

- Bar-Shalom, Y. and Fortmann, T. (1988). *Tracking and Data Association*. Academic Press.
- Bellotto, N. and Hu, H. (2007). People tracking with a mobile robot: A comparison of Kalman and particle filters. In *Proceedings of the 13th IASTED International Conference on Robotics and Applications*, pages 388–393. International Association of Science and Technology for Development.
- Fod, A., Howard, A., and Mataric, M. J. (2002). Laser-based people tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3024–3029.
- Forney Jr., G. D. (1973). The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.
- Fortmann, T. E., Bar-Shalom, Y., and Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3).
- Gonzalez, R. C. and Woods, R. E. (1992). *Digital Image Processing*. Addison-Wesley.
- Kräußling, A. (2006a). Tracking extended moving objects with a mobile robot. In *Proceedings of the 3rd IEEE Conference on Intelligent Systems, also to be published in the Post Conference Volume (IEEE IS-06): Intelligent Techniques and Tools for Novel System Architectures, Series: Studies of Computational Intelligence*, Springer.
- Kräußling, A. (2006b). Tracking multiple objects using the Viterbi algorithm. In *Proceedings of the 3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, also to be published in the Springer book of best papers of ICINCO 2006, pages 18–25.
- Kräußling, A., Schneider, F. E., and Wildermuth, D. (2004a). Tracking expanded objects using the Viterbi algorithm. In *Proceedings of the 2nd IEEE Conference on Intelligent Systems*.
- Kräußling, A., Schneider, F. E., and Wildermuth, D. (2004b). Tracking of extended crossing objects using the Viterbi algorithm. In *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 142–149.
- Kräußling, A., Schneider, F. E., and Wildermuth, D. (2005). A switching algorithm for tracking extended targets. In *Proceedings of the 2nd International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 126–133.
- Kräußling, A., Schneider, F. E., Wildermuth, D., and Sehestedt, S. (2007). *A switching algorithm for tracking extended targets*, volume II, pages 117–128. Springer, informatics in control, automation and robotics edition.
- Kräußling, A. and Schulz, D. (2006). Tracking extended targets — a switching algorithm versus the SJPDAF. In *Proceedings of the 9th IEEE International Conference on Information Fusion, CD-ROM*.
- Munkres, J. (1957). Algorithms for assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1).
- Prassler, E., Scholz, J., and Elfes, E. (1999). *Tracking people in a railway station during rush-hour*, volume 1542, pages 162–179. Springer Lecture Notes, computer vision systems edition.
- Reid, D. B. (1979). An algorithm for tracking multiple targets. *IEEE Trans. Automatic Control*, 24:843–854.
- Romera, M. M., Vazquez, M. A. S., and Garcia, J. C. G. (2004). Tracking multiple and dynamic objects with an extended particle filter and an adapted k-means clustering algorithm. In *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, CD-ROM*.
- Schulz, D. (2006). A probabilistic exemplar approach to combine laser and vision for person tracking. In *Proceedings of Robotics: Science and Systems II*.
- Schulz, D., Burgard, W., Fox, D., and Cremers, A. B. (2001). Tracking multiple moving objects with a mobile robot. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Schulz, D., Fox, D., and Hightower, J. (2003). People tracking with anonymous and ID-sensors using Rao-Blackwellised particle filters. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*.
- Schumitch, B., Thrun, S., Bradski, G., and Olukotun, K. (2006). The information-form data association filter. In *Proceedings of the 2005 Conference on Neural Information Processing Systems (NIPS)*. MIT Press.
- Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2).
- Zhao, H. and Shibasaki, R. (2005). A novel system for tracking pedestrians using multiple single-row laser-range scanners. *IEEE Transactions on Systems, Man and Cybernetics — Part A: Systems and Humans*, 35(2):283–291.

ROBOTIC WHEELCHAIR CONTROL CONSIDERING USER COMFORT

Modeling and Experimental Evaluation

Razvan Solea and Urbano Nunes

ISR - Institute of Systems and Robotics, Department of Electrical and Computer Engineering

University of Coimbra, 3030-290, Coimbra, Portugal

razvan@isr.uc.pt, urbano@isr.uc.pt

Keywords: Users comfort, sliding-mode control, human head-neck complex model, intelligent wheelchair.

Abstract: This paper analyzes the comfort of wheelchair users when a sliding-mode trajectory-tracking controller is used. The transmission of the horizontal (fore-and-aft) vibration to the head-neck complex (HNC) in the seated human body may cause unacceptable discomfort and motion sickness. A double-inverted pendulum model with two degrees of freedom is considered as a model for the HNC. The user comfort is examined not only in the time domain (using the fourth power vibration dose value), but also in the frequency domain (using the cross-spectral density method). For measuring the acceleration of the wheelchair, along the trajectory, an inertial measurement unit was used.

1 INTRODUCTION

Few studies have been performed addressing how dynamic acceleration affects wheelchair's users. A barrier to performing in-depth analysis during the processes of wheelchair design and ride comfort is a lack of wheelchair-acceleration data, measured over time, that vary with the activity of the wheelchair user. Furthermore, little is known about how this dynamic acceleration affects user comfort. Most current literature focus on the vibration exposure of a seated occupant. To this end, standards have been developed by the International Organization for Standardization (ISO) to quantify how much exposure is allowable for various frequencies of exposure. To standardize the methods of data collection for whole-body vibration, the ISO introduced the ISO-2631 (ISO-2631, 1997). The boundaries in ISO-2631 are based on cumulative root-mean-square (rms) amplitude over a single day, specified for frequencies between 1 and 80 Hz.

The human body is a complex dynamic system, the properties of which vary from moment to moment and from one individual to another. From the results of large amount of experimental data, various biomechanical models have been developed to describe the human motion. These models can be grouped as lumped or distributed parameter models. The lumped parameter models consider the human body as several rigid bodies, springs and dampers (Atapourfard

et al., 2002), (Atapourfard et al., 2004), (Gurses et al., 2005). Some distributed models treat the spine as a layered structure of rigid elements, representing the vertebral bodies and deformable elements representing the intervertebral disc by the finite elements (Kitazaki and Griffin, 1997).

The dynamic response of seated subjects exposed to vibration has been widely assessed in terms of the driving point impedance, apparent mass and transmissibility (transmission of motion through the human body). The transmission of the acceleration to the head-neck complex (HNC) in the seated human body may be the cause of discomfort and motion sickness in wheelchairs. The seat back, by limiting the horizontal and rotational motion of the trunk, increases the transmission of the trunk horizontal acceleration to the HNC. This may has considerable influence on discomfort.

The present study focuses specifically on the influence of sliding-mode trajectory-tracking (SM-TT) controller action on user comfort. The user comfort is examined not only in the time domain (using the transmissibility parameter), but also in the frequency domain. For measuring accelerations of the wheelchair, a three-dimensional inertial sensor was used. The analysis of user comfort is made in three different situations: i) SM-TT control under odometry navigation; ii) when the odometric data is fused with absolute position data from magnetic markers (using

an EKF-based fusion in the on-line pose estimation);
 iii) SM-TT control with purposely-incorrectly-tuned parameters.

2 CONTROL OF WHEELED MOBILE ROBOTS

The application of sliding mode control strategies in nonlinear systems has received considerable attention in recent years (Yang and Kim, 1999), (Chwa, 2004), (Chwa et al., 2006), (Solea and Nunes, 2007). A well-studied example of a non-holonomic system is a wheeled mobile robot (WMR) that is subject to the *rolling without slipping* constraint.

In trajectory-tracking, is an objective to control the non-holonomic WMR to follow a desired path, with a given orientation relatively to the path tangent, even when disturbances exist. In the case of trajectory-tracking the path is to be followed under time constraints. The path has an associated velocity profile, with each point of the trajectory embedding spatiotemporal information that is to be satisfied by the WMR along the path. By other words, path tracking is formulated as having the WMR following a virtual target WMR which is assumed to move exactly along the path with specified velocity profile.

2.1 Kinematic Model of a Unicycle-type Mobile Robot

Let the pose of the mobile robot be defined by the vector $q_r = [x_r, y_r, \theta_r]^T$, where $[x_r, y_r]^T$ denotes the robot position on the plane and θ_r the heading angle with respect to the x -axis. In addition, v_r denotes the translational velocity of the robot, and ω_r the angular velocity around the vertical axis. For a unicycle WMR rolling on a horizontal plane without slipping, the kinematic model can be expressed by:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (1)$$

which represents a non-linear system.

Controllability of the system (1) is easily checked using the Lie algebra rank condition for nonlinear systems. However, the Taylor linearization of the system about the origin is not controllable, thus excluding the application of classical linear design approaches.

2.2 Trajectory Tracking Model

Without loss of generality, it can be assumed that the desired trajectory $q_d(t) = [x_d(t), y_d(t), \theta_d(t)]^T$ is

Table 1: ISO 2631-1 Standard.

Overall Acceleration	Consequence
$a_w < 0.315m/s^2$	not uncomfortable
$0.315 < a_w < 0.63m/s^2$	a little uncomfortable
$0.5 < a_w < 1m/s^2$	fairly uncomfortable
$0.8 < a_w < 1.6m/s^2$	uncomfortable
$1.25 < a_w < 2.5m/s^2$	very uncomfortable
$a_w > 2.5m/s^2$	extremely uncomfortable

generated by a virtual unicycle mobile robot. The kinematic relationship between the virtual configuration q_d and the corresponding reference velocity inputs $[v_d, \omega_d]^T$ is similar to (1). From the error vector (Solea and Nunes, 2007),

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta_d & \sin\theta_d & 0 \\ -\sin\theta_d & \cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d \end{bmatrix} \quad (2)$$

we get the error dynamics:

$$\begin{cases} \dot{x}_e = -v_d + v_r \cdot \cos\theta_e + \omega_d \cdot y_e \\ \dot{y}_e = v_r \cdot \sin\theta_e - \omega_d \cdot x_e \\ \dot{\theta}_e = \omega_r - \omega_d \end{cases} \quad (3)$$

2.3 Trajectory Planner

A trajectory planner for human-transport robots must generate smooth velocity profiles (linear and angular) with low associated accelerations. The trajectory planning process can be divided into two separate parts. First, a continuous collision-free path is generated. In a second step, called trajectory generation, a velocity profile along the path is determined. A method to generate a velocity profile, respecting human body comfort, for any two-dimensional path in static environments was proposed in (Solea and Nunes, 2007).

Figures 1 - 3 show an example of a planned trajectory using the method described in (Solea and Nunes, 2007) were the goal was to obtain an overall rms acceleration in the range of "not uncomfortable" (see Table 1). The overall rms acceleration is defined as:

$$a_w = \sqrt{k_x^2 \cdot a_{wx}^2 + k_y^2 \cdot a_{wy}^2 + k_z^2 \cdot a_{wz}^2} \quad (4)$$

where a_{wx} , a_{wy} , a_{wz} , are the rms accelerations along x , y , z axes respectively, and k_x , k_y , k_z , are multiplying factors. For a seated person $k_x = k_y = 1.4$, $k_z = 1$. For motion on the x - y plane, $a_{wz} = 0$. The local coordinate system is chosen so that the x -axis is the longitudinal trajectory direction, and y -axis is the lateral trajectory direction.

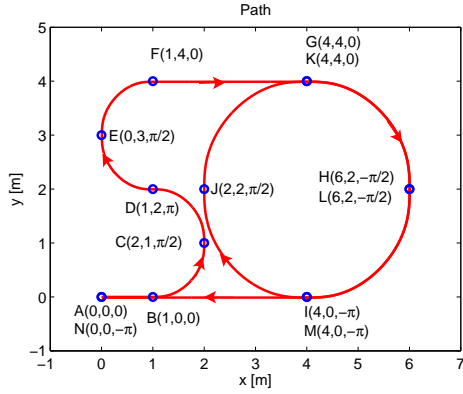


Figure 1: Path example composed by thirteen path segments calculated by the trajectory planner, from the fourteen waypoints A to N.

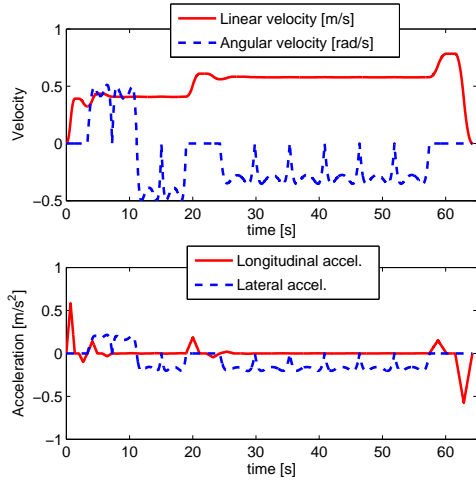


Figure 2: Velocity and acceleration profiles for path depicted in Fig. 1.

The ISO 2631-1 standard (ISO-2631, 1997) (Table 1) relates comfort with the overall rms acceleration, acting on the human body.

Figure 1 shows the generated path, where the larger circles represent the used fourteen waypoints (A, B, ..., N). Each waypoint is defined by a position, in meters, and an orientation, in radians. The generated velocity and acceleration profiles are shown in Fig. 2. As can be observed in Fig.3, the rms overall accelerations, in each path segment, are below the imposed acceleration constraint of $a_{w(i,i+1)} < 0.31m/s^2$.

2.4 Sliding-mode Controller

The objective of SMC is the same as for classical controllers, i.e., force the output states to follow the desired input states. However, the SMC is a model-based control strategy in which the controller struc-

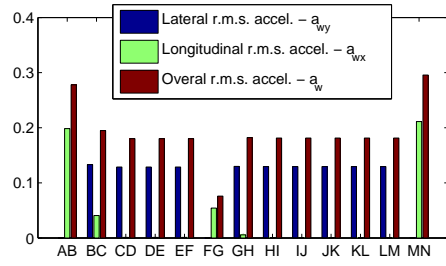


Figure 3: Acceleration rms values for each path segment (AB, BC, ..., MN) of path depicted in Fig. 1.

ture and gains are designed based on the system model.

Uncertainties which exist in real mobile robot applications degrade the control performance significantly, and accordingly, need to be compensated. In (Solea et al., 2006) a SM-TT controller for WMRs is proposed, where trajectory tracking is achieved even in the presence of large initial pose errors and disturbances.

Let us define the sliding surface $s = [s_1 \ s_2]^T$ as

$$\begin{aligned} s_1 &= \dot{x}_e + k_1 \cdot x_e, \\ s_2 &= \dot{y}_e + k_2 \cdot y_e + k_0 \cdot \text{sgn}(y_e) \cdot \theta_e. \end{aligned} \quad (5)$$

where k_0, k_1, k_2 are positive constant parameters, x_e, y_e and θ_e are the trajectory-tracking errors defined in (3). If s_1 converges to zero, trivially x_e converges to zero. If s_2 converges to zero, in steady-state it becomes $\dot{y}_e = -k_2 \cdot y_e - k_0 \cdot \text{sgn}(y_e) \cdot \theta_e$. For $y_e < 0 \Rightarrow \dot{y}_e > 0$ if only if $k_0 < k_2 \cdot |y_e| / |\theta_e|$. For $y_e > 0 \Rightarrow \dot{y}_e < 0$ if only if $k_0 < k_2 \cdot |y_e| / |\theta_e|$. Finally, it can be known from s_2 that convergence of y_e and \dot{y}_e leads to convergence of θ_e to zero. Using the reaching law defined in (Gao and Hung, 1993)

$$\dot{s} = -Q \cdot s - P \cdot \text{sgn}(s) \quad (6)$$

$$\begin{aligned} Q &= \text{diag}[q_1, q_2], \quad q_i > 0, \\ P &= \text{diag}[p_1, p_2], \quad p_i > 0, \quad i = 1, 2 \\ \text{sgn}(s) &= [\text{sgn}(s_1), \text{sgn}(s_2)]^T \end{aligned}$$

together with (5), and after some mathematical manipulation, we get the commands for trajectory-tracking controller:

$$\begin{aligned} \dot{v}_c &= \frac{1}{\cos\theta_e} (-q_1 s_1 - p_1 \text{sgn}(s_1) - k_1 \dot{x}_e - \\ &\quad - y_e \dot{\omega}_d - \dot{y}_e \omega_d + v_r \dot{\theta}_e \sin\theta_e + \dot{v}_d). \end{aligned} \quad (7)$$

$$\begin{aligned} \omega_c &= \frac{1}{v_r \cos\theta_e + k_0 \text{sgn}(y_e)} (-q_2 s_2 - p_2 \text{sgn}(s_2) - \\ &\quad - k_2 \dot{y}_e - \dot{v}_r \sin\theta_e + x_e \dot{\omega}_d + \dot{x}_e \omega_d) + \omega_d. \end{aligned} \quad (8)$$

The SM-TT architecture with a on-line robot's pose estimator, fusing odometry with absolute position data, as described in (Lopes et al., 2007), is depicted in Fig. 4.

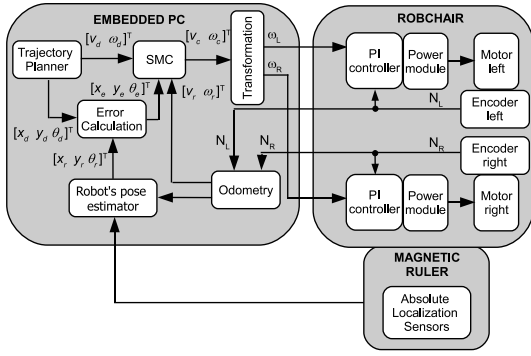


Figure 4: SM-TT control architecture with a robot's pose estimator, fusing odometry and absolute position data.

RobChair has a two-level control architecture (see Fig. 4). High-level control algorithms (including reference motion generation) are written in C and run with a sampling time of $T_s = 50$ ms on a embedded PC, which also provides a user interface with real-time visualization and a simulation environment. The PC communicates through a CAN bus with several devices. Wheel velocity commands,

$$\omega_R = \frac{v_c + \frac{L}{2} \cdot \omega_c}{R}, \quad \omega_L = \frac{v_c - \frac{L}{2} \cdot \omega_c}{R} \quad (9)$$

where R is radius of the drive wheels and L the distance between drive wheels, are sent to the PI controllers, and encoder measures N_R and N_L are received in the robot's pose estimator for odometric computations.

The low-level control layer is in charge of the execution of the wheels velocity control. For each wheel, a microcontroller implements a digital PI with a cycle time of $T_c = 5$ ms. Two power amplifiers drive the motors with PWM voltage.

3 HUMAN HEAD-NECK COMPLEX MODEL AND EVALUATION OF COMFORT

In general, comfort while riding depends not only on the amplitude, but also on the frequency of wheelchair vibrations and accelerations. Oscillations have influence on users comfort and may affect users health. Moreover, natural frequency of the wheelchair and human organ is strongly related with the user's un-comfort while riding.

3.1 Model of Head-neck Complex

A double-inverted pendulum model with two degrees-of-freedom is considered for the HNC model (Fig. 5).

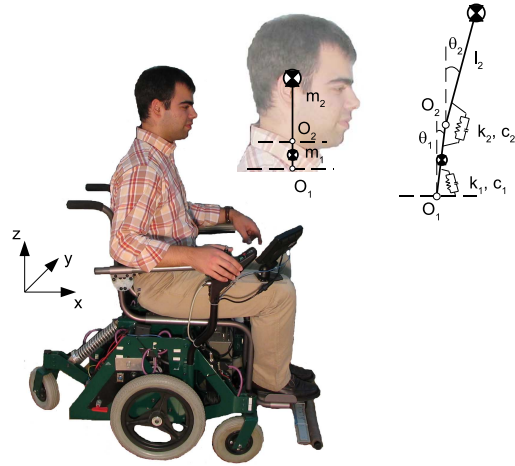


Figure 5: Human head-neck model.

One of the centers of rotation of the model was assumed to be at C7-T1 (O_1 in Fig. 5), and the other at C0-C1 (O_2 in Fig. 5) of the cervical spine. Two lumped masses, indicating the mass of the neck and the mass of the head, were considered in the model. The center of mass of the neck was assumed to be exactly at the mid-point of the two centers of rotation. Moreover, the center of mass of the head was assumed to be exactly over the center of mass of the neck and the center of rotation (Fig. 5). The equation of motion using generalized coordinates can be expressed as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + Kq + G(q) = Q_q \quad (10)$$

$$\text{where: } q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \quad Q_q = 0,$$

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

$$M_{11} = m_1 l_1^2 + m_2 L_1^2 + J_1$$

$$M_{12} = m_2 L_1 l_2 \cos(\theta_1 - \theta_2)$$

$$M_{21} = m_2 L_1 l_2 \cos(\theta_1 - \theta_2)$$

$$M_{22} = m_2 l_2^2 + J_2$$

$$C(q, \dot{q}) = \begin{bmatrix} c_1 + c_2 & C_{12} \\ C_{21} & c_2 \end{bmatrix}$$

$$C_{12} = m_2 L_1 l_2 \sin(\theta_1 - \theta_2) \dot{\theta}_2 - c_2,$$

$$C_{21} = -m_2 L_1 l_2 \sin(\theta_1 - \theta_2) \dot{\theta}_1 - c_2,$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} -(m_2 L_1 + m_1 l_1) \ddot{x} \cos(\theta_1) \\ -m_2 l_2 \ddot{x} \cos(\theta_2) \end{bmatrix}$$

In our study, the user characteristic elements, shown in Table 2 (from Atapourfard et al., 2002), were used to model the dynamic behavior of the human HNC.

Table 2: Characteristics of user's elements.

Element	Neck	Head
Length of segment $L_i[m]$	0.080	0.138
Center of gravity $l_i[m]$	0.040	0.069
Mass $m_i[kg]$	1.01	4.22
Moment of Inertia $J_i[kgm^2]$	0.0011	0.210
Spring constant $k_i[Nm/rad]$	14.04	10.29
Damping constant $c_i[Nms/rad]$	0.347	0.230

3.2 Time-domain Calculations

The fourth power vibration dose value (VDV) is primarily a measurement procedure used to report the relative severity of complex oscillation exposures, being preferred to other measures due to its use of the duration and variability of the motion:

$$VDV = \left\{ \int_0^T [a_w(t)]^4 \cdot dt \right\}^{\frac{1}{4}} \quad (11)$$

in units of $ms^{-1.75}$, where the frequency-weighted acceleration is defined by $a_w(t)$, and T is the duration of an experiment. VDV was selected over the difference between the peak accelerations and acceleration rms parameter. The problem with the peak-to-peak value is that it only represents one instance in time, rather than the entire signal. Furthermore, the peak-to-peak parameter does not accurately represent oscillatory motion. The problem with the rms value is that it is independent of the duration of the signal, and is designed to describe oscillatory motions.

According to the EU Directive on mechanical vibration (European-Parliament and the Council of the European Union, 2002) the average limit value of VDV is $9.1ms^{-1.75}$ and the upper limit is $21ms^{-1.75}$.

The transmissibility (T_r) is defined as the output VDV divided by the input VDV,

$$T_r = \frac{VDV_{output}}{VDV_{input}} \quad (12)$$

The transmissibility defines the performance of the wheelchair in terms of the amplification or attenuation of the vibration that is transmitted to the occupant. A value less than unity indicates that the accelerations were attenuated by the combination of wheelchair and human, whereas a value great than unity indicates an amplification of accelerations by the wheelchair-human system.

3.3 Frequency-domain Calculations

Given the input, wheelchair acceleration, and the output, acceleration obtained from the HNC model, the

transfer function is usually calculated using the cross-spectral density (CSD) method defined as:

$$H_{CSD}(f) = \frac{CSD_{input-output}(f)}{PSD_{input}(f)} \quad (13)$$

where $CSD_{input-output}(f)$ is the CSD of the input and output, and $PSD_{input}(f)$ is the power spectral density (PSD) of the input. The advantage of using the CSD method is that the function generates the phase of the response and also only includes data at the input and output that are correlated, thus reducing the effects of noise in the measurement system.

4 EXPERIMENTAL RESULTS

In order to validate the applicability of the sliding-mode controller for trajectory-tracking, real experiments have been performed using RobChair (intelligent platform developed in ISR-UC (Pires and Nunes, 2002), (Lopes et al., 2007)). RobChair, shown in Fig.6, has two differentially driven rear wheels and two passive castor front wheels. There is also a fifth rear wheel connected to the back of the wheelchair with a damper used for stability. It is powered by two 12-V batteries (60 Ah) and reaches a maximum speed of 7 Km/h. It has been equipped, in ISR-UC, with several devices such as: two power driver modules, which provide an independent control of each motor, optical encoders, laser range finders, an inertial sensor and a magnetic sensing ruler, developed at ISR-UC, that is able to perform a robust detection of magnetic markers (Lopes et al., 2007). Figure 7 presents a block diagram of the actual hardware control architecture. The current implementation of the framework is based on Linux as its underlying real-time operating system. The component-based software selected for the proposed software framework is GenoM (Generator of Modules) (Fleury et al., 1997), which is an environment for description and implementation of software components.

An embedded PC is responsible for giving some degree of intelligence to the robot. This computer is connected to distributed devices through fieldbuses. The platform is connected to external devices through a wireless link. This connection allows the implementation of a distributed architecture, which exhibits the possibility and capability to extend our single robot to other perspectives, like multi-robot cooperation, its integration in intelligent environments, etc.

All the distributed devices, connected through CAN, use a base printed circuit board, containing a microchip micro-controller (μC), as described in (Maia, 2004). A custom communication protocol,

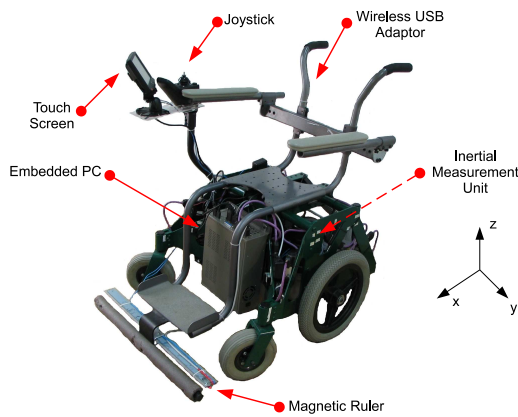


Figure 6: RobChair platform.

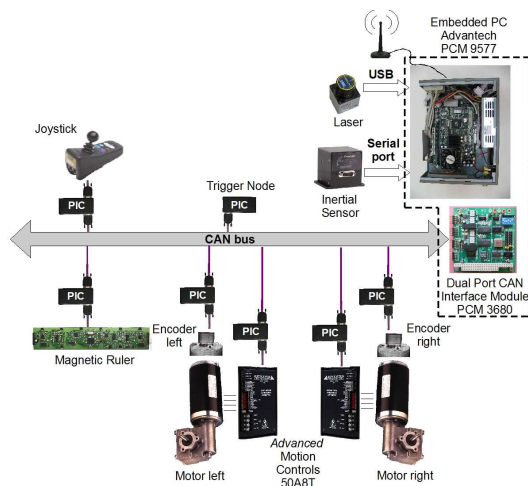


Figure 7: RobChair hardware architecture.

based on the time-triggered protocol paradigm, was designed and implemented. All events are synchronized by a message, sent from a Synchronization Micro-Controller Unit (Trigger Node, in Fig. 7), that synchronizes the other Micro-Controller Units, and defines the control loop time reference.

The odometric data provided by the wheel encoders is fused with the data from magnetic markers. The extended Kalman filter (EKF) was chosen for the fusion process (Bento et al., 2005). This navigation technology, based on sensing magnetic markers, is well suited when high precision navigation and robustness is required, and it can be used to complement other navigation systems, such as GPS.

The inertial sensor RGA300CA-100 (Crossbow) was used for measuring the wheelchair accelerations in three orthogonal directions.

Experimental results of the SM-TT controller using the planned path presented in Fig 1, are shown

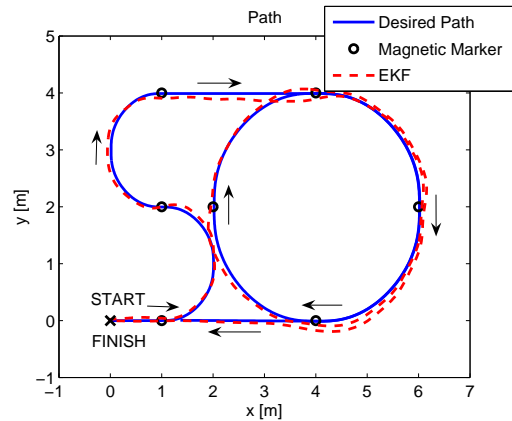


Figure 8: Experimental sliding-mode trajectory-tracking control using an EKF-based fusion in the on-line pose estimation.

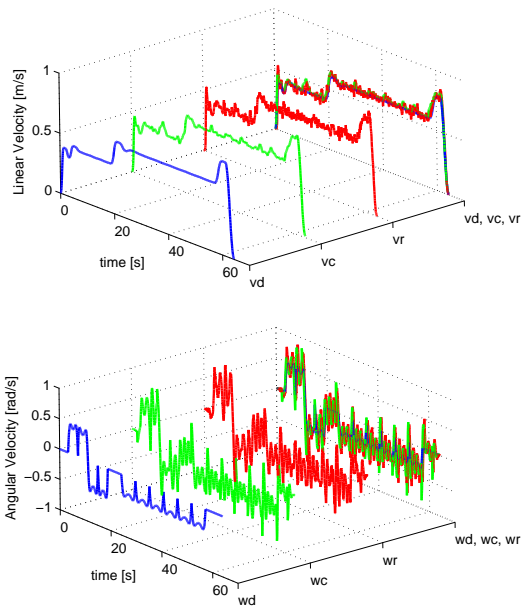


Figure 9: Desired (v_d , ω_d), command (v_c , ω_c) and real (v_r , ω_r) linear/angular velocities for SM-TT control under magnetic-markers navigation (case B).

in Figs. 8- 10. Figure 9 shows desired, command and real linear and angular velocities for SM-TT control under magnetic-markers navigation. Corrections in the pose after each magnetic marker detection provokes an error signal that is efficiently dealt by the SM-TT controller, and rapidly the tracking errors converge to zero (see Fig. 10).

The analysis of user comfort is made in three different situations:

- case A: SM-TT control under odometry navigation;

Table 3: Experimental Results.

Case	No	VDV_x	VDV_x	T_r	RMS accel.	Max. accel.	RMS accel.	Max. accel.
		RobChair	Head		RobChair	RobChair	Head	Head
A	I	2.9219	1.3826	0.4732	0.3287	1.5686	0.1656	0.7286
	II	2.7634	1.3429	0.4859	0.3191	1.4988	0.1589	0.7307
	III	2.7177	1.2802	0.4710	0.3187	1.5590	0.1510	0.8210
average		2.8010	1.3352	0.4767	0.3222	1.5421	0.1585	0.7601
B	I	2.9481	1.4408	0.4887	0.3538	1.2853	0.1797	0.7100
	II	3.2946	1.5707	0.4767	0.3663	1.7635	0.1856	0.7988
	III	3.0318	1.4860	0.4901	0.3629	1.4136	0.1828	0.7755
average		3.0915	1.4992	0.4852	0.3610	1.4875	0.1827	0.7614
C	-	11.4623	5.4607	0.4764	1.5185	4.1708	0.7039	2.2210

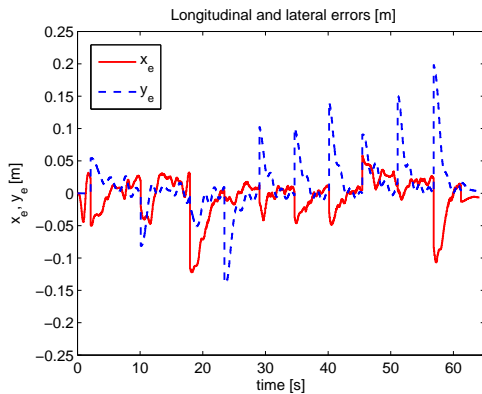


Figure 10: Longitudinal and lateral errors for SM-TT control under magnetic-markers navigation (case B).

- case B: SM-TT control under magnetic-markers navigation (odometric data is fused, using an EKF-based fusion, with absolute position data from magnetic markers detection);
- case C: SM-TT control with purposely-incorrectly-tuned parameters.

The experimental data of all three cases are summarized in Table 3. Three experimental trials were executed in cases A and B. The table shows the vibration dose value (VDV), transmissibility (T_r), root mean square accelerations (RMS) and maximum values (Max). The results of columns "RMS accel. Robchair" and "Max. accel. RobChair" concern the acceleration results obtained by the inertial sensor; and the "RMS accel. Head" and "Max. accel. Head" were obtained from the model of head-neck complex (10). The overall rms acceleration of head (along the x axes) in cases A and B are in range of "not uncomfortable", but in case C is in range of "uncomfortable" (see Table 1 and equation (4)).

Each experiment was made for the same trajectory (see Fig. 1). The time domain VDV values ob-

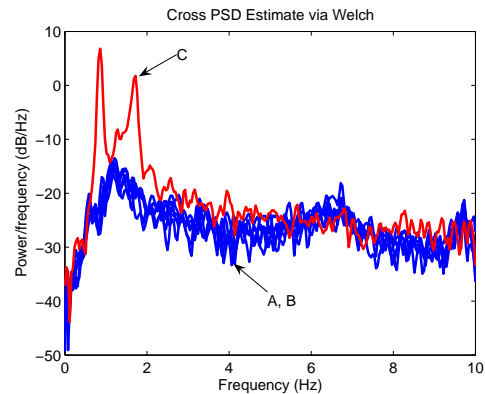


Figure 11: Cross-spectral density functions for all experiments.

tained in cases A and B are below the limit value of $9.1ms^{-1.75}$, only in case C, VDV values are above that limit. As can be observed from Table 3, the transmissibility tends to be under unity, suggesting that the vibrations are attenuated.

Figure 11 shows cross-spectral density values for all experiments. The maximum of H_{CSD} magnitude occurs in case C, and the corresponding frequencies are between $0.8 - 1.8Hz$. When the magnitude of H_{CSD} increases, the user comfort decreases.

5 CONCLUSIONS

Wheelchair is exposed to vibration coming not only from a variety of different road surface but also from the command of the wheelchair (manually - using Joystick or automated - using different type of controllers). The paper analyzes the comfort of wheelchair users when a SM-TT controller is used. The user comfort is examined not only in time domain (using the fourth power VDV), but also in frequency

domain (using the cross-spectral density method). Outdoor experiments, using RobChair with SM-TT controller were performed. The experimental tests presented in this paper are representative of the average performance of the controllers. We had summarized our acquired experience in general observations that can be useful guidelines for implementation of the same control strategies in other type of mobile robots.

ACKNOWLEDGEMENTS

This work was supported in part by ISR-Coimbra and Portuguese Technology and Science Foundation (FCT), under contract NCT04:POSC/EEA/SRI/58016/2004. R.Solea acknowledges a PhD research fellowship from FCT.

REFERENCES

- Atapourfard, M., Ishihara, T., and Inooka, H. (2002). The influences of trunk horizontal vibration to the head-neck complex. In *SICE02*, pages 1053–1058, Osaka.
- Atapourfard, M., Ishihara, T., and Inooka, H. (2004). Identification of the head-neck complex in response to trunk horizontal vibration. *Biological Cybernetics*, 90(6):418–426.
- Bento, L., Nunes, U., Moita, F., and Surrecio, A. (2005). Sensor fusion for precise autonomous vehicle navigation in outdoor semi-structured environments. In *8th IEEE International Conference on Intelligent Transportation Systems (ITSC'05)*, pages 245–250, Vienna.
- Chwa, D. (2004). Sliding-mode tracking control of non-holonomic wheeled mobile robots in polar coordinates. *IEEE Transactions on Control Systems Technology*, 12(4):637–644.
- Chwa, D., Hong, S., and Song, B. (2006). Robust posture stabilization of wheeled mobile robots in polar coordinates. In *17th International Symposium on Mathematical Theory of Networks and Systems*, pages 343–348, Kyoto.
- European-Parliament and the Council of the European Union (2002). *On the minimum health and safety requirements regarding the exposure of workers to the risks arising from physical agents (vibration)*. Official Journal of the European Communities, Directive 2002/44/EC.
- Fleury, S., Herrb, M., and Chatila, R. (1997). Genom: A tool for the specification and the implementation of operating modules in a distributed robot architectures. In *IEEE of the International Conference on Intelligent Robots and Systems*, pages 842–848, Grenoble.
- Gao, W. and Hung, J. (1993). Variable structure control of nonlinear systems: A new approach. *IEEE Transactions on Industrial Electronics*, 40(1):45–55.
- Gurses, S., Dhaher, Y., Hain, T., and Keshner, E. (2005). Perturbation parameters associated with nonlinear responses of the head at small amplitudes. *CHAOS*, 15(2):023905.
- ISO-2631 (1997). *Mechanical vibration and shock - Evaluation of human exposure to whole body vibration - Part 1: General requirements*. International Organization for Standardization.
- Kitazaki, S. and Griffin, M. (1997). A modal analysis of whole-body vertical vibration using a finite element model of the human body. *Journal of Sound and Vibration*, 200(1):83–103.
- Lopes, A. C., Moita, F., Nunes, U., and Solea, R. (2007). An outdoor guidpath navigation system for AMRs based on robust detection of magnetic markers. In *12th IEEE International Conference on Emerging Technologies and Factory Automation*, pages 989–996, Patras.
- Maia, R. (2004). Movimento de robots moveis com rodas de tracao diferencial: modelacao e controlo do sistema motriz. Master's thesis, Universidade de Coimbra (in portuguese), Coimbra.
- Pires, G. and Nunes, U. (2002). A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller. *Journal of Intelligent and Robotic Systems*, 34:301–314.
- Solea, R. and Nunes, U. (2007). Trajectory planning and sliding-mode control based trajectory-tracking for cybercars. *Integrated Computer-Aided Engineering, IOS Press*, 14(1):33–47.
- Solea, R., Nunes, U., and Filipescu, A. (2006). Trajectory planning and sliding-mode control for WMR trajectory-tracking and path-following respecting human comfort travel. In *7th Portuguese Conference on Automatic Control - CONTROLO'2006*, Lisbon.
- Yang, J. and Kim, J. (1999). Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 15(3):578–587.

VISUAL TRACKING ON THE GROUND

A Comparative Analysis

Jorge Raul Gomez, Jose J. Guerrero and Elias Herrero-Jaraba

Aragon Institute for Engineering Research, University of Zaragoza, Maria de Luna 1, Zaragoza, Spain

jrg@unizar.es, jguerrer@unizar.es, jeliass@unizar.es

Keywords: Tracking on the ground, Kalman filter, Homography.

Abstract: Tracking is an important field in visual surveillance systems. Trackers have been applied traditionally in the image, but a new concept of tracking has been used gradually, applying the tracking on the ground map of the surrounding area. The purpose of this article is to compare both alternatives and prove that this new usage makes possible to obtain a higher performance and a minimization of the projective effects. Moreover, it provides the concept of *multi-camera* as a new tool for mobile object tracking in surveillance scenes, because a common reference system can be defined without increasing complexity. An automatic camera re-calibration procedure is also proposed, which avoids some practical limitations of the approach.

1 INTRODUCTION

Real-time object tracking is recently becoming more and more important in the field of video analysis and processing. Applications like traffic control, user-computer interaction, on-line video processing and production and video surveillance need reliable and economically affordable video tracking tools. In the last years this topic has received an increasing attention by researchers. However, many of the key problems are still unsolved. The surveillance tracking community in particular has studied target tracking techniques for a number of years, mainly in the context of finding efficient methods to track missiles, aircrafts etc. and tracking targets of unknown motion. Their work has been used for a variety of applications.

There have been previous contributions in order to improve such systems. Tissainayagam and Suter (Tissainayagam and Suter, 2001) proposed a tracking method in which a model switching was used. Other authors, as Isler (Isler et al., 2005), use the technique of multiple or distributed sensors, assigning sensors to track targets so as to minimize the expected error in the resulting estimation for target locations. In most of the cases, the sensors used for these tasks are inherently limited, and individually incapable of estimating the target state, and they only can be used for a unique task. This limitation disappears with the use of cameras. In this way, Lee et al. (Lee et al., 2000) suggested to establish a common coordinate frame and to

capture image signals from several cameras arranged in a particular environment. This last idea is used in this paper in order to demonstrate that this solution contributes a better solution to the tracking problem.

We propose a simple tracker based on the Kalman Filter. This tracker is used in two different ways (on the image plane and on the ground plane), making a comparative between both. Theoretically, the perspective effects must disappear in the second one, and therefore, the tracking must involve better. This paper shows this event such in laboratory conditions as in real environments.

This paper is organized as follows: Section 2 briefly details the transformation between the image and the ground. After that an automatic re-calibration procedure that avoids some of the practical limitations of the approach is proposed. Section 3 provides details of the tracking on the floor, showing the different stages of the proposed tracking system: detection, tracking, uncertainty transformation, and tuning. In Section 4 we provide our results in a particular environment (our laboratory) under stable conditions. Finally, some results for a more complex environment (a football match) are shown also in section 4.

2 FROM IMAGE TO THE GROUND

In order to transform the coordinates from the image to the planar ground, a plane projection transformation is used. At the moment no distortion of the camera lens is assumed. A point in the projective plane is represented by three coordinates, $\mathbf{p} = (x_1, x_2, x_3)^T$, which represents a ray through the origin in the 3D space (Mundy and Zisserman, 1992). Only the direction of the ray is relevant, so all points written as $\lambda\mathbf{p} = (\lambda x_1, \lambda x_2, \lambda x_3)^T$ are equivalent. The classical Cartesian coordinates of the point (x, y) can be obtained intersecting the ray with a special plane perpendicular to x_3 axis and located at unit distance along x_3 . This is equivalent to scale \mathbf{p} as, $\mathbf{p} = (x, y, 1)^T$. Projected points in an image and real points in a planar ground are both represented in this way.

A projective transformation between two projective planes (1 and 2) can be represented by a linear transformation $\mathbf{p}_2 = \mathbf{T}_{21}\mathbf{p}_1$. If the transformation is represented in Cartesian coordinates it results non-linear. Since points and lines are dual in the projective plane, the transformation for the line coordinates is also linear, being $(\mathbf{T}_{21}^{-1})^T$ the corresponding transformation matrix for lines.

2.1 Computing the Transformation to Calibrate the Camera

Let it be $\mathbf{p}_c = (x_i, y_i, 1)^T$ the coordinates of a point i in the camera reference system. Let it be (x_i^g, y_i^g) the coordinates of the corresponding point in a reference system of the planar ground obtained from the plane of the building, and therefore let it be $\mathbf{p}_g = (x_i^g, y_i^g, 1)$ its homogeneous coordinates.

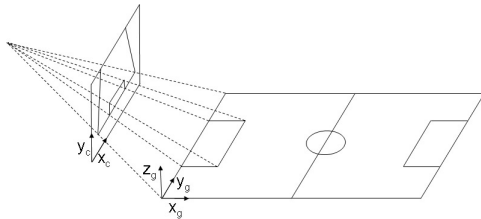


Figure 1: Diagram depicting the transformation of coordinates from image to the ground.

We obtain the projective transformation \mathbf{T}_{gc} up to a non-zero scale factor, for points, $\mathbf{p}_g = \mathbf{T}_{gc}\mathbf{p}_c$. For each couple i of corresponding points, two homogeneous equations to compute the projective transformation are considered. They can be written as, $(\lambda_i x_i^g, \lambda_i y_i^g, \lambda_i)^T = \mathbf{T}_{gc}(x_i, y_i, 1)^T$. Developing them in

function of the elements of the homography matrix, we have

$$\begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i^g & -y_i^g & -x_i^g \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i^g & -x_i^g & -y_i^g \end{pmatrix} \mathbf{t} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

where $\mathbf{t} = (t_{11} t_{12} t_{13} t_{21} t_{22} t_{23} t_{31} t_{32} t_{33})^T$ is a vector with the elements of the homography matrix \mathbf{T}_{gc} .

Using four pairs of corresponding points (no three of them being collinear), we can construct a 8×9 matrix \mathbf{M} , where $\mathbf{M}\mathbf{t} = \mathbf{0}$. Then, the solution \mathbf{t} corresponds with the eigenvector associated to the least eigenvalue (in this case the null eigenvalue) of the matrix $\mathbf{M}^T \mathbf{M}$, which can be easily solved by singular value decomposition (svd) of matrix \mathbf{M} . In order to have a reliable transformation, more than the minimum number of point correspondences must be considered, solving in a similar way (Hartley and Zisserman, 2000).

It is known that a previous normalization of data is suitable to avoid numerical computation problems (Hartley, 1997). We have transformed the coordinates of the points (in the image and in the ground) before the computation of the homography to reference systems located in the centroid of the points and scaled in such that the maximum distance of the points to its centroid is 1. After computation of the homography, it is inversely transformed by simple matrix computation to express the homography in the desired reference systems.

2.2 Automatic Camera Re-calibration

Once we have calibrated the camera using at least 4 pairs of corresponding points in the image and in the ground, it cannot be moved, which is the main limitation of this proposal. In practice, due for example to the flexibility of the camera support, the orientation of the camera changes. A little change of orientation has a great influence in the image coordinates of a point, and therefore invalidates previous calibration. However if the camera is not changed in position, or position change is small with respect to the depth of the observed scene, the homography can be re-calibrated automatically with high robustness and without 3D computations. As camera position changes suppose main reconfiguration of the surveillance system, but orientation changes are usual, the automatic re-calibration procedure presented below eliminates the limitation in practice. Besides that, this re-calibration procedure can also be used for changes in zoom lens or motions in pan-tilt cameras demanded by the user.

The re-calibration can be made using features extracted in the image like points and/or lines. We propose to do it using lines because they are plentiful in

man made environments and have other advantages. The straight lines have a simple mathematical representation, they can be extracted more accurately than points being also easier to match them and they can be used in cases where there are partial occlusions.

After extracting the lines, automatic computation of correspondences and homographies is carried out, as previously presented in (Guerrero and Sagüés, 2003), which uses robust estimation techniques. Thus, initially the extracted lines are matched to the weighted nearest neighbor using brightness-based and geometric-based image parameters.

With the coordinates of at least four pairs of corresponding lines we can obtain an homography that transforms both images. As usually we have many more than four line correspondences, an estimation method can be used to process all of them, getting better results. The least squares method assumes that all the measures can be interpreted with the same model, which makes it to be very sensitive to wrong correspondences. The solution is to use robust estimation techniques which detect the outliers in the computation. From the existing robust estimation methods, we have chosen the least median of squares method (Rousseeuw and Leroy, 1987).

In figure 2 we can see an example of two images before and after an unexpected camera motion. The automatic robust matching of lines that allows to compute the camera re-calibration has been superimposed to the images. The initial matching has about 20% of wrong correspondences, but the robust computation of the homography allows to reject wrong matches and also to search more matches according to it in a subsequent step. From the line correspondences the homography to recalibrate the camera is accurately obtained.

3 TRACKING ON THE GROUND

3.1 Detection and Tracking

A widely used technique for separating moving objects from their backgrounds is based on background subtraction (Herrero et al., 2003). In this approach, an image $I_B(x, y)$ of the background is stored before the introduction of a foreground object. Then, given an image $I(x, y)$ from a sequence, feature detection of moving objects are restricted to areas of $I(x, y)$ where:

$$|I(x, y) - I_B(x, y)| > \sigma \quad (1)$$

where σ is a suitable chosen noise threshold.

But this approach exhibits poor results in most real image sequences due to four main problems:

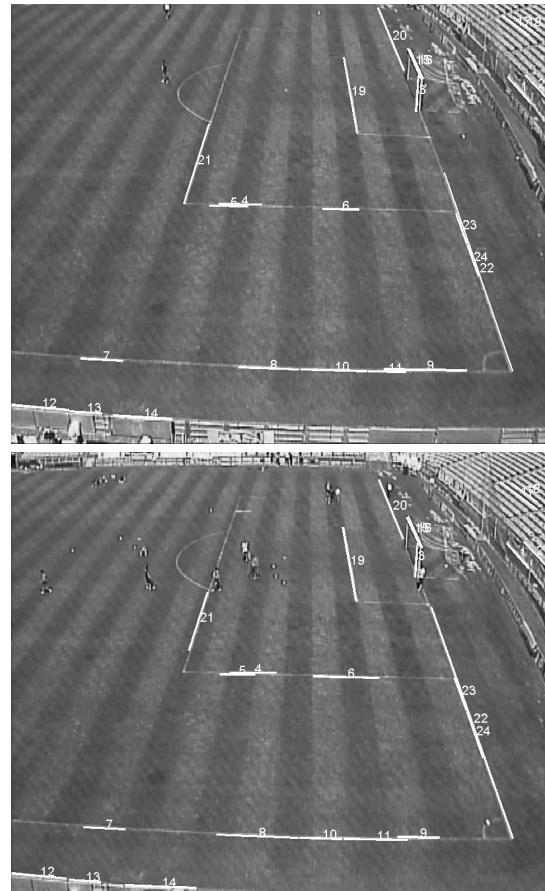


Figure 2: Two images of a football match before and after an unexpected camera motion in a real application. The automatic robust line correspondences has been superimposed to the images.

- Noise in the image.
- Gray level similarity between background and moving objects, even if the color is different.
- Continuous or quick illumination changes in the scene.
- Variation of the static objects in the background.

These problems can be partially solved by an appropriate selection of the threshold value. Some authors (Durucan and Ebrahimi, 2001) (Fabrice Moscheni and Kunt, 1998) have proposed a region-based motion segmentation using adaptive thresholding, according to illumination changes. In addition to this, morphological filters have to be used to eliminate noisy pixels and to fill the moving regions poorly segmented. However, in spite of these improvements, the results that can be found in real situations are far away from a satisfactory solution.

To detect the moving objects, we present an ap-

proach in motion detection, based on difference, introducing two procedures, *Neighborhood-Based Detection* and *Overlapping-Based Labelling*, in order to obtain a more robust segmentation in real scenes. The first one uses a local convolution mask, instead of a punctual one, to compute difference and obtain a more reliable difference image. The second one uses an overlapping criterion between two difference image to classify blobs in two different types: static or dynamic. This last characteristic makes a distinction between moving objects and shadows or illumination changes.

After the motion detection, a Kalman tracker (Kalman, 1960) with a constant velocity model (Bar-Shalom and Fortmann, 1988) is used for tracking, using the center of each detected object as measure data. Internally, the tracker has a state with 4 elements: 2 for the position and 2 for the velocity.

The Kalman filter is divided in two main parts: prediction and estimation. Between them, a matching procedure associates the measures obtained in the motion detection with the prediction of the tracking. It select the nearest-neighbor if it is close enough in function of the covariance of the innovation.

3.2 Uncertainty Transformation

The measure in the Kalman tracker is the position (x, y) of the mobile object. The measure noise has pixel units, but in order to do the tracking in the ground, it must be transformed according to the homography to metric units. To transform the covariance matrix from image to ground, as proposed in (A. Criminisi and Zisserman, 1997), a transformation in three steps is required: change to homogeneous coordinates, transformation of coordinates from image to ground and transformation to inhomogeneous coordinates again.

Given a covariance matrix, which express the uncertainty location in image coordinates:

$$\Lambda_{\mathbf{x}_c}^{2 \times 2} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix} \quad (2)$$

the correspondent homogeneous one is obtained:

$$\Lambda_{\mathbf{x}_c} = \begin{pmatrix} \Lambda_{\mathbf{x}_c}^{2 \times 2} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix} \quad (3)$$

The change of homogeneous coordinates from image to the ground is made with the homography matrix \mathbf{T}_{gc} . Therefore, the covariance matrix is transformed as:

$$\Lambda_{\mathbf{x}_g} = \mathbf{T}_{gc} \Lambda_{\mathbf{x}_c} \mathbf{T}_{gc}^\top \quad (4)$$

Once we have the uncertainty in the ground in homogeneous coordinates, we need to transform to non-

homogeneous coordinates in order to have the measurement. We will use ∇f as a first-order approximation of the relationship between homogeneous and inhomogeneous coordinates. If $\mathbf{X}_g = (X, Y, W)^\top$

$$\nabla f = 1/W^2 \begin{pmatrix} W & 0 & -X \\ 0 & W & -Y \end{pmatrix} \quad (5)$$

Therefore, the covariance matrix of the measurements noise in ground coordinates is

$$\Lambda_{\mathbf{x}_g}^{2 \times 2} = \nabla f \Lambda_{\mathbf{x}_c} \nabla f^\top \quad (6)$$

This transformation allows to have a noise model which considers the influence of the perspective effect when we made the tracking in the ground.

3.3 Tuning in Practice

The constant velocity model only can be considered locally valid. In practice, there are velocity changes that we model in the process noise. If we consider that the goal have an acceleration which is modelled as a white noise with zero mean and covariance q_i , the state noise matrix \mathbf{Q}_i for each coordinate $i = x_g, y_g$ is (Bar-Shalom and Fortmann, 1988):

$$\mathbf{Q}_i = q_i \cdot \begin{bmatrix} \frac{dt^4}{4} & \frac{dt^3}{2} \\ \frac{dt^3}{2} & dt^2 \end{bmatrix} \quad (7)$$

where dt is the time interval.

If we consider the tracker in the ground, the tuning has a well known meaning, because $\sqrt{q_i}$ represents directly the acceleration of the mobile. On the other hand, the classical tracker in the image needs an empirical tuning in pixel units, that depends of the perspective effect.

The *measure noise matrix* \mathbf{R} is defined in the image for both trackers and transformed to the ground using the homography matrix for the tracker on the ground, as seen in section 3.2.

4 EXPERIMENTS

4.1 Description

The objective of these experiments is to compare the performance of a tracker on the ground versus a tracker on the image. Two Kalman trackers will be compared, using the same constant velocity model, although each one may have a different, but equivalent, tuning, since coordinates in the image and coordinates in the ground represent different magnitudes.

The three first tests performed compare the precision of the predictions of both trackers in a sequence

of 550 frames, recorded with a still camera located in a corridor at 2.5 meters high. The target is a remote-controlled car moving at nearly constant velocity through a corridor, as it can be seen in fig. 3.

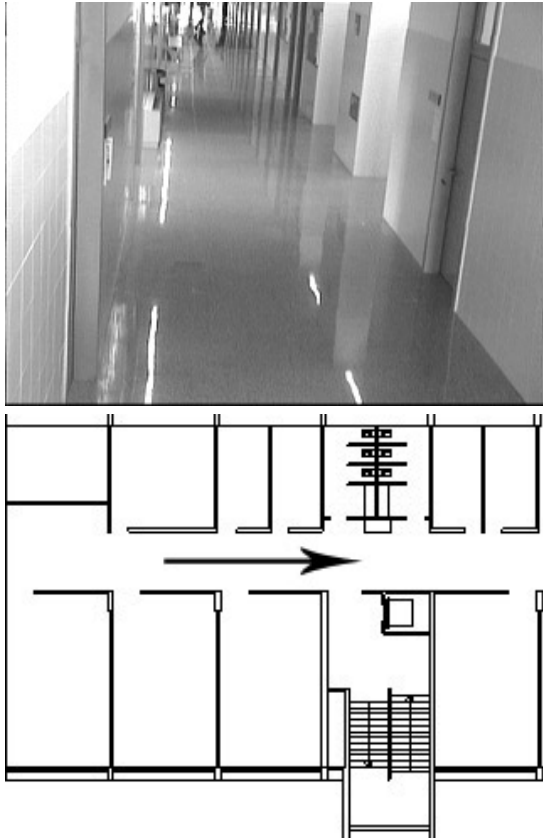


Figure 3: First image of the sequence (up) and building plane (down) where the tests have been performed. The remote-controlled car moves along the corridor in the tests, as the arrow shows.

After obtaining the position of the car in each frame, a tracker on the image has been applied. At the same time, the homography previously calculated, allows us to obtain the corresponding points in the ground for each point in the image, making possible to track the same object with another independent similar tracker on the ground.

To compare the performance of these two trackers, the mean difference between the prediction points and their corresponding measures has been computed. This difference is only taken into account if the measure is considered to belong to the object being tracked. Predictions from these two trackers are going to be compared at three levels: short, medium and long-term.

4.2 Comparative Analysis

Test 1: The first test compares the precision of short-term predictions. In the original sequence, the object is moving away the camera, with occasional lateral movements. The sequence will be tested also in reverse mode, starting from the last frame to the first, making the target go towards the camera. Mean distances obtained between prediction and measures, denoted as d_{pm} , are shown in table 1. All distances are measured in the ground, using Euclidean distance.

Table 1: Mean distances in mm. between predictions and measures for the image tracker and the ground tracker with the sequence processed forwards and backwards.

d_{pm}	Image	Ground
Forwards	54,90	18,89
Backwards	44,02	20,01

In this first test, a tracking on the ground has a great advantage over a tracking on the image, since the effect of the perspective deformation is avoided. Distances between predictions and measures are represented in figure 4.

As it can be seen in figure 4, a tracker on the ground obtains better results if the moving object is near the camera, because the velocity of the object in the image is more changeable. However, a tracker on the image obtains opposite results, since only measure noise causes this error. In any case, even when the moving object is far from the camera, the tracker on the ground obtains better results.

It must be noticed that two kinds of source of noise could be considered in the measure: one generated in the detection and other caused by the errors in the homography. At the moment, this second noise source has not been modelled, considering that its effect is small, because all the trajectories are inside the points used to compute the homography matrix in the calibration phase.

Test 2: To compare the accuracy of medium-term predictions, another test will be carried out, in the same sequence, consisting in decimating the number of measures. After the decimation, we will have $550/f$ measures, where f is the decimation factor. The same values for \mathbf{Q} and \mathbf{R} are used. Mean distance d_{pm} will be used again to measure the precision in each case. Obtained results can be seen in table 2, also in mm.

Mean distances grow as decimation factor increases, since the movement of the object is not totally predictable. The tracker on the image has an extra error originated by the deviation in velocity produced by the perspective effect.

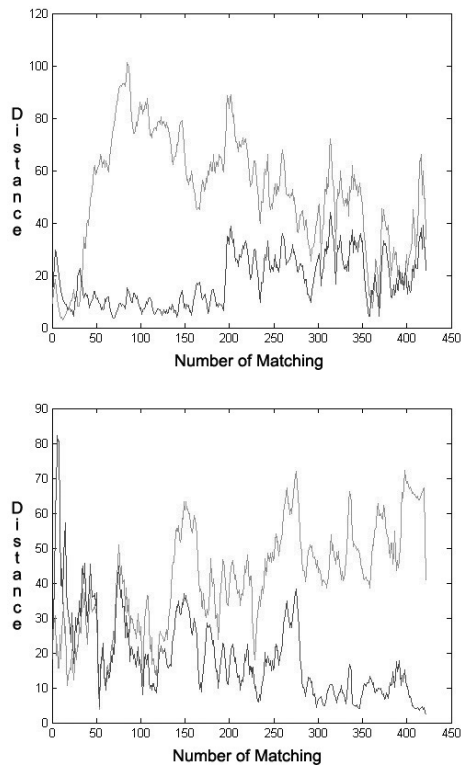


Figure 4: Distance between prediction and measure with tracking on the ground (dark line) and tracking on the image (light line). (a) Target moving away the camera (b) Target moving to the camera.

Table 2: Mean distances in mm. between prediction and measure applying different grades of decimation.

f	Image	Ground
1	54,9	18,89
2	78,49	22,89
3	97,75	26,58
5	132,01	31,06
8	180,02	40,33
12	248,08	55,93

Test 3: To test the precision of long-time predictions, a determined number of consecutive measures will be erased for both trackers. Also the same values of \mathbf{Q} and \mathbf{R} are used. This test evaluates the possibility of recovering the object after an occlusion.

Different numbers of measures have been erased in each test, from 2 to 100. To measure the accuracy of each tracker, the distance in the ground between the measure and the prediction after the erased block of measures has been used. The results can be seen in figure 5.

Here the differences between both trackers are higher. The ground tracker does not lost the measure

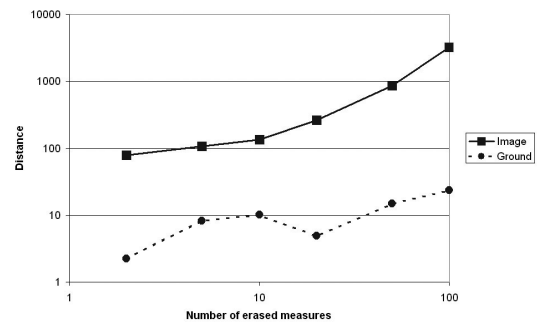


Figure 5: Distances obtained after erased blocks of measures of different lengths (in logarithmic scale).

even after an occlusion of about 100 frames, maintaining short distances between prediction and measure. However, the image tracker easily lost the measure after an occlusion of about 10 frames, giving very bad predictions.

4.3 Using in Practice

Once tested the superiority of the ground tracker in the laboratory, it has been confirmed in real-life videos of a football match. The test consists of a video sequence, as the frame in figure 6.a, where two football players are going to be tracked. Both players are running in parallel trajectories, but at different distances from the camera. In this test, the necessity of re-tuning of each trackers when trying to track different objects will be evaluated.

Both trackers will be configured with different equivalent tunings for the distant player. Using the homography, we can determine the relationship between a pixel in the image and the scale of the field to tune both trackers in a equivalent way. The measure noise \mathbf{R} is fixed, defined for the image tracker, and translated using the homography matrix for the ground tracker. This tuning will be used on the nearby player to check its validity.

The results can be seen on figure 7. As the two trajectories are not equivalent, and may have different accelerations and noises, the results cannot be directly compared. In any case, the figure shows that the number of matchings obtained for the distant player is similar for the two trackers, while the ground tracker obtains more matchings than the image tracker for the nearby player using the same tuning.

Although the number of matchings always can be increased with higher values of the the matrices \mathbf{R} and \mathbf{Q} , the prediction error and the possibility of crossing with other measures would increase as well. Hence, the application for a ground tracker is more important if multiple objects are attempted to be tracked.

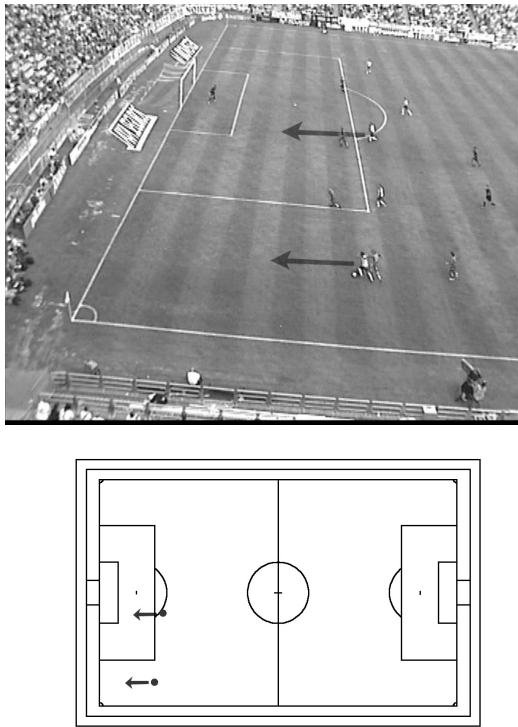


Figure 6: Frame of the video (up) and plane of the football field (down) of the video used. The arrows represent the direction of the players that have been used for the test.

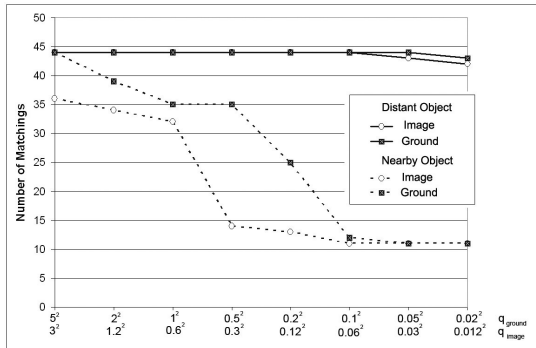


Figure 7: Number of matchings for different state noises. The two scales are equivalent for the position of the distant player.

5 CONCLUSIONS

In this paper we compared a tracker on the image versus a tracker on the ground. A plane projective transformation allows to make the tracking in real coordinates which facilitates the tuning of the tracker, gives measures in real coordinates and allows to relate different cameras in a common reference system. Experimental results from laboratory test and from

real environments proved empirically that the tracker on the ground achieves better results. We have also shown some preliminary results for the automatic recalibration of the camera which avoids some of the practical limitations of the approach. The continuation of this work will be focused to the usage of multiple cameras, having the plane of the surroundings as a common reference for the tracking.

ACKNOWLEDGEMENTS

This work was supported by project OTRI 2005/0388 "UZ - Real Zaragoza - DGA Collaboration Agreement for development of a research project in the sport performance improvement based on the image analysis", and it establishes the grounding for taking real measures and statistics over the ground.

REFERENCES

A. Criminisi, I. R. and Zisserman, A. (September 1997). A plane measuring device. In *IEEE Transactions on Pattern Analysis and Machine In Proc. BMVC, UK*.

Bar-Shalom, T. and Fortmann, T. (1988). *Tracking and Data Association*. Academic Press In.

Durucan, E. and Ebrahimi, T. (October 2001). Change detection and background extraction by linear algebra. In *Proceedings of the IEEE, 89(10):1368-1381*.

Fabrice Moscheni, S. B. and Kunt, M. (September 1998). Spatiotemporal segmentation based on region merging. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(9):897-915*.

Guerrero, J. and Sagüés, C. (2003). Robust line matching and estimate of homographies simultaneously. In *IbPRIA, Pattern Recognition and Image Analysis, LNCS 2652, 297-307*.

Hartley, R. (1997). In defense of the eight-point algorithm. In *IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(6):580-593*.

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge.

Herrero, E., Orrite, C., and Senar, J. (2003). Detected motion classification with a double-background and a neighborhood-based difference. In *Pattern Recognition Letters, 24:2079-2092*.

Isler, V., Khanna, S., Spletzer, J., and Taylor, C. J. (2005). Target tracking with distributed sensors: The focus of attention problem. In *Computer Vision and Image Understanding, 100, 225-247*.

Kalman, R. E. (1960). New approach to linear filtering and prediction problems. In *Transactions of the ASME—Journal of Basic Engineering, Volume 82, Series D, 35-45*.

- Lee, L., Romano, R., and Stein, G. (August 2000). Monitoring activities from multiple video streams: Establishing a common coordinate frame. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, n. 8.
- Mundy, J. and Zisserman, A. (1992). *Geometric Invariance in Computer Vision*. MIT Press, Boston.
- Rousseeuw, P. and Leroy, A. (1987). *Robust Regression and Outlier Detection*. John Wiley, New York.
- Tissainayagam, P. and Suter, D. (2001). Visual tracking with automatic motion model switching. In *Pattern Recognition*, 34, 641-660.

PARAMETER TUNING OF ROUTING PROTOCOLS TO IMPROVE THE PERFORMANCE OF MOBILE ROBOT TELEOPERATION VIA WIRELESS AD-HOC NETWORKS

Florian Zeiger, Nikolaus Kraemer and Klaus Schilling

*University of Wuerzburg, Department of Robotics and Telematics, Am Hubland, Wuerzburg, Germany
zeiger@informatik.uni-wuerzburg.de, kraemer@informatik.uni-wuerzburg.de, schi@informatik.uni-wuerzburg.de*

Keywords: Networked robots, mobile robot teleoperation, mobile robot ad-hoc networks, wireless network, ad-hoc network, wireless teleoperation, wireless remote control.

Abstract: Currently, the use of wireless networks is very common in the field of networked robotics and can be considered as a key issue for capable multi robot systems with a high grade of mobility. Nevertheless, this mobility requests for special features of the communication infrastructure, which leads to the integration of mobile robots into wireless ad-hoc networks. Since the late nineties, more than 80 ad-hoc routing protocols were developed and nowadays some of them are implemented and ready to use in real world applications. A comparison of four ad-hoc routing protocols (AODV, DSR, OLSR, and BATMAN) showed some shortfalls of the default parameter settings not allowing a reliable teleoperation of mobile robots while using AODV, OLSR, or BATMAN. This work is focused on the parameter tuning of the routing protocols to use them in wireless ad-hoc networks of mobile robots. The time required for route reestablishing, as well as the packet loss during rerouting is investigated in hardware tests of a network with dynamic network topology consisting of mobile robots. It could be demonstrated, that an appropriate parameter setting of OLSR and AODV allow the teleoperation of mobile robots in outdoor environments via a wireless ad-hoc network.

1 INTRODUCTION

Currently, more and more research is done in the field of teleoperation of mobile robot teams via wireless networks. As now a larger number of mobile robots are developed which are capable to operate in impassable or hazardous environments with little or no communication infrastructure, the communication infrastructure is set up by the robots itself on demand. Within these wireless ad-hoc networks, different types of nodes might be present: human workers or rescue personnel equipped with modern communication devices, mobile robots, or even some stationary nodes. All of them are able to act as data source, data sink, and communication relay and must support a dynamic network topology. In the field of networked robotics, several approaches are using wireless ad-hoc networks in many different areas of robot teleoperation. Multi-robot exploration with robots using wireless networks (Rooker and Birk, 2007) or a mobile robot team connected via wireless network which performed localization and control tasks (Das et al., 2002). Also in the field of rescue robotics (Rooker and Birk, 2005), or for integrating UAVs into IP based

ground networks (Zeiger et al., 2007), the use of wireless networks is quiet common nowadays.

An example for the network topology of these future scenarios is given in Figure 1. The network consists of several stationary nodes or ground stations and several mobile nodes which can be ground vehicles, aerial vehicles, or humans equipped with com-

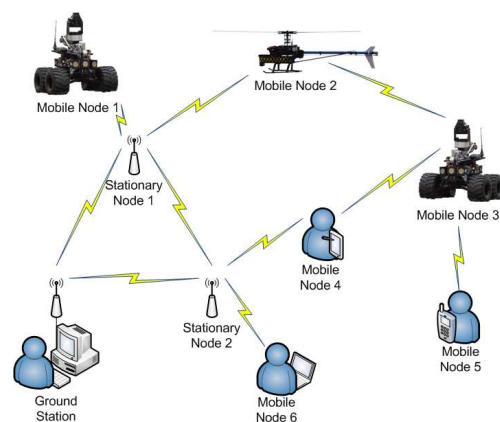


Figure 1: Future scenario of a heterogeneous network of mobile robots and human.

munication devices. All these nodes are connected by an ad-hoc wireless network which should guarantee a transparent any-to-any communication. Nevertheless, wireless communication always implies unpredictable communication delays, packet loss, or in worst case the loss of the link which makes the provision of the required quality a challenging task (Hsieh et al., 2006). To avoid the loss of communication, research focused on a dynamic setup of the required telecommunication infrastructure by placing relay nodes on demand (Nguyen et al., 2004)(Pezeshkian et al., 2007) or using mobile robots as relay nodes (Nguyen et al., 2003)(Pezeshkian et al., 2006). These approaches are using communication relays in wireless ad-hoc networks to setup communication networks with dynamic topologies. In these wireless networks no fixed infrastructure exists, and each mobile node not only works as host but also as router for data packets of other nodes. These dynamic topologies of wireless communication networks have advantages like providing direct and indirect any-to-any communication of each network node, redundant communication links in larger networks, no central administration, and a distribution of the traffic load in large networks. Of course, these advantages can only be used with rather complex and special routing protocols providing each node the necessary information about the network topology. The nodes itself are working as routers and must store the routing information of the complete network locally. In the field of wireless telecommunication, more than 80 ad-hoc routing protocols for wireless networks were developed (Johnson and Maltz, 1996)(Redi and Welsh, 1999)(Das et al., 2003)(Chakeres and Belding-Royer, 2004). Also some simulations for performance evaluations for larger scale telecommunication networks were done in the past (Broch et al., 1998)(Das et al., 2001)(Dyer and Boppana, 2001). (Johansson et al., 1999) compared several ad-hoc routing protocols in a simulation study and (Kiess and Mauve, 2007) gives a survey of currently existing real-world implementations of ad-hoc routing protocols, including some real-world scenario tests.

This work is based on the results of a former publication (Zeiger et al., 2008), which compares several ad-hoc routing protocols with respect to mobile robot teleoperation. The standard parameter settings of the routing protocols AODV, OLSR, DSR, and BATMAN were investigated. Unfortunately, only DSR showed to be an appropriate solution for mobile robot teleoperation and the performance other three routing protocols had to be improved by parameter tuning. This work is focused on the parameter tuning of OLSR, AODV, and BATMAN.

The objective is an acceptable packet loss and time for rerouting in a highly dynamic network topology. Therefore, existing protocol implementations of AODV and DSR (<http://core.it.uu.se/core/index.php>) from Uppsala University and the University of Basel, OLSR (<http://www.olsr.org>) and BATMAN (<https://www.open-mesh.net/batman>) are used in real-world test scenarios where mobile robots are teleoperated in an outdoor environment.

The presented work is structured as follows. In Section 2, the used hardware and the real-world test scenarios are described. Section 3 gives a brief summary of a comparison of AODV, DRS, OLSR, and BATMAN with respect to mobile robot teleoperation with standard parameter settings. The results of this work – the parameter tuning of ad-hoc routing protocols to enable mobile robot teleoperation via wireless ad-hoc networks – are presented in Section 4. A conclusion is given in Section 5.

2 HARDWARE AND TEST SETUP

2.1 Hardware

For the performed tests, several mobile nodes were used. One node is a PC for the operator. Up to 4 MERLIN robots (standard version) were used as stationary communication relay nodes, and one Outdoor MERLIN was used (cf. Figure 2) (Eck et al., 2007).



Figure 2: The Teleoperated OutdoorMERLIN Robot.

All MERLIN robots have a C167 micro controller for low-level operations and sensor data processing, as well as a PC-104 for more complex and computationally more intensive tasks. The PC-104 uses a Linux operating system and all nodes are equipped with 802.11b standard WLAN equipment (Atheros chip). More information on the MERLIN robots is given at www7.informatik.uni-wuerzburg.de. For steering the mobile robot, the operator's PC is running an application which generates command pack-

ets of a size between 6 and 22 bytes of payload. These packets are sent via UDP over the wireless network to the mobile robot. The onboard software of the mobile robot generates a UDP packet stream of packets with variable size containing the sensor data.

2.2 Test Setup

The scenario is set up in a way that the rerouting procedure will start with the mobile robot being at a certain location. Therefore, a large building is used as obstacle (cf. Figure 3). Relay nodes are placed at the corners of the building, such that they have always the neighbor nodes at the next and previous corner of the building within their communication range. As soon as the mobile robot is moved out of the line-of-sight of one node, the rerouting procedure is initialized. This scenario represents a worst case in terms of link redundancy, as only one route between operator PC and mobile robot is available. Relevant measurement categories are the packet loss and the duration of a communication drop-out during rerouting.

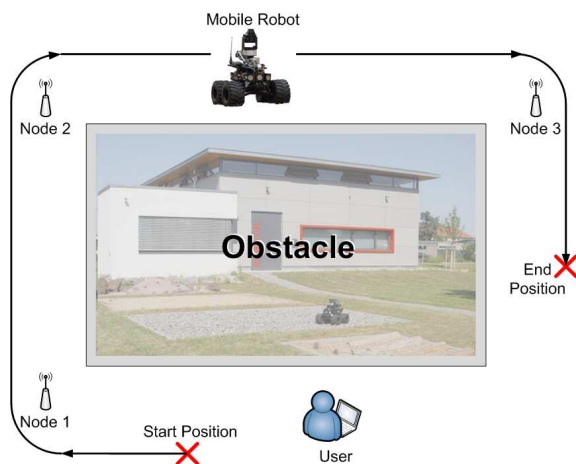


Figure 3: Test Setup.

3 AD-HOC ROUTING & TELEOPERATION

3.1 Investigated Protocols

This work investigates the parameter settings for different ad-hoc routing protocols: Ad-hoc On-demand Distance Vector (AODV), Dynamic Source Routing (DSR), and Optimized Link State routing (OLSR).

AODV (Das et al., 2003) (Chakeres and Belding-Royer, 2004) is a reactive routing protocol and determines required routes on-demand. To discover a

route to an unknown destination, a Route Request (RREQ) message is broadcasted. Each intermediate node which is not the destination and without a route to the destination receiving a RREQ broadcasts it further. In case the RREQ is received more than once, only the first reception will result in a broadcast. To avoid uncontrolled dissemination of RREQs, each has a certain time to live (TTL) after which it is discarded. When the destination receives a RREQ message a Route Reply (RREP) message is generated and sent back to the source in unicast hop by hop fashion along the route which was determined by the RREQ message. After generating a RREP message, the RREQ message is discarded at this node. As the RREP propagates, each intermediate node creates a route to the destination. After the source receives the RREP, it records the route to the destination and begins sending data. In case the source receives multiple RREPs, the route with the shortest hop count is chosen. The status of each route is maintained in the local routing table and timers are used to determine link failures. In case a certain node is part of an active route, Hello messages are used to obtain the route status. These Hello messages are broadcasted periodically to all neighbors. If a neighbor does not send a Hello message within a specified time a link loss is detected and the node is deleted from the routing table. In addition, a Route Error message (RRER) is generated. More detailed information on AODV is given in (Das et al., 2003). In the test scenarios of this work, AODV-UU version 0.9.5 from Uppsala University (Sweden) is used (<http://core.it.uu.se/core/index.php/AODV-UU>).

DSR is also a reactive ad-hoc routing protocol which works similar to AODV but without using Hello messages for route maintenance. However, it uses source routing (DARPA Internet Program, 1981). DSR does not use any periodic routing advertisement, link status sensing, or neighbor detection packets, and does not rely on these functions from any underlying protocols in the network. DSR is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes in the ad-hoc network. In case source node (S) wants to send data to an unknown destination host (D), S initiates the route discovery mechanism. S broadcasts a route request message which identifies the source and destination of the route discovery to all neighbors. A route request also contains a record listing the address of each intermediate node which forwarded this particular copy of the route request. A node which receives this route request without being the destination looks up for a source route to the requested destination in its own route cache. Without any source route present in its route cache, the node

appends its own address to the route record and broadcasts the route request message. In case this request message was received more than once, it is simply discarded. As soon as the route request message arrives at the desired destination D , a route reply message to S is created which contains an accumulated route record of the route request. After S receives this route reply, it caches the corresponding route in its route cache and S is ready to transmit data. Of course, there exist mechanisms to omit flooding of the network with route requests. A hop limit was introduced and every time a route request is forwarded, the hop limit is decremented by one. As soon as it reaches zero, the request is discarded. Also mechanisms for avoiding infinite recursion of route discoveries are implemented. A more detailed description of this protocol is given in (Johnson and Maltz, 1996) (Hu et al., 2004). The presented work uses DSR-UU version 0.2 from Uppsala University (Sweden) (<http://core.it.uu.se/core/index.php/DSR-UU>).

OLSR is a table-driven pro-active routing protocol for mobile ad-hoc networks. It uses hop-by-hop routing (each node uses its local information to route packets). OLSR minimizes the overhead from flooding of control traffic by using only selected nodes called Multipoint Relays (MPR) to retransmit control messages. Each node in the network selects a set of nodes in its neighborhood, which may retransmit its messages. This set of selected neighbor nodes is called the MPR set of that node. The neighbors of node N which are not in its MPR set, receive and process broadcast messages but will never retransmit broadcast messages received from node N . The MPR set is selected such, that every node in the 2-hop neighborhood of N has a link to the MPRs of N . OLSR continuously maintains routes to all destinations in the network by distributing link and neighborhood information (periodically exchange Hello messages). These messages are also used for link sensing and for checking the connectivity. More details on OLSR are given in (Clausen, 2003). The scenario tests in the present work are performed with OLSR version 0.5.3 (<http://www.olsr.org/index.cgi?action=download>).

BATMAN (Better approach to mobile ad-hoc networking) is a new approach to ad-hoc routing. Unlike other algorithms that exist right now, BATMAN does not calculate routes. It continuously detects and maintains the routes by receiving and broadcasting packets from other nodes. Instead of discovering the complete route to a destination node, BATMAN only identifies the best single-hop neighbor and sends a message to this neighbor. These messages contain the source address, a sequence number, and a time-to-live (TTL)

value that is decremented by 1 every time before the packet is broadcasted. A message with a TTL value of zero is dropped. The sequence number of these messages is of particular importance for the BATMAN algorithm. As a source numbers its messages, each node knows whether a message is received the first time or repeatedly. More details on BATMAN are given in (B.A.T.M.A.N. (better approach to mobile ad-hoc networking), 2007). In the test scenarios of the presented work, BATMAN version 0.2 is used (<https://www.open-mesh.net/batman>).

3.2 Rerouting Time and Packet Loss with Standard Parameter Settings

In (Zeiger et al., 2008), four different ad-hoc routing protocols were investigated with respect to mobile robot tele-operation. A mobile robot was commanded in a test scenario which forced the routing protocols to increase the number of participating nodes in the communication link while the robot was moved around an obstacle. The four compared ad-hoc routing protocols were all used with the standard parameter settings and behaved quite different. While it was

Table 1: Packet Loss & Times for Route Reestablishing from (Zeiger et al., 2008).

Protocol	Packet loss during test run	Time for re-routing	
		min.	max.
AODV	29.2%	2.1s	> 30s
OLSR	14.2%	10.1s	> 30s
DSR	11.2%	2.4s	2.7s
BATMAN	conn. lost	–	–

not possible to accomplish the scenario with BATMAN, the other protocols at least allowed a teleoperation of the mobile robot – often with only very limited performance. AODV was originally designed for highly dynamic networks. Routes are established on demand. In some cases this rerouting took only a very short time (cf. Table 1), but sometimes, the communication drop-out duration was longer than 30 seconds. This is by far too long for the telecommand of a mobile robot. Compared to AODV, the minimum of the required re-routing time, OLSR is slower. Rarely, also communication drop outs were observed. With only half of the packet loss, OLSR showed a slightly better performance as AODV. Although OLSR worked more reliable than BATMAN or AODV, the observed minimum time for re-routing of 10.1 seconds is quiet high with respect to teleoperation and will not be appropriate for any kind of control via this network. DSR showed to be the most reliable and the fastest proto-

col which was tested. A packet loss of about 11% and a re-routing time between 2.4 and 2.7 seconds make this protocol suitable for reliable telecommand of a mobile robot. With respect to the test scenario, it was expected that DSR performs best, as only one node (the robot) is mobile and all other nodes are stationary. Here, DSR discovers the topology quiet fast and only the changes due to the robots movement result in routing messages. The used test scenario also represents a worst case in the means of route redundancy due to the availability of only one possible route between controller and mobile robot. This could be the reason for the relatively poor performance of AODV and OLSR. Originally, these protocols were developed to handle much larger networks with higher node mobility and a higher grade of meshing as in the current test scenario. Nevertheless, the presented test scenario is quiet typical with respect to teleoperation. In Table 1, also the average packet loss during route reestablishing is given. Again DSR showed the best performance (11.2% packet loss) compared to OLSR (14.2% packet loss) and AODV (29.2% packet loss). BATMAN was not able to establish a new route via additional relay nodes.

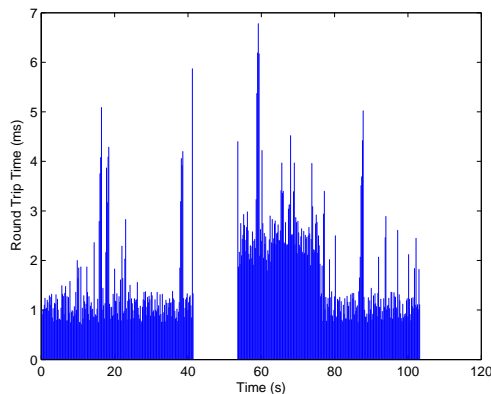


Figure 4: Example for OLSR round trip times for scenario 1 with default parameter setting.

Figure 4 displays an example of the round trip times for test scenario 1 and shows a clear communication drop out between 40 and 50 seconds test time. In Figure 5 the behavior of DSR in the same scenario is showed. Here, the communication drop-outs were significantly shorter. The compared ad-hoc routing protocols followed different principles (e.g. proactive vs. reactive) and were designed by different inspirations and for different application areas. Thus, they have also different parameters with different default settings, too. Nevertheless, this investigation showed that a better performance is required for mobile robot teleoperation. Based on these experiments, protocol

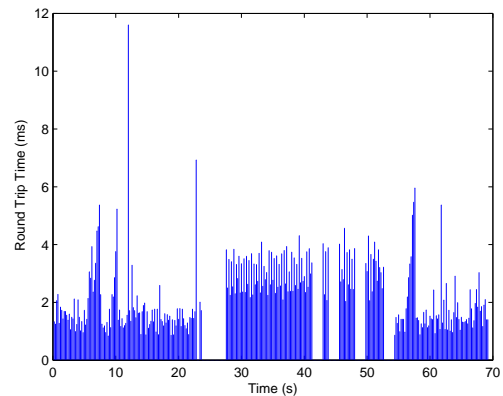


Figure 5: Example for DSR round trip times for scenario 1 with default parameter setting.

parameters are identified and tuned in order to increase the quality of the communication. The results are presented in the following sections.

4 PARAMETER TUNING

4.1 Variable Protocol Parameters

This section summarizes the default parameter settings of the investigated ad-hoc routing protocols OLSR (cf. Table 2), AODV, and BATMAN. In (Zeiger et al., 2008), these settings were chosen as a trade-off between generated routing overhead and fast topology discovery. Unfortunately, these parameter settings for OLSR, AODV, and BATMAN showed to be not suitable to use these ad-hoc routing protocols for mobile robot tele-operation.

Table 2: Variable parameters for OLSR.

Name:	Default Value
Willingness	dyn. calc.
LinkQualityLevel	2
LinkQualityWinSize	10
Pollrate	0.05 sec
TcRedundancy	0
HelloInterval	2 sec
HelloValidity	6.0 sec
TCInterval	5 sec
TCValidity	15.0 sec

For AODV, relevant parameters are "force gratuitous", "local repair", and "no wait on reboot" which are disabled by default. The BATMAN protocol offers less possibilities for parameter changes. In the

following sections, "originator interval" will be used for optimization. By default, "originator interval" is set to 1000 milliseconds.

4.2 OLSR

To increase the performance of OLSR in order to use it for mobile robot teleoperation, the four parameters were changed as shown in Table 3. As the parameter settings are interdependent, some simple rules must be followed as not all combinations of values are useful. The hello- and tc-intervals have to be smaller than half the corresponding validity times. Of course, the traffic for routing will be increased due to more hello and tc-messages but will not cause decrease the throughput significantly.

Table 3: Tuned parameters for OLSR.

Name:	Default Value:	New Value:
HelloInterval	2 sec	0.5 sec
HelloValidity	6 sec	1.5 sec
TcInterval	5 sec	2.5 sec
TcValidityTime	15 sec.	5.0 sec

The rerouting time was reduced from more than 10.1 to an average value of 5.96 seconds (with a minimum of 5.2 and a maximum of 7.4) for including the first relay node into the communication link. The packet loss during rerouting was reduced from 14.2% to 4%. Figure 6 shows an example of the behaviour of the round trip time during a rerouting process with OLSR. As the rerouting is initiated at 32.4 seconds, a short communication drop out occurred until the link is reestablished at 37.2 seconds simulation time. Thus, the OLSR performance could be increased but still, the observed performance of DSR during the comparison in (Zeiger et al., 2008) could not be reached.

4.3 BATMAN

As already mentioned, the number of variable parameters for BATMAN is very small. A suitable parameter to be tuned is the "originator interval", which is the time to wait sending one message and before the batman daemon sends the next message (default value is 1000 milliseconds). In (Zeiger et al., 2008), all BATMAN test runs ended up in a communication loss. Now, the value for "originator interval" is decreased which should result in a faster response on topology changes by the routing protocol. Figure 7 shows an Example of a test run with "originator interval" set to 125 ms. Of course, this setting increases the routing

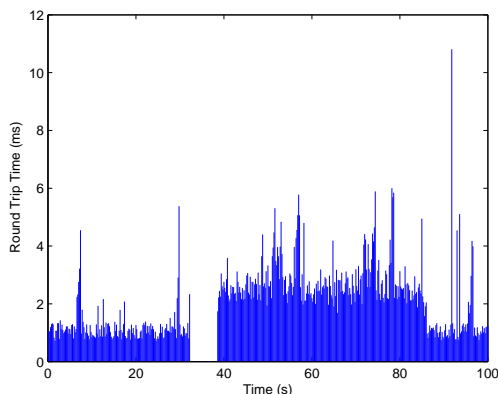


Figure 6: Example for OLSR round trip times for scenario 1 with HelloInterval=0.5, HelloValidity=1.5, TCInterval=2.5 and TCValidity=5.0.

overhead but as a result, BATMAN now can handle the rerouting without losing the communication. During the tests, the average rerouting duration is 7.78 seconds with a packet loss of 8%. This parameter setting significantly improved the performance of BATMAN but still, the duration of the rerouting procedure takes too long to be used for mobile robot teleoperation.

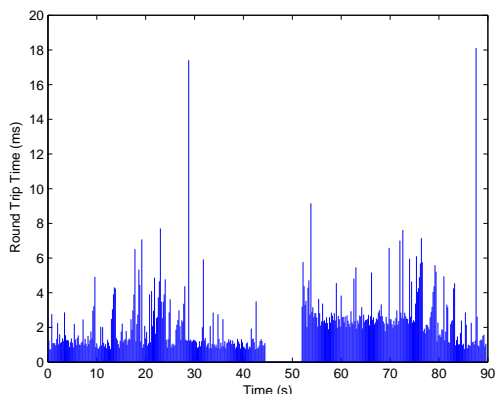


Figure 7: Example for BATMAN with "originator interval" set to 125 ms.

4.4 AODV

For AODV, it is possible to tune several parameters. The performed tests showed, that only an combined change of several parameters might have the opportunity to show a positive effect with respect to teleoperation. Unfortunately, it was not possible to find a suitable parameter setting – all combinations made the protocol behaving very unstable. For example, enabling "force gratuitous", "local repair", "no wait on reboot", and setting "treat node as neighbor" to 2 (cf.

Figure 8) lead to many communication losses.

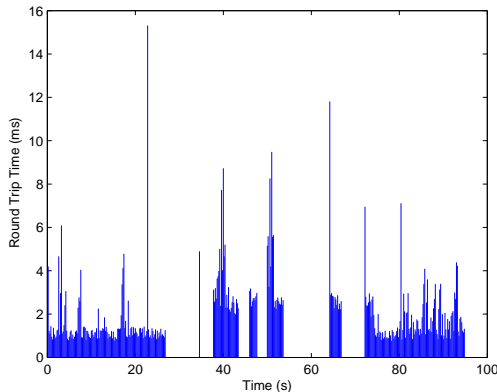


Figure 8: Example for AODV with "force gratuitous", "local repair", and "no wait on reboot" enabled and "treat node as neighbor" = 2.

5 CONCLUSIONS

This work uses results of a comparison of four ad-hoc routing protocols as a basis where the default parameter setting showed to be not appropriate for mobile robot teleoperation. Here, results of a study for parameter tuning of real implementations of the ad-hoc routing protocols OLSR, AODV, and BATMAN are presented. Real hardware tests of a mobile robots teleoperation scenarios were performed and the behavior of the communication link was analyzed with respect to mobile robot teleoperation. It is shown, that a tuning of a combination of relevant parameters for OLSR can improve its performance. Also the reduction of originator interval" of BATMAN results in a more reliable network than experienced with the default parameter settings. For AODV, sometimes a better reaction in terms of required time for rerouting could be observed. The evaluated protocol implementations are suitable for some teleoperation approaches – e.g. systems with local autonomy functions to ensure a defined behavior of the mobile robot during the rerouting times. Nevertheless, the behavior of all tested ad-hoc routing implementations was less than expected. The previously mentioned simulation results showed much shorter rerouting times (in a magnitude of some milliseconds). These short rerouting times were never observed in the presented real hardware tests. Future work will now be focused on a detailed analysis of the differences between the simulation studies and the effects described in this work to further improvement the usability of AODV, OLSR, DSR, and BATMAN for mobile robot teleoperation or remote control approaches via wireless multi-hop

networks. In addition, the required interaction between applications (e.g. teleoperation interfaces or control algorithms) and the network status via the lower protocol layers must be analyzed.

REFERENCES

- B.A.T.M.A.N. (better approach to mobile ad-hoc networking) (2007). <https://www.open-mesh.net/batman>, visited 2007/09/07.
- Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., and Jetcheva, J. (1998). A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Mobile Computing and Networking*, pages 85–97.
- Chakeres, I. D. and Belding-Royer, E. M. (2004). AODV Routing Protocol Implementation Design. In *Proceedings of the International Workshop on Wireless Ad hoc Networking (WWAN), Tokyo, Japan*, pages pp. 698–703.
- Clausen, P. J. T. (2003). RFC 3626, Optimized link state routing protocol (OLSR). IETF, Network Working Group.
- DARPA Internet Program (1981). RFC 791, Internet Protocol Specification.
- Das, A., Spletzer, J., Kumar, V., and Taylor, C. (2002). Ad Hoc Networks for Localization and Control. In *Proceedings of the 41st IEEE Conference on Decision and Control, (CDC 2002)*, volume 3, pages 2978–2983.
- Das, S., Perkins, C. E., and Belding-Royer, E. M. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. IETF RFC 3561.
- Das, S. R., Perkins, C. E., Royer, E. M., and Marina, M. K. (2001). Performance Comparison of Two On-demand Routing Protocols for Ad hoc Networks. *IEEE Personal Communications Magazine special issue on Ad hoc Networking*, February 2001:16–28.
- Dyer, T. D. and Boppana, R. V. (2001). A comparison of TCP performance over three routing protocols for mobile ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Long Beach, CA, USA*, pages 56–66.
- Eck, D., Stahl, M., and Schilling, K. (2007). The Small Outdoor Rover MERLIN and its Assistance System for Tele-Operations. In *Proceedings of International Conference on Field and Service Robotics (FSR), Chamonix (France)*.
- Hsieh, M. A., Cowley, A., Kumar, V., and Taylor, C. (2006). Towards the deployment of a mobile robot network with end-to-end performance guarantees. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'06), Orlando, FL (USA)*, pages 2085–2090.
- Hu, Y.-C., Johnson, D. B., and Maltz, D. A. (2004). The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. IETF MANETWorking Group, IETF RFC 4728.

- Johansson, P., Larsson, T., Hedman, N., Mielczarek, B., and Degermark, M. (1999). Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In *Proceedings of Mobicom'99, Seattle, Washington*.
- Johnson, D. B. and Maltz, D. A. (1996). Dynamic Source Routing in Ad-Hoc Wireless Networks. *Mobile Computing*, 353.
- Kiess, W. and Mauve, M. (2007). A Survey on Real-World Implementations of Mobile Ad-Hoc Networks. *Elsevier's Ad Hoc Networks*, 5(3):324–339.
- Nguyen, H. G., Pezeshkian, N., Gupta, A., and Farrington, N. (2004). Maintaining Communication Link for a Robot Operating in a Hazardous Environment. In *Proceedings of ANS 10th International Conference on Robotics and Remote Systems for Hazardous Environments, Gainesville, Florida, March 28-31*.
- Nguyen, H. G., Pezeshkian, N., Raymond, M., Gupta, A., and Spector, J. M. (2003). Autonomous Communication Relays for Tactical Robots. In *The 11th International Conference on Advanced Robotics, Proceedings of ICAR 2003, Coimbra, Portugal, June 30 - July 3*.
- Pezeshkian, N., Nguyen, H. G., and Burmeister, A. (2006). Unmanned Ground Vehicle Non-Line-of-Sight Operations Using Relaying Radios. In *Proceedings of the 12th IASTED International Conference on Robotics and Applications, Honolulu, Hawaii (USA)*.
- Pezeshkian, N., Nguyen, H. G., and Burmeister, A. (2007). Unmanned Ground Vehicle Radio Relay Deployment System for Non-Line-Of-Sight Operations. In *Proceedings of the 13th IASTED International Conference on Robotics and Applications, Wuerzburg (Germany)*.
- Redi, J. and Welsh, R. (1999). Energy-Conservation for Tactical Mobile Robots. In *Proceedings of Military Communications Conference, MILCOM*.
- Rooker, M. N. and Birk, A. (2005). Combining Exploration and Ad-Hoc Networking in RoboCup Rescue. In Nardi, D., Riedmiller, M., and Sammut, C., editors, *RoboCup 2004: Robot Soccer World Cup VIII*, volume 3276 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 236–246. Springer.
- Rooker, M. N. and Birk, A. (2007). Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445.
- Zeiger, F., Kraemer, N., and Schilling, K. (2008). Commanding Mobile Robots via Wireless Ad-Hoc Networks - A Comparison of Four Ad-Hoc Routing Protocol Implementations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'08)*.
- Zeiger, F., Selbach, C., Ruderisch, B., and Schilling, K. (2007). An Application Protocol to Integrate a Small Size Helicopter into an IP based Ad-Hoc Network. In *Proceedings of ROBOCOMM 2007, Athens (Greece)*.

LEARNING BY EXAMPLE

Reinforcement Learning Techniques for Real Autonomous Underwater Cable Tracking

Andres El-Fakdi, Marc Carreras

*Computer Vision and Robotics Group (VICOROB), Institute of Informatics and Applications
University of Girona, 17074 Girona, Spain
aelfakdi@eia.udg.edu, marcc@eia.udg.edu*

Javier Antich, Alberto Ortiz

*Department of Mathematics and Computer Science, University of Balearic Islands, 07071 Palma de Mallorca, Spain
javi.antich@uib.es, dmiaor0@uib.es*

Keywords: Machine learning in control applications, space and underwater robots.

Abstract: This paper proposes a field application of a high-level Reinforcement Learning (RL) control system for solving the action selection problem of an autonomous robot in cable tracking task. The learning system is characterized by using a Direct Policy Search method for learning the internal state/action mapping. Policy only algorithms may suffer from long convergence times when dealing with real robotics. In order to speed up the process, the learning phase has been carried out in a simulated environment and, in a second step, the policy has been transferred and tested successfully on a real robot. Future steps plan to continue the learning process on-line while on the real robot while performing the mentioned task. We demonstrate its feasibility with real experiments on the underwater robot *ICTINEU^{AUV}*.

1 INTRODUCTION

Reinforcement Learning (RL) is a widely used methodology in robot learning (Sutton and Barto, 1998). In RL, an agent tries to maximize a scalar evaluation obtained as a result of its interaction with the environment. The goal of a RL system is to find an optimal policy to map the state of the environment to an action which in turn will maximize the accumulated future rewards. The agent interacts with a new, undiscovered environment selecting actions computed as the best for each state, receiving a numerical reward for every decision. The rewards are used to teach the agent and in the end the robot learns which action it must take at each state, achieving an optimal or sub-optimal policy (state-action mapping).

The dominant approach over the last decade has been to apply reinforcement learning using the value function approach. Although value function methodologies have worked well in many applications, they have several limitations. The considerable amount of computational requirements that increase time consumption and the lack of generalization among continuous variables represent the two main disadvantages of "value" RL algorithms. Over the past few years, studies have shown that approximating a pol-

icy can be easier than working with value functions, and better results can be obtained (Sutton et al., 2000) (Anderson, 2000). Informally, it is intuitively simpler to determine *how to act* instead of *value of acting* (Aberdeen, 2003). So, rather than approximating a value function, new methodologies approximate a policy using an independent function approximator with its own parameters, trying to maximize the future expected reward. Only a few but promising practical applications of policy gradient algorithms have appeared, this paper emphasizes the work presented in (Bagnell and Schneider, 2001), where an autonomous helicopter learns to fly using an off-line model-based policy search method. Also important is the work presented in (Rosenstein and Barto, 2001) where a simple "biologically motivated" policy gradient method is used to teach a robot in a weightlifting task. More recent is the work done in (Kohl and Stone, 2004) where a simplified policy gradient algorithm is implemented to optimize the gait of Sony's AIBO quadrupedal robot.

All these recent applications share a common drawback, gradient estimators used in these algorithms may have a large variance (Marbach and Tsitsiklis, 2000)(Konda and Tsitsiklis, 2003) what means that policy gradient methods learn much more slower

than RL algorithms using a value function (Sutton et al., 2000) and they can converge to local optima if the expected reward (Meuleau et al., 2001), making them less suitable for on-line learning in real applications. In order to decrease convergence times and avoid local optima, newest applications combine policy gradient algorithms with other methodologies, it is worth to mention the work done in (Tadrake et al 2004) and (Matsubara et al., 2005), where a biped robot is trained to walk by means of a “hybrid” RL algorithm that combines policy search with value function methods.

One form of robot learning, commonly called *teaching* or *learning by example* techniques, offers a good proposal for speeding up gradient methods. In those ones, the agent learns to perform a task by analyzing or “watching” the task being performed by human or control code. The advantages of teaching are various. Teaching can direct the learner to explore the promising part of search space which contains the goal states. This is a very important aspect when dealing with large state-spaces whose exploration may be infeasible. Also, local maxima dead ends can be avoided with example learning techniques (Lin, 1992). Differing from *supervised learning*, bad examples or “bad lessons” will also help the agent to learn a good policy, so the teacher can also select bad actions during the teaching period. The idea of providing high-level information and then use machine learning to improve the policy has been successfully used in (Smart, 2002) where a mobile robot learns to perform a corridor following task with the supply of example trajectories. In (Atkenson et al., 1997) the agent learns a reward function from demonstration and a task model by attempting to perform the task. Finally, cite the work done in (Hammer et al., 2006) concerning an outdoor mobile robot that learns to avoid collisions by observing a human driver operate the vehicle.

This paper proposes a reinforcement learning application where the underwater vehicle *ICTINEU^{AUV}* carries out a visual based cable tracking task using a direct gradient algorithm to represent the policy. An initial example policy is first computed by means of computer simulation where a model of the vehicle simulates the cable following task. Once the simulated results are accurate enough, in a second phase, the policy is transferred to the vehicle and executed in a real test. A third step will be mentioned as a future work, where the learning procedure continues on-line while the robot performs the task, with the objective of improving the initial example policy as a result of the interaction with the real environment. This paper is structured as follows. In Section 2 the learning

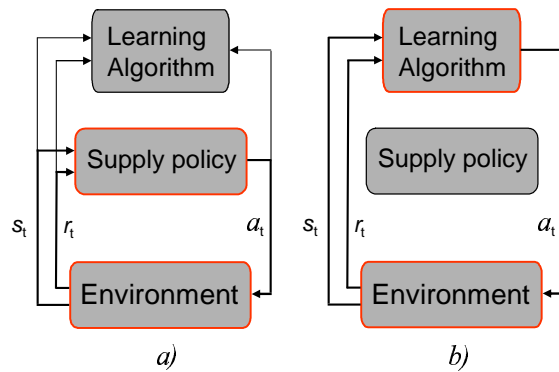


Figure 1: Learning phases.

procedure and the policy gradient algorithm are detailed. Section 3 describes all the elements that affect our problem: the underwater robot, the vision system, the simulated model and the controller. Details and results of the simulation process and the real test are given in Section 4 and finally, conclusions and the future work to be done are included in Section 5.

2 LEARNING PROCEDURE

The introduction of prior knowledge in a gradient descent methodology can dramatically decrease the convergence time of the algorithm. This advantage is even more important when dealing with real systems, where timing is a key factor. Such learning systems divide its procedure into two phases or steps as shown in Fig. 1. In the first phase of learning (see Fig. 1(a)) the robot is being controlled by a supply policy while performing the task; during this phase, the agent extracts all useful information. In a second step, once it is considered that the agent has enough knowledge to build a “secure” policy, it takes control of the robot and the learning process continues, see Fig. 1(b).

In human teaching, the pilot applies its policy to solve the problem while the agent is passively watching the states, actions and rewards that the human is generating. During this phase, the human will drive the learning algorithm through “hot spots” of the state-space, in other words, the human will expose the agent to those areas of the state-space where the rewards are high, not with the aim of learning a human control itself but to generate a positive dataset to feed the learning algorithm. All this information is used by the RL system to compute an initial policy. Once it is considered that the agent’s policy is good enough, the learning procedure will switch to the second phase, continuing to improve the policy as it would be in a standard RL implementation. But the supply policy

mentioned before can be represented by a human, by another robot or even a coded control policy. The proposal presented here takes advantage of learning by simulation as an initial startup for the learner. The objective is to transfer an initial policy, learned in a simulated environment, to a real robot and test the behavior of the learned policy in real conditions. First, the learning task will be performed in simulation with the aim of a model of the robot. Once the learning process is considered to be finished, the policy will be transferred to *ICTINEU^{AUV}* in order to test it in the real world. The Baxter and Bartlett approach (Baxter and Bartlett, 1999) is the gradient descent method selected to carry out the simulated learning corresponding to phase one. Next subsection gives details about the algorithm.

2.1 The Gradient Descent Algorithm

The Baxter and Bartlett's algorithm is a policy search methodology with the aim of obtaining a parameterized policy that converges to an optimal by computing approximations of the gradient of the averaged reward from a single path of a controlled POMDP. The convergence of the method is proven with probability 1, and one of the most attractive features is that it can be implemented on-line. In a previous work (El-Fakdi et al., 2006), the same algorithm was used in a simulation task achieving good results. The algorithm's procedure is summarized in Algorithm 1. The algorithm works as follows: having initialized the parameters vector θ_0 , the initial state i_0 and the eligibility trace $z_0 = 0$, the learning procedure will be iterated T times. At every iteration, the parameters' eligibility z_t will be updated according to the policy gradient approximation. The discount factor $\beta \in [0, 1)$ increases or decreases the agent's memory of past actions. The immediate reward received $r(i_{t+1})$, and the learning rate α allows us to finally compute the new vector of parameters θ_{t+1} . The current policy is directly modified by the new parameters becoming a new policy to be followed by the next iteration, getting closer to a final policy that represents a correct solution of the problem.

The algorithm is designed to work on-line. The function approximator adopted to define our policy is an artificial neural network (ANN) whose weights represent the policy parameters to be updated at every iteration step (see Fig. 2). As input, the network receives an observation of the state and, as output, a soft-max distribution evaluates each possible future state exponentiating the real-valued ANN outputs $\{o_1, \dots, o_n\}$, being n the number of neurons of the output layer (Aberdeen, 2003). After applying the soft-

max function, the outputs of the neural network give a weighting $\xi_j \in (0, 1)$ to each of the possible control actions. The probability of the i th control action is then given by:

$$Pr_i = \frac{\exp(o_i)}{\sum_{a=1}^n \exp(o_a)} \quad (1)$$

where n is the number of neurons at the output layer. Actions have been labeled with the associated control action and chosen at random from this probability distribution, driving the learner to a new state with its associated reward.

Once the action has been selected, the error at the output layer is used to compute the local gradients of the rest of the network. The whole expression is implemented similarly to error back propagation (Haykin, 1999). The old network parameters are updated following expression 3.(e) of Algorithm 1:

Algorithm 1: Baxter and Bartlett's OLPOMDP algorithm.

1. Initialize:
 - $T > 0$
 - Initial parameter values $\theta_0 \in R^K$
 - Initial state i_0
 2. Set $z_0 = 0$ ($z_0 \in R^K$)
 3. for $t = 0$ to T do:
 - (a) Observe state y_t
 - (b) Generate control action u_t according to current policy $\mu(\theta, y_t)$
 - (c) Observe the reward obtained $r(i_{t+1})$
 - (d) Set $z_{t+1} = \beta z_t + \frac{\nabla \mu_{u_t}(\theta, y_t)}{\mu_{u_t}(\theta, y_t)}$
 - (e) Set $\theta_{t+1} = \theta_t + \alpha r(i_{t+1}) z_{t+1}$
 4. end for
-

$$\theta_{t+1} = \theta_t + \alpha r(i_{t+1}) z_{t+1} \quad (2)$$

The vector of parameters θ_t represents the network

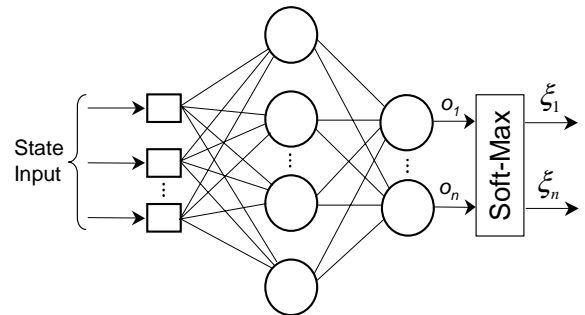


Figure 2: Schema of the ANN architecture adopted.

weights to be updated, $r(i_{t+1})$ is the reward given to the learner at every time step, z_{t+1} describes the estimated gradients mentioned before and, at last, we have α as the learning rate of the algorithm.

3 CASE TO STUDY: CABLE TRACKING

This section is going to describe the different elements that take place into our problem: first, a brief description of the underwater robot *ICTINEU^{AUV}* and its model used in simulation is given. The section will also present the problem of underwater cable tracking and, finally, a description of the neural-network controller designed for both, the simulation and the real phases is detailed.

3.1 *ICTINEU^{AUV}*

The underwater vehicle *ICTINEU^{AUV}* was originally designed to compete in the SAUC-E competition that took place in London during the summer of 2006 (Ribas et al., 2007). Since then, the robot has been used as a research platform for different underwater inspection projects which include dams, harbors, shallow waters and cable/pipeline inspection.

The main design principle of *ICTINEU^{AUV}* was to adopt a cheap structure simple to maintain and upgrade. For these reasons, the robot has been designed as an open frame vehicle. With a weight of 52 Kg, the robot has a complete sensor suite including an imaging sonar, a DVL, a compass, a pressure gauge, a temperature sensor, a DGPS unit and two cameras: a color one facing forward direction and a B/W camera with downward orientation. Hardware and batteries are enclosed into two cylindrical hulls designed to withstand pressures of 11 atmospheres. The weight is mainly located at the bottom of the vehicle, ensuring the stability in both *pitch* and *roll* degrees of freedom. Its five thrusters will allow *ICTINEU^{AUV}* to be operated in the remaining degrees of freedom (*surge*, *sway*, *heave* and *yaw*) achieving maximum speeds of 3 knots (see Fig. 3).

The mathematical model of *ICTINEU^{AUV}* used during the simulated learning phase has been obtained by means of parameter identification methods (Ridao et al., 2004). The whole model has been uncoupled and reduced to emulate a robot with only two degrees of freedom (DOF), X movement and rotation respect Z axis.

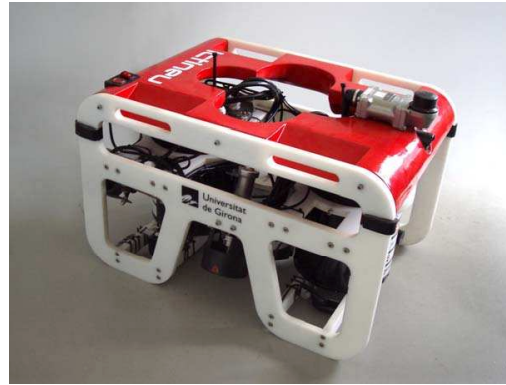


Figure 3: The autonomous underwater vehicle *ICTINEU^{AUV}*.

3.2 The Cable Tracking Vision System

The downward-looking B/W camera installed on *ICTINEU^{AUV}* will be used for the vision algorithm to track the cable. It provides a large underwater field of view (about 57° in width by 43° in height). This kind of sensor will not provide us with absolute localization information but will give us relative data about position and orientation of the cable respect to our vehicle: if we are too close/far or if we should move to the left/right in order to center the object in our image. The vision-based algorithm used to locate the cable was first proposed in (Ortiz et al., 2002) and later improved in (Antich and Ortiz, 2003). It exploits the fact that artificial objects present in natural environments usually have distinguishing features; in the case of the cable, given its rigidity and shape, strong alignments can be expected near its sides. The algorithm will evaluate the polar coordinates ρ and Θ of the straight line corresponding to the detected cable in the image plane (see Fig. 4).

Once the cable has been located and the polar coordinates of the corresponding line obtained, as the cable is not a thin line but a large rectangle, we will also compute the cartesian coordinates (x_g, y_g) (see Fig. 4) of the object's centroid with respect to the image plane by means of (3).

$$\rho = x \cos(\Theta) + y \sin(\Theta) \quad (3)$$

where x and y correspond to the position of any point of the line in the image plane. The computed parameters Θ , x_g and y_g together with its derivatives will conform the observed state input of the neural-network controller. For the simulated phase, a downward-looking camera model has been used to emulate the vision system of the vehicle.

3.3 The Neural-network Controller

A one-hidden-layer neural-network with 6 input nodes, 3 hidden nodes and 5 output nodes was used to generate a stochastic policy. As can be seen in Fig. 5 the inputs to the network correspond to the normalized state vector computed in the previous section $s = \{\theta, x_g, y_g, \frac{\delta\theta}{\delta t}, \frac{\delta x_g}{\delta t}, \frac{\delta y_g}{\delta t}\}$. Each hidden and output layer has the usual additional bias term. The activation function used for the neurons in the hidden layer is the hyperbolic tangent type while the output layer nodes are linear. The five output neurons represent the possible five control actions (see Fig. 6). The discrete action set $A = \{a_1, a_2, a_3, a_4, a_5\}$ has been considered where $A_1 = (Surge, Yaw)$, $A_2 = (Surge, -Yaw)$, $A_3 = (-Surge, Yaw)$, $A_4 = (-Surge, -Yaw)$, $A_5 = (Surge, 0)$. Each action corresponds to a combination of a constant scalar value of *Surge* force (movement) and *Yaw* force (rotation respect Z axis).

As explained in Section 2.1, the outputs have been exponentiated and normalized to produce a probability distribution. Control actions are selected at random from this distribution.

4 RESULTS

4.1 1st Phase: Simulated Learning

The model of the underwater robot *ICTINEU^{AUV}* navigates a two dimensional world at 1 meter height above the seafloor. The simulated cable is placed at the bottom in a fixed circular position. The controller

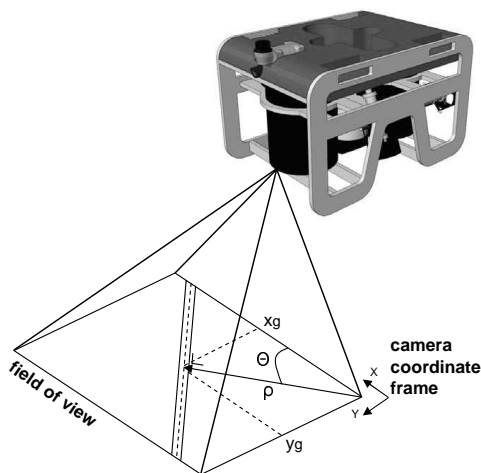


Figure 4: Coordinates of the target cable with respect to *ICTINEU^{AUV}*.

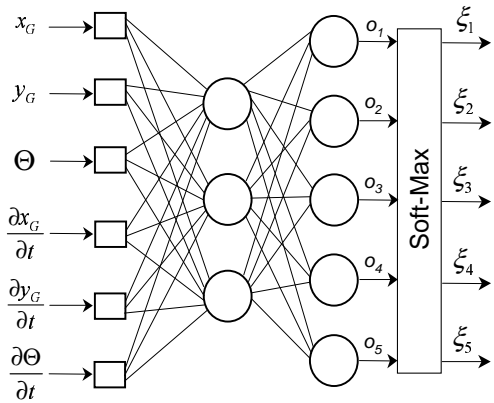


Figure 5: The ANN used by the controller.

has been trained in an episodic task. An episode ends either every 15 seconds (150 iterations) or when the robot misses the cable in the image plane, whatever comes first. When the episode ends, the robot position is reset to a random position and orientation around the cable's location, assuring any location of the cable within the image plane at the beginning of each episode. According to the values of the state parameters $\{\theta, x_g, y_g\}$, a scalar immediate reward is given each iteration step. Three values were used: -10, -1 and 0. In order to maintain the cable centered in the image plane, the positive reward $r = 0$ is given when the position of the centroid (x_g, y_g) is around the center of the image $(x_g \pm 0.15, y_g \pm 0.15)$ and the angle θ is close to 90° ($90^\circ \pm 15^\circ$), a $r = -1$ is given in any other location within the image plane. The reward value of -10 is given when the vehicle misses the target and the episode ends.

The number of episodes to be done has been set to 2.000. For every episode, the total amount of reward perceived is calculated. Figure 7 represents the performance of the neural-network robot controller as a function of the number of episodes when trained using Baxter and Bartlett's algorithm on the controller detailed in Section 3.3. The experiment has been repeated in 100 independent runs, and the results here presented are a mean over these runs. The learning

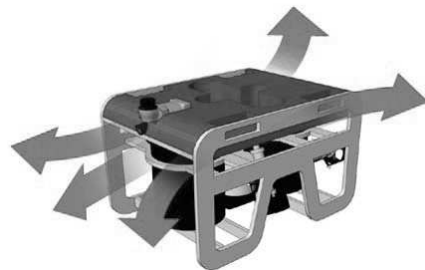


Figure 6: *ICTINEU^{AUV}* discrete action set.

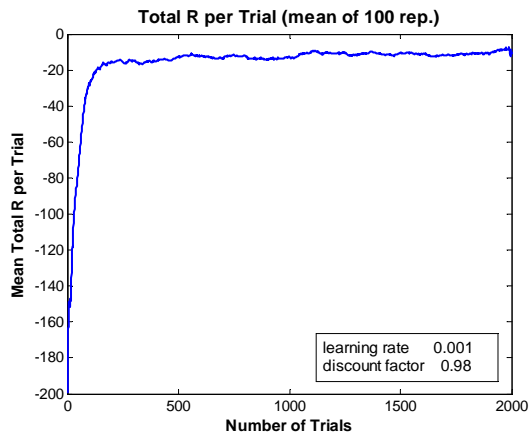


Figure 7: Performance of the neural-network robot controller as a function of the number of episodes. Performance estimates were generated by simulating 2.000 episodes. Process repeated in 100 independent runs. The results are a mean of these runs. Fixed $\alpha = 0.001$, and $\beta = 0.98$.

rate was set to $\alpha = 0.001$ and the discount factor $\beta = 0.98$. In Figure 8 we can observe a state/action mapping of a trained controller, y_g and the state derivatives $\frac{\delta\theta}{\delta t}$, $\frac{\delta x_g}{\delta t}$, $\frac{\delta y_g}{\delta t}$ have been fixed in order to represent a comprehensive graph. Figure 9 represents the trajectory of a trained robot controller.

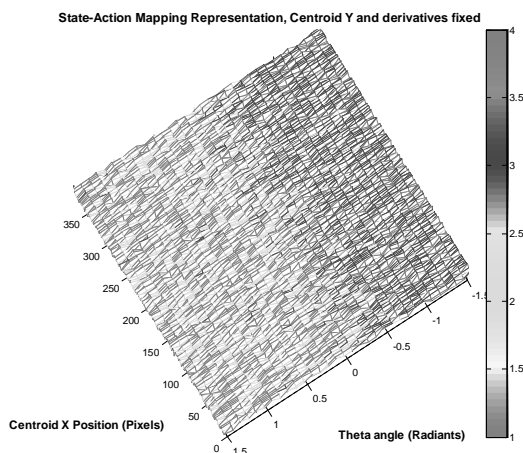


Figure 8: Centroid X - Theta mapping of a trained robot controller. The rest of the state variables have been fixed. Colorbar on the right represents the actions taken.

4.2 2nd Phase: Learned Policy Transfer. Real Test

Once the learning process is considered to be finished, the weights of the trained ANN representing the policy are transferred to *ICTINEU^{AUV}* and its performance tested in a real environment. The robot's con-

troller is the same one used in simulation. The experimental setup can be seen in Fig. 10 where the detected cable is shown while the vehicle performs a test inside the pool. Fig. 11 represents real trajectories of the θ angle of the image plane while the vehicle performs different trials to center the cable in the image.

5 CONCLUSIONS AND FUTURE WORK

This paper proposes a field application of a high-level Reinforcement Learning (RL) control system for solving the action selection problem of an autonomous robot in cable tracking task. The learning system is characterized by using a direct policy search algorithm for robot control based on Baxter and Bartlett's direct-gradient algorithm. The policy is represented by a neural network whose weights are the policy parameters. In order to speed up the process, the learning phase has been carried out in a simulated environment and then transferred and tested successfully on the real robot *ICTINEU^{AUV}*.

Results of this work show a good performance of the learned policy. Convergence times of the simulation process were not too long if we take into account the reduced dimensions of the ANN used in the simulation. Although it is not a hard task to learn in simulation, continue the learning autonomously in a real situation represents a challenge due to the nature of underwater environments. Future steps are focused on improving the initial policy by means of on-line learning processes and comparing the results obtained with human pilots tracking trajectories.

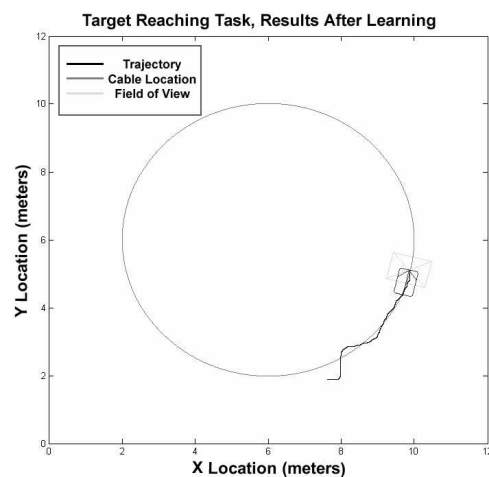


Figure 9: Behavior of a trained robot controller, results of the simulated cable tracking task after learning period is completed.

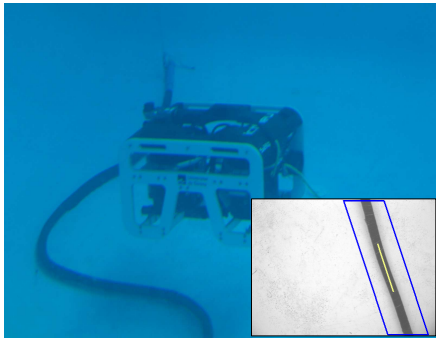


Figure 10: *ICTINEU*^{AUV} in the test pool. Small bottom-right image: Detected cable.

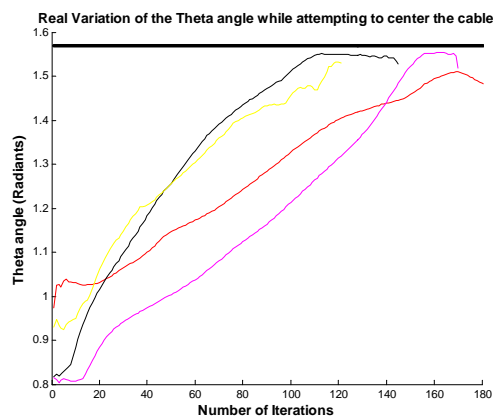


Figure 11: Real measured trajectories of the θ angle of the image plane while attempting to center the cable.

ACKNOWLEDGEMENTS

This work has been financed by the Spanish Government Commission MCYT, project number DPI2005-09001-C03-01, also partially funded by the MOMARNET EU project MRTN-CT-2004-505026 and the European Research Training Network on Key Technologies for Intervention Autonomous Underwater Vehicles FREESUBNET, contract number MRTN-CT-2006-036186.

REFERENCES

Aberdeen, D. A. (2003). *Policy-Gradient Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Australian National University.

Anderson, C. (2000). Approximating a policy can be easier than approximating a value function. Computer science technical report, University of Colorado State.

Antich, J. and Ortiz, A. (2003). Underwater cable tracking by visual feedback. In *First Iberian Conference*

on Pattern recognition and Image Analysis (IbPRIA, LNCS 2652), Port d'Andratx, Spain.

- Atkenson, C., Moore, A., and Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, 11:11–73.
- Bagnell, J. and Schneider, J. (2001). Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Korea.
- Baxter, J. and Bartlett, P. (1999). Direct gradient-based reinforcement learning: I. gradient estimation algorithms. Technical report, Australian National University.
- El-Fakdi, A., Carreras, M., and Ridao, P. (2006). Towards direct policy search reinforcement learning for robot control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Hammer, B., Singh, S., and Scherer, S. (2006). Learning obstacle avoidance parameters from operator behavior. *Journal of Field Robotics, Special Issue on Machine Learning Based Robotics in Unstructured Environments*, 23 (11/12).
- Haykin, S. (1999). *Neural Networks, a comprehensive foundation*. Prentice Hall, 2nd ed. edition.
- Kohl, N. and Stone, P. (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- Konda, V. and Tsitsiklis, J. (2003). On actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42, number 4:1143–1166.
- Lin, L. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3/4):293–321.
- Marbach, P. and Tsitsiklis, J. N. (2000). Gradient-based optimization of Markov reward processes: Practical variants. Technical report, Center for Communications Systems Research, University of Cambridge.
- Matsubara, T., Morimoto, J., Nakanishi, J., Sato, M., and Doya, K. (2005). Learning sensory feedback to CPG with policy gradient for biped locomotion. In *Proceedings of the International Conference on Robotics and Automation ICRA*, Barcelona, Spain.
- Meuleau, N., Peshkin, L., and Kim, K. (2001). Exploration in gradient based reinforcement learning. Technical report, Massachusetts Institute of Technology, AI Memo 2001-003.
- Ortiz, A., Simo, M., and Oliver, G. (2002). A vision system for an underwater cable tracker. *International Journal of Machine Vision and Applications*, 13 (3):129–140.
- Ribas, D., Palomeras, N., Ridao, P., Carreras, M., and Hernandez, E. (2007). Ictineu auv wins the first saucée competition. In *IEEE International Conference on Robotics and Automation*.
- Ridao, P., Tiano, A., El-Fakdi, A., Carreras, M., and Zirilli, A. (2004). On the identification of non-linear models of unmanned underwater vehicles. *Control Engineering Practice*, 12:1483–1499.

- Rosenstein, M. and Barto, A. (2001). Robot weightlifting by direct policy search. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
- Smart, W. (2002). *Making Reinforcement Learning Work on Real Robots*. PhD thesis, Department of Computer Science at Brown University, Rhode Island.
- Sutton, R. and Barto, A. (1998). *Reinforcement Learning, an introduction*. MIT Press.
- Sutton, R., McAllester, D., Singh, S., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12:1057–1063.
- Tedrake, R., Zhang, T. W., and Seung, H. S. (2004). Stochastic policy gradient reinforcement learning on a simple 3D biped. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'04*, Sendai, Japan.

VIDEO TRANSMISSION WITH ADAPTIVE QUALITY BASED ON NETWORK FEEDBACK FOR MOBILE ROBOT TELEOPERATION IN WIRELESS MULTI-HOP NETWORKS

Florian Zeiger, Markus Sauer and Klaus Schilling

University of Wuerzburg, Department of Robotics and Telematics, Am Hubland, Wuerzburg, Germany

{zeiger, sauer, schi}@informatik.uni-wuerzburg.de

Keywords: Networked robots, mobile robot teleoperation, mobile robot ad-hoc networks, wireless network, ad-hoc network, wireless teleoperation, wireless remote control.

Abstract: A video stream is still one of the most important data sources for the user while remote-operating a mobile robot. Human operators have comprehensive capabilities to interpret the displayed image information, but therefore, some constraints must be fulfilled. Constant frame rates and delays below a certain threshold are a minimum requirement to use video for teleoperation. Modern multi-hop networks often use WLAN to set up ad-hoc networks of mobile nodes with each node acting as traffic source, sink, or router. Considering these networks, routes between sources and destinations might be established via several relay nodes. Thus, the utilization of intermediate nodes which are part of a route influences the overall route performance, whereas sender and receiver have no direct feedback of the overall route status. In case video is transmitted via wireless ad-hoc networks in a teleoperation scenario, the displayed video-stream for the operator might have variable frame rates, very high packet loss, and packet inter-arrival times which are not appropriate for mobile robot teleoperation. This work presents an approach using a feedback generated by the network to adapt the image quality to present communication constraints. Thus, according to the current network status, the best possible video image is provided to the operator while keeping constant frame rates and low packet loss.

1 INTRODUCTION

In the meantime, mobile robots are planned to be used or even already used in many civil applications like surveillance or search and rescue to support and relieve the humans in place. Often, wireless communication is chosen to distribute and share information between the humans and robots in the team. This includes the transmissions of sensor data from the robots, observations from the humans, commands, and plans to the different team entities from the human coordinators. In many cases IEEE 802.11 wireless LAN is used as underlying technology for the wireless network interconnecting the team. Nowadays modern telecommunication equipment with small power consumption and interfaces for easy integration is available. This even allows an affordable system of wireless ad-hoc networks of mobile robots and human team members. These wireless ad-hoc networks offer a lot of advantages in contrast to static wireless network configurations, but also raise a lot of new challenges in the system design. In (Hu and Johnson, 2002) a live audio and

video data transmission via a multi-hop wireless network is demonstrated. In addition, several systems of rovers with autonomous functionalities (Parker, 1994), groups of unmanned aerial vehicles (Ollero et al., 2004), as well as heterogeneous multi robot systems were proposed. Rooker and Birk presented multi-robot exploration with robots using wireless networks (Rooker and Birk, 2007). For ground based systems Chung (Chung et al., 2002) presented a test bed for a network of mobile robots. In the field of rescue robotics (Rooker and Birk, 2005) or for integrating UAVs into IP based ground networks (Zeiger et al., 2007) the use of wireless networks is quite common nowadays. With respect to unmanned aerial vehicles (UAVs), (Ollero et al., 2003) presented a system using an access point running in WLAN infrastructure mode onboard the UAV. (Vidal et al., 2002) presented a system for communication between a ground station and a UAV using WLAN in combination with a high-gain antenna and radio modem. The University of Pennsylvania presented a mobile robot team connected via wireless network which performed localization and control tasks (Das et al.,

2002). Currently, wireless ad-hoc networks for mobile robots are a challenging and interesting scientific topic and scenarios connecting several mobile robots, humans in place (e.g. search and rescue applications), and stationary network nodes (e.g. communication relay nodes) are evaluated and analyzed (cf. Figure 1).

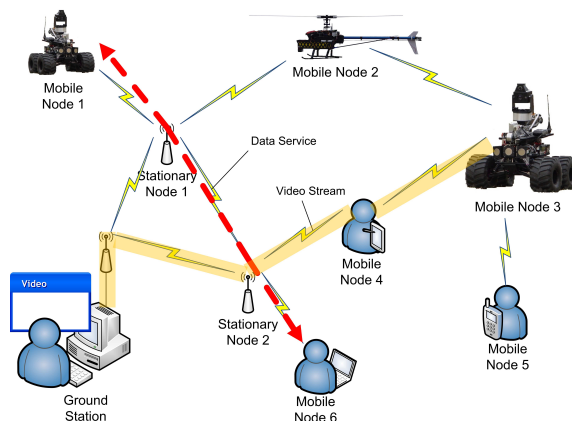


Figure 1: Future scenario of a heterogeneous network of mobile robots and human personnel.

Special ad-hoc routing protocols like AODV (Das et al., 2003)(Chakeres and Belding-Royer, 2004), DSR (Johnson and Maltz, 1996), or OLSR (Clausen, 2003) allow communication also in a highly dynamic network topology which increases the capabilities and the ease of use of mobile robots. These networks allow any-to-any communication between all nodes inside the network on a logical layer. Nevertheless, the radio link always implies the potential danger of a complete communication drop-out and the unpredictable loss of packets with a variable packet loss probability. Also the delay of packets delivered via the same route by hop-by-hop fashion can be variable. The same also applies for the bandwidth – e.g. IEEE 802.11 WLAN usually reduces its bandwidth as the link quality decreases.

The dynamic characteristics of WLAN and especially if it is used together with ad-hoc routing protocols has special drawbacks if direct teleoperation should be implemented over these communication links. Although there was a lot of progress in the area of autonomy for mobile robots, still many applications need the direct teleoperation of mobile robots, which requires in many cases reliable and high bandwidth links for video streams from the robots. For low-bandwidth conditions and very defined environments, e.g. in tele-education, also virtual representations can be used to provide the necessary information for direct teleoperation (Sauer et al., 2005). For the more dynamic scenarios, (e.g. in search and rescue) where the application of wireless ad-hoc networks is

very desirable, direct teleoperation with high bandwidths is mostly required. These needs for high bandwidth result from the fact, that the video feedback still delivers the most and richest information from the remote environment to the operator. This detailed information from the remote site is needed to increase and maintain the situation awareness and common ground between robot and human operator as basis for any future decisions and commands done by the human operator. (Murphy and Burke, 2005) showed that this situation awareness is even more important than any autonomy or assistance function implemented in the robot. Dependent on the human teleoperation task different characteristics of the video stream are important. If a navigation task is considered, the most important parameters are a high frame rate, low number of frame losses, and a constant inter-arrival time between to frames. Compared to these parameters the quality and resolution of the video stream is less important for navigation. On the other hand if the human has a search task (e.g. identify objects in a delivered video stream), the quality and resolution has a higher importance than the frame rate. Here, the proposed mechanism for the video-stream adaptation according to the load status of the route is designed for navigating a mobile robot with direct teleoperation.

The presented mechanism allows a variable image quality of the video stream for the operator. The quality is adjusted automatically to the current state of the wireless multi-hop network and respectively the available bandwidth of the used route by using a feedback of the network status. As above mentioned, the state of each single node of a route has a strong influence on the quality of the used link in terms of bandwidth, delay, and packet loss. To increase the performance of mobile robot teleoperation, the available frame rate at the operator PC should be almost constant. In order to adjust the image quality according to the link, an active feedback mechanism is implemented at the application layer of each node. Thus, a feedback of the network is available for the video stream source which can be used to adapt the image quality. The proposed mechanism requires only little resources, is portable and easy to implement, and provides the operator the highest possible video quality for mobile robot teleoperation which can be guaranteed for the current network state. As it supports no traffic classes as it is known from wired IP networks, it should not be considered as a quality of service (QoS) mechanism. Anyway, available quality of service (QoS) mechanisms – e.g. integrated services (IntServ) or differentiated services (DiffServ) – are currently not applicable in ad-hoc networks of mobile robots due to very specific hardware requirements and the special solu-

tions which are currently available for network service providers.

The remainder of this work is structured as follows. In Section 2, the investigated scenario is described and a short definition of the problem is given. Section 4 presents the implementation of the network feedback mechanism and the adaptation of the video stream in detail. The next Section gives an overview of the mobile robot, the communication, and video hardware which is used in the test scenarios. Section 5 defines the test scenarios of real hardware tests and gives the results. Finally, a conclusion is given in Section 6.

2 PROBLEM DEFINITION AND SCENARIO

2.1 Problem Definition

In the above mentioned scenarios, the available throughput of a route via a wireless multi-hop network is a highly dynamic parameter which depends on many environmental influences and affects the quality of the application significantly. The throughput of a wireless node can be decreased due to different reasons. In case intermediate nodes of a route are also part of a route which has to transport other bandwidth intensive flows, the available bandwidth must be shared between all present routes via this node, which will reduce the available bandwidth for the video link. Furthermore, also a decreasing link quality will reduce the bandwidth and increase the packet loss probability. If the network is not reacting to traffic overload at a specific node, this will lead to unpredictable packet loss at this point and delays at the different receivers. For the teleoperation scenario the effect will be that the video stream will get randomly stuck, because packets get lost. Most probably the operator will get confused and will stop the robot.

2.2 Test Scenario Additional Traffic

To set up the scenario where a node is used for more than one bandwidth intensive traffic flow, four nodes are used (cf. Figure 2). All nodes are located such that they are in direct communication range. During the tests, defined additional UDP traffic will be generated between node 3 and node 4 while the investigated video stream is transmitted via UDP from the mobile robot to the user's PC via node 3. The generated UDP traffic is used to reach certain load levels at intermediate node 3. As in this scenario, node 3 and node 4

are in communication range to all other nodes which will also cause interferences at the physical layer.

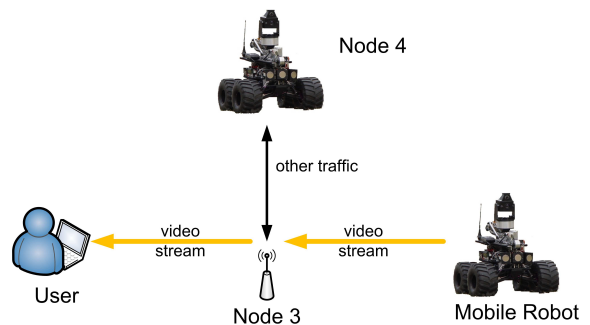


Figure 2: The test setup for additional traffic.

To provide best repeatability of the tests, all nodes are stationary. Only the additional traffic between node 3 and node 4 will be varied according to a defined profile. Measured categories are the packet loss and the packet inter-arrival times. These categories are measured while the amount of additionally generated traffic is increased. As reference test, video transmissions of constant target quality are used and compared to the packet loss of the transmission with adaptive quality.

3 HARDWARE

The proposed mechanism was tested in a real outdoor environment with a wireless ad-hoc network of four nodes. One is the PC of the operator, one is an Outdoor MERLIN (cf. Figure 3) (Eck et al., 2007), and two intermediate nodes are MERLIN robots (indoor version). More details on the scenario are shown in Figure 2 and a detailed description of the test setup is given in Section 2. Figure 4 shows the detailed system setup. All MERLIN robots have a C167 microcontroller for low-level operations and sensor data processing, as well as a PC-104 for more complex and computationally more intensive tasks. The PC-104 uses a Linux operating system and all nodes are equipped with 802.11b standard WLAN equipment (Atheros chip).

To grab the video from an analog camera (approx. 65 degree field of view) an Axis video server is used. It can grab the video from up to four cameras with a resolution of 768x576 pixels. Dependent on the configuration and connected clients, a frame rate of up to 25 images per second can be provided either as MJPEG or MPEG4 over a TCP/IP connection. For the described tests the PC-104 is connected over a cross-link cable to the Ethernet interface of the

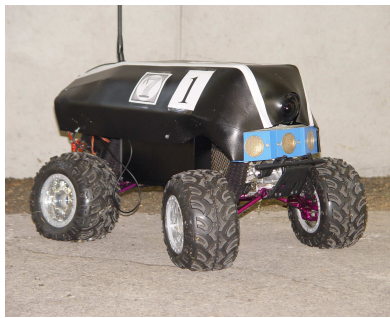


Figure 3: The Teleoperated OutdoorMERLIN Robot.

video server. As nothing else is connected to this Ethernet interface of the PC-104 it can be exclusively used for the video traffic. For the presented tests four MJPEG video streams with full resolution are established with four different compression rates. MJPEG as video compression was selected, as MPEG4 compression takes a significant longer time on the Axis server what causes a significant delay in the video stream. Secondly a loss of a packet during transmission of MPEG4 streams to the robot might lead to longer set of distorted images because compared to MJPEG not all frames of the stream contain the full image information needed. In case of the investigated scenario, the MJPEG frames are transmitted via UDP protocol.

4 NETWORK FEEDBACK AND ADAPTIVE VIDEO QUALITY

The proposed mechanism mainly consists of two parts: the network feedback, and the adaptive adjustment of the video quality. The mechanism is used for a simple admission control of the video source and intends to provide the best possible video image quality considering the current state of the link. The objective is an efficient use of the available bandwidth without overloading the route with video traffic to the operator. Thus, it is not used to increase the link quality directly but uses the available resources most efficient and reliable for the operators' video stream.

4.1 Network Feedback

The network feedback is responsible to transmit the status of a node to the video source. Therefore, nodes of the network host a small client program at the application layer. This client application is listening in promiscuous mode at layer 3 of the ISO/OSI model (IP-layer) and measures the utilization of the wireless link. All kinds of traffic are monitored: incoming and

outgoing packets, packets for forwarding, and packets with other nodes in range as destination – basically all traffic causing the radio link of this node to be busy. The network feedback client sends very small UDP packets with an adjustable frequency (in the test setup 10 Hz) and 8 bytes as payload to the video-source if it is a used hop in the video stream route between video-source and receiving node. This payload is used to indicate the status of the corresponding node, either “normal operation” or “overload situation”. In the beginning, each node is in the “normal operation” mode. As soon as a certain utilization of the supported bandwidth is exceeded, the status of this node switches to “overload situation”. Important parameters for the network feedback clients are the feedback frequency f and the threshold for status determination d . In case f is too high, too much feedback traffic is generated which degrades the performance of the network. Even these packets are very small, too many small packets with a high sending frequency will have a very bad effect on 802.11b WLAN and will significantly decrease the throughput. Thus, the generated feedback traffic should be limited depending on the interpretation rate of the video adjustment mechanism and the selected load window for the wireless nodes. Often it is also not necessary to run a feedback client on each network node. For setting parameter d , it should be considered, that d specifies the percentage of the nominal bandwidth (e.g. for 802.11b this would be 11 Mbit/sec) which can be used without switching to the “overload situation” state. The feedback clients measures packets on layer 3, where the maximum available bandwidth corresponds to the “goodput” of the wireless link which is about 75% of the nominal link bandwidth (e.g. for 802.11b this would be 75% of 11 Mbit/sec).

As the proposed mechanism is used within a network where a link failure can occur at any time, the measurement and signaling mechanism must be active. Thus, link failures and link reestablishing can be monitored reliably. As the mechanism for video quality adaptation performs best with a feedback frequency of $f = 10$ Hz (according to the presented scenario), the generated measurement traffic has a bandwidth of less than 0.003 Mbit/sec per measurement node. To set parameter d , the “goodput” of about 7 to 7.5 Mbit/sec (for an 11 Mbit/sec WLAN link) must be considered. In order to allow a reaction on potential overload situations while providing the user a video stream with a bandwidth of 1 to 1.5 Mbit/sec for the best quality, d is set to 50.

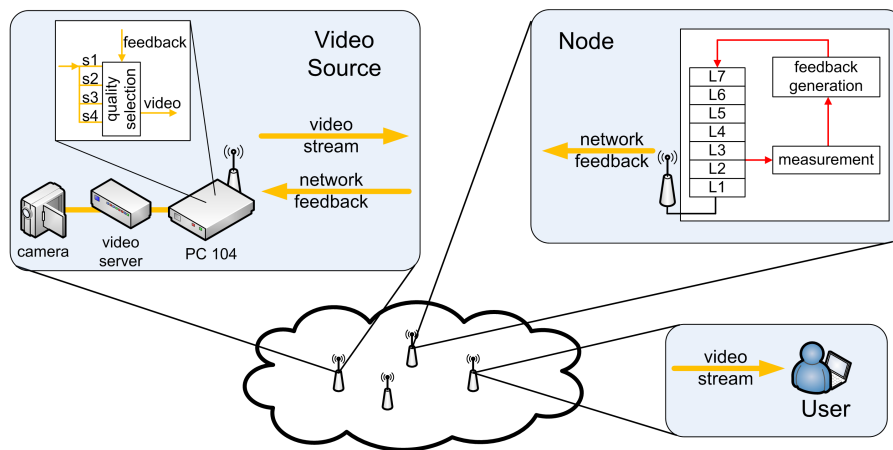


Figure 4: System setup.

4.2 Adaptive Video Quality

The Video Quality in the presented system is adapted according to current state of the ad-hoc route for the video transfer. The adaption mechanism receives all status packages from the nodes between two received frames from the image source, interprets these packages and selects the quality for the next frames with a combination of previous status data and the current state. Do reduce oscillating behavior in quality switching near the selected load limit of the nodes a kind of inertia mechanism for the adaptation process was integrated. The implemented inertia mechanism guarantees not to change the image quality whenever a status of a node changes. It is possible to set a certain number (cf. Algorithm 1, min/max of inertia_counter) of receiving same successive route load states until the quality is changed. Algorithm 1 shows this mechanism how the quality for the next frame is selected according to the received network status messages.

In the current test setup, four different video qualities are used at a frame rate of 11 frames per second each. Table 1 shows the average size of one image for the corresponding image quality level.

A higher number of different quality scales would also be possible. In the current test setup a minimum of -3 and a maximum of 3 are selected for the inertia_counter. With this value the mechanism reacts in the worst case after six frames with subsequent overload states and in average after three frames. This keeps the load caused by the video traffic on the different nodes in a certain defined window around the selected threshold for overload state. In combination with parameter d of the above described feedback mechanism, the quality adjustment intervenes as soon as a node exceeds a radio link utilization of more than

Algorithm 1: Video quality adaptation.

Input: video streams of different quality;
load status messages

initialization;

foreach frame of current selected quality **do**

if one of the nodes overloaded **then**

 increment inertia_counter by one;

else

 decrement inertia_counter by one;

end

 reset node states;

 send video frame;

if inertia_counter above max **then**

 select lower quality if possible;

 set inertia_counter to zero;

else

if inertia_counter below min **then**

 select higher quality if possible;

 set inertia_counter to zero;

end

end

end

Table 1: Average size of one image per quality level.

Quality	minimum	low	medium	high
Size (kbytes)	15	26	34	47

approx. 78% ($\approx 50\%$ of nominal bandwidth). This prevents the node from reaching a utilization of 100% of the available maximum throughput which would result in a high packet loss rate due to an increasing number of packet collisions.

5 TEST AND RESULTS

In a first step, a reference scenario was set up and measured. Therefore, no network feedback mechanism is used and a mobile robot generates a video stream which is sent to the PC of the operator as it is displayed in Figure 2. Between node 4 and node 3, additional traffic is generated during the different test phases according to Table 2 to reach a defined load at intermediate node 3.

Table 2: Generated additional traffic.

Phase	generated additional traffic (Mbit/sec)
1	0
2	3,2
3	4
4	4,8
5	5,6
6	6,4
7	7,2
8	8
9	8,8

The results of this reference test are shown in Figure 5. The x-axis shows the test time in milliseconds. The left y-axis describes the received frame rate in frames per second (fps) and the right y-axis displays the received video data rate in bytes per second (bps) at the receiving node (operator’s PC).

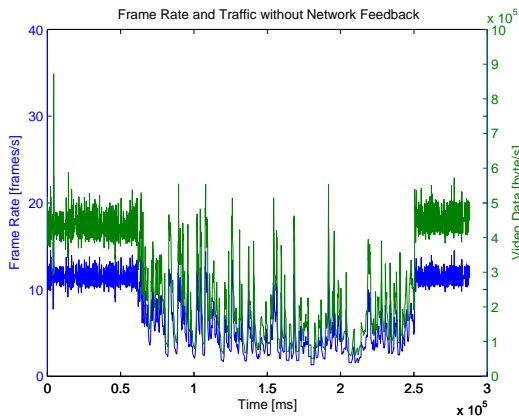


Figure 5: Framerate and Traffic without Network Feedback.

The test started with no additional traffic being generated. Successively, more and more additional traffic is generated by switching to the next phase each 20 seconds according to Table 2. After 200 seconds of test time, the additionally generated traffic is reduced by switching back one phase each 10 seconds. In the beginning of the test – during phase 1 up to the

end of phase 3 – the received frame rate is about 11 fps. After switching to phase 4 at about 60 seconds, the received video frame rate decreases significantly. The received frame rate between 100 and 200 seconds drops to 2 – 3 fps while node 3 is overloaded. After the additionally generated traffic is reduced, the received frame rate recovered to 11 fps. Increasing the additional traffic forces node 3 to an overload situation. As the bandwidth used by the video stream cannot be adapted to the new situation, a packet loss of the video data is inevitable which is shown in Figure 6. The y-axis shows the number of lost packets vs. the test time on the x-axis.

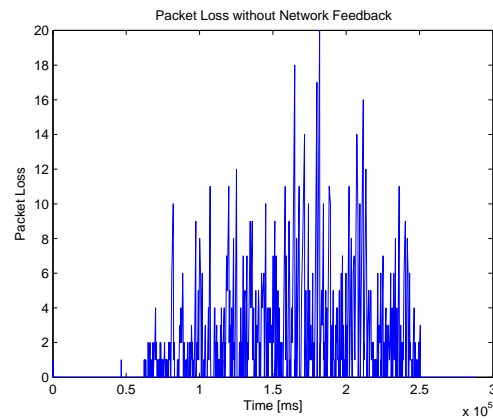


Figure 6: Packet Loss without Network Feedback.

Another measured category is the frame inter-arrival time of the video stream. This is a quite sensitive aspect, as a large jitter (variance of the frame inter-arrival time) is very irritating for the operator due to a very unsteady motion of the video image. Without additional traffic, the frame inter-arrival time is smaller than 100 ms with a variance close to 0 (cf. Figure 7) what corresponds to the average frame rate of 11 fps. After 60 seconds and an additionally generated traffic of 4.8 Mbit/sec, the frame inter arrival time increases to more than 400 ms with a variance of more than 10000 which indicates an unacceptable video for the operator.

The same test setup is used again – now with the network feedback and adaptive quality mechanism (cf. Section 4), which should improve the observed behavior. In Figure 8, the frame rate and the video data rate is shown while using an adaptive video quality together with the network feedback mechanism. In the beginning, without additional traffic, the mobile robot generates a video stream of about 450000 bytes/sec. During the test, the additionally generated traffic is increased similar to the test described above. The implemented mechanism takes care that the video

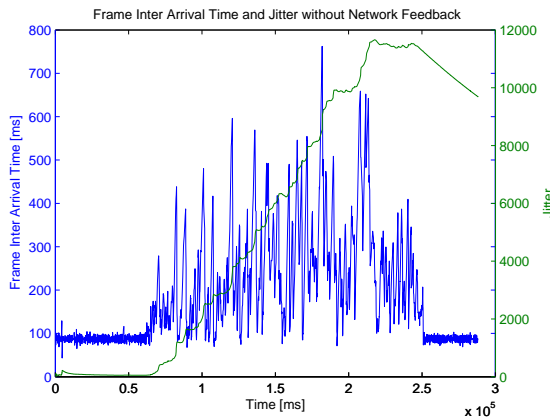


Figure 7: Frame Inter Arrival Time without Network Feedback.

source reduces its generated video traffic to about 300000 as soon as phase 3 (with an additional load of 4 Mbit/sec) is entered. Increasing the additional load at node 3 to more than 4.8 Mbit/sec results again in a reduction of the video traffic (180000 bytes/sec). During the complete test run, the frame rate stays almost constantly at 11 fps as the adaptive video bandwidth reduction avoids the loss of video traffic. Also the frame inter arrival time stays constantly below 100 ms with a jitter of almost 0 (cf. Figure 9).

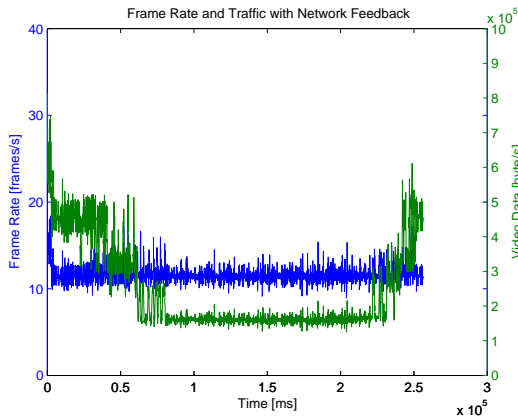


Figure 8: Framerate and Traffic with Network Feedback.

6 CONCLUSIONS

In this work, a mechanism for providing a video stream over a dynamic multi-hop route with an adaptive quality for mobile robot teleoperation is proposed. The mechanism uses a feedback from the network which is generated at dedicated nodes and adjusts the image quality to the current communication

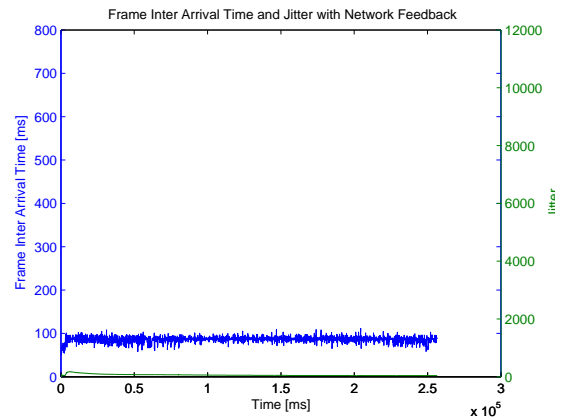


Figure 9: Frame Inter Arrival Time with Network Feedback.

link status. The proper functionality of this adaptive quality mechanism is tested in teleoperation scenarios with real hardware under different network load situations. In situations with a very high link load due to additional other network traffic, usually the packet loss rate and the packet inter-arrival time is affected in a way that reliable and proper teleoperation is not possible anymore. By adjusting the image quality of the video stream it is possible to provide a stable video frame rate for the operator. In fact, the remaining bandwidth for the video stream is used efficiently in terms of providing a video with a stable frame rate suitable for mobile robot teleoperation.

REFERENCES

- Chakeres, I. D. and Belding-Royer, E. M. (2004). AODV Routing Protocol Implementation Design. In *Proceedings of the International Workshop on Wireless Ad hoc Networking (WWAN), Tokyo, Japan*, pages 698–703.
- Chung, T., Cremean, L., Dunbar, W. B., Jin, Z., Klavins, E., Moore, D., Tiwari, A., van Gogh, D., and Stephen-Waydo (2002). A Platform for Cooperative and Coordinated Control of Multiple Vehicles. *3rd Conference on Cooperative Control and Optimization*.
- Clausen, P. J. T. (2003). RFC 3626, Optimized link state routing protocol (OLSR). IETF, Network Working Group.
- Das, A., Spletzer, J., Kumar, V., and Taylor, C. (2002). Ad Hoc Networks for Localization and Control. In *Proceedings of the 41st IEEE Conference on Decision and Control, (CDC 2002)*, volume 3, pages 2978–2983.
- Das, S., Perkins, C. E., and Belding-Royer, E. M. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. IETF RFC 3561.
- Eck, D., Stahl, M., and Schilling, K. (2007). The Small Outdoor Rover MERLIN and its Assistance System for Tele-Operations. In *Proceedings of International*

- Conference on Field and Service Robotics (FSR), Chamonix (France).*
- Hu, Y. and Johnson, D. (2002). Design and Demonstration of Live Audio and Video over Multihop Wireless Ad Hoc Networks. In *Proceedings of the MILCOM 2002*.
- Johnson, D. B. and Maltz, D. A. (1996). Dynamic Source Routing in Ad-Hoc Wireless Networks. *Mobile Computing*, 353.
- Murphy, R. and Burke, J. L. (2005). Up from the Rubble: Lessons Learned about HRI from Search and Rescue. In *Proceedings of the 49th Annual Meetings of the Human Factors and Ergonomics Society*.
- Ollero, A., Alcazar, J., Cuesta, F., Lopez-Pichaco, F., and Nogales, C. (2003). Helicopter Teleoperation for Aerial Monitoring in the COMETS Multi-UAV System. In *3rd IARP Workshop on Service, Assistive and Personal Robots (IARP 2003), Madrid (Spain)*.
- Ollero, A., Hommel, G., Gancet, J., Gutierrez, L.-G., Viegas, D., Forssn, P.-E., and Gonzlez, M. (2004). COMETS: A multiple heterogeneous UAV system. In *IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR 2004)*.
- Parker, L. E. (1994). ALLIANCE: An Architecture for Fault Tolerant, Cooperative Control of Heterogeneous Mobile Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1994)*.
- Rooker, M. N. and Birk, A. (2005). Combining Exploration and Ad-Hoc Networking in RoboCup Rescue. In Nardi, D., Riedmiller, M., and Sammut, C., editors, *RoboCup 2004: Robot Soccer World Cup VIII*, volume 3276 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 236–246. Springer.
- Rooker, M. N. and Birk, A. (2007). Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445.
- Sauer, M., Zeiger, F., Driewer, F., and Schilling, K. (2005). Remote control on mobile robots in low bandwidth environments. In *International Conference on Informatics in Control, Automation and Robotics, Barcelona, Spain*, pages 163 – 168.
- Vidal, R., Shakernia, O., Kim, H. J., Shima, H., and Sastry, S. (2002). Multi-Agent Probabilistic Pursuit-Evasion Games with Unmanned Ground and Aerial Vehicles. *IEEE Transactions on Robotics and Automation*, Vol. 18, Number 5:662–669.
- Zeiger, F., Selbach, C., Ruderisch, B., and Schilling, K. (2007). An Application Protocol to Integrate a Small Size Helicopter into an IP based Ad-Hoc Network. In *Proceedings of ROBOCOMM 2007, Athens (Greece)*.

ALTITUDE CONTROL OF SMALL HELICOPTERS USING A PROTOTYPE TEST BED

Nikos I. Vitzilaios and Nikos C. Tsourveloudis

Intelligent Systems & Robotics Laboratory, Technical University of Crete, Chania, Greece
vitzilaios@dpem.tuc.gr, nikost@dpem.tuc.gr

Keywords: Unmanned Helicopters, Experimental Test Bed, Altitude Control, Fuzzy Control, Aerial Robotics.

Abstract: In this paper we present an experimental test bed for the development and evaluation of control systems for unmanned helicopters. The test bed consists of a small unmanned helicopter, mounted on a flying stand that permits all possible movements but prevents the helicopter from damaging or crashing. A fuzzy controller is developed in MATLAB and tested in the helicopter using the test bed. The controller is able to perform hovering and altitude control. Experimental results are presented for various test cases.

1 INTRODUCTION

Unmanned helicopters are the most flexible flying machines among the variety of UAVs (Unmanned Aerial Vehicles), since they have the ability to take off and land vertically as well as to perform aggressive maneuvers and hovering, which gives them the advantage of effective observation from various positions. These advantages along with the continuous development of robotic vehicles' technology have led to the use of unmanned helicopters in many applications, both civil and military, such as surveillance, traffic management, land management, border patrol, and search and rescue missions. As a result, there has been remarkable growth in the market of unmanned helicopters (aka VTOL UAVs for Vertical Take-Off and Landing UAVs), which nowadays includes vehicles of various types, sizes and operational capabilities (Spanoudakis et al., 2003). During the last years, small scale (about 1500 mm in length) helicopters are preferred for development and experimentation due to their low cost and expendability.

Although small scale unmanned helicopters offer as experimentation platforms the advantages of low cost and easy operation, the development of autonomous navigation systems for such vehicles is a difficult and dangerous procedure that may increase this overall cost, since except from the equipment needed (helicopter, sensors, telemetry systems etc) one should add the cost of crashes and damages that may occur during experimentation.

Since helicopters are very unstable and difficult to control, experimentation on real vehicles often result in damaging accidents. For this reason, the development of an autonomous navigation controller begins with numerous tests in a software-based simulation environment. In this environment, controllers are evaluated for their ability to control efficiently the helicopter. If the simulation results are encouraging, the controller may be tested on the real vehicle.

The simulation procedure has drawbacks as well. At first, the simulation environment cannot imitate helicopter's navigation in detail with all possible environmental disturbances. Therefore, a controller that seems to work satisfactorily in the simulation may be insufficient for the navigation of the real vehicle in a real environment. Moreover, independently of any simulation evaluation, first/initial tests with a real vehicle generally are the most dangerous, since a lot of unexpected problems may arise at this time. As a result, it would be desirable to test the controller on a real vehicle but in a safe environment, without having the danger of crashing and destroying the equipment or harm people that monitor the flight.

In the past years, there have been proposed ways of testing controllers on a real vehicle safely. Normally there is a mechanical construction where a real helicopter (or a simplified model of it) can fly indoors without crashing or harming the humans involved in the experimentation.

In the literature we meet constructions that simulate a real helicopter. In (Tanaka, Ohtake, and

Wang, 2004), a custom helicopter-like construction whose degrees of freedom are reduced, is used for the design and evaluation of a flight stabilization controller. In (Andrievsky, Peaucelle, and Fradkov, 2007), a mechanical construction is used to emulate the flying behavior of a helicopter. The experimental setup consists of a base on which a long arm is mounted that carries the helicopter body. Two motors with propellers mounted on the helicopter body can generate the force that causes the helicopter body to lift off the ground. A similar test bed is also used in (Kutay et al., 2005).

Further in the literature, we meet systems that use real helicopters for the experiments. In (Dzul, Lozano, and Castillo, 2004) and (Mancini et al., 2007), a mechanical construction holds the helicopter in a stable position allowing only small and safe movements. Using mechanical limitations, the helicopter is able to move in only one or two axes and within limits. As a result the helicopter cannot take any dangerous orientation or collide to the ground.

The drawback of the work presented in the above references, is that either a helicopter emulation construction is used, or a real helicopter with reduced degrees of freedom. In both cases, the developed controller partially covers the control of the vehicle in one or two axes and it is not sufficient to fully control a helicopter in real conditions. The motivation of this paper is the construction of a laboratory test bed where small helicopters can be safely (for both humans and the equipment involved) used indoors for experimental validation without limitations in helicopter's movement. Indoor flying gives the ability for continuous tests regardless of weather conditions. Moreover, the suggested setup minimizes the need for experienced helicopter pilots within the research group. Flying small helicopters requires pilot training which stems research efforts towards autonomous helicopter flights.

In this paper we propose a fuzzy controller for the altitude and hovering control of an unmanned helicopter. The controller is developed using the proposed test bed and is able to stabilize the helicopter in desired positions (each position is defined by horizontal and vertical coordinates). Except from hovering at a desired altitude, the tasks of autonomous take-off and landing are also considered here.

In the literature there is previous work on the autonomous altitude control of unmanned helicopters. Usually altitude control is a part of an autonomous navigation controller (Shin et al., 2005), (Kim and Shim, 2003), where a subsystem dedicated

to altitude control cooperates with other subsystems in order to navigate the helicopter. In (Kim et al., 2004) an adaptive approach is proposed for altitude control for an unmanned helicopter which utilizes rotor RPM to track altitude commands. Significant work has been done also in the field of autonomous landing problem for unmanned helicopters (Sapiralli, Sukhatme, and Montgomery, 2002), (Merz, Duranti, and Conte, 2006).

This paper is organised as follows. In Section 2 we present the experimental test bed that we use in order to develop the controller. Main parts and systems of the test bed are presented as well as the way this test bed works. In Section 3 we present a fuzzy controller able to control the altitude of the helicopter and perform hovering at a stable desired position. In Section 4, experimental results are presented and remarked. At last, a conclusion is derived as well as future work on the subject is suggested.

2 EXPERIMENTAL TEST BED

The laboratory test bed consists of three basic elements; a customized flying stand, a customized helicopter and a ground control station (Figure 1).

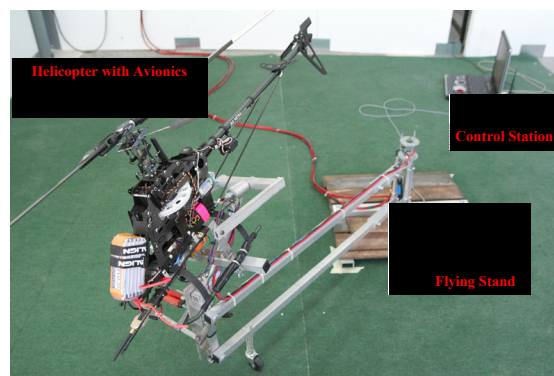


Figure 1: View of the experimental test bed.

2.1 Helicopter Flying Stand

Helicopter flying stand is a mechanical construction able to hold the helicopter, allowing full movements (6 degrees of freedom) while protecting it from damaging and crashing. It is a customized construction based on a commercially available flying stand that it is used by inexperienced pilots for flight training.

The stand allows the helicopter move naturally without any constraint around a 2.1m diameter circle

(Figure 6), flying forwards, backwards or sideways. A gas strut is used to counterbalance the weight of the stand. As a result the helicopter does not lift any extra weight. In Figure 2, rotations as well as the Euler angles of the helicopter are presented.

Since the test bed will be used for indoor experiments, a positioning system must be developed in order to know helicopter's position during testing. To avoid high cost indoor positioning and localization systems, we utilize the rotary movement of the central shaft of the stand. The stand and consequently the helicopter move around a circle (planar rotation at Figure 2, Figure 6) with a rotation angle which may easily be monitored. For this reason, we put a rotation encoder on the central shaft of the stand (Figure 3). The encoder initializes its position to zero and then gives signed numbers that denote the current position relative to the initial position. Positive numbers denote rotation to the left while negative numbers denote rotation to the right side. The rotation encoder gives the planar position of the helicopter at each time instant.

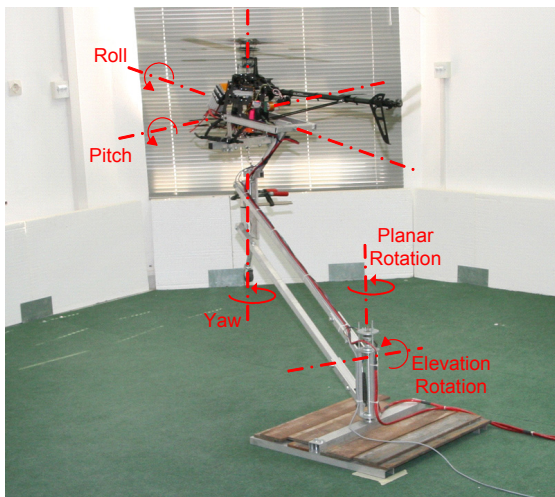


Figure 2: Euler angles and rotation axes.

Moreover, we need to know the altitude in which the helicopter flies. The flying stand gives the ability to the helicopter to fly at a maximum height of 60cm. An infrared sensor is used to monitor the actual value of altitude. The sensor is mounted at the lower part of the bracket that holds the helicopter, as it is shown in Figure 3. The accuracy of the altitude readings is less than 1cm, which is far better than the accuracy of outdoor altimeters or GPS.

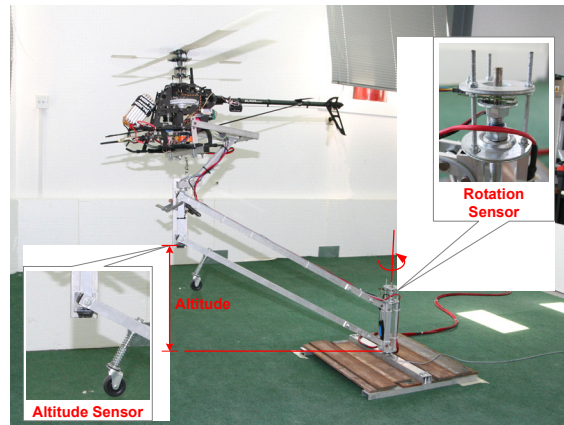


Figure 3: Positioning sensors.

2.2 Helicopter and Avionics

The VTOL that we use in our test bed is a customization of a 50-size (1200 mm length, 405 mm height, 1350 mm main rotor diameter) commercially available electric powered RC helicopter. An important characteristic of this helicopter is that it has electric motor so there is no need for fuel gas, and therefore it does not produce any exhaust gasses during its operation, which is important for indoor testing. This helicopter has been heavily customized in order to be ready for experimental use. In what follows we describe the additional equipment and avionics we have put on board.

2.2.1 Inertial Measurement Unit (IMU)

This unit gives the orientation of the helicopter. The unit consists of 3D gyroscopes, accelerometers and magnetometers and outputs the 3 Euler angles (roll, pitch and yaw). The IMU used is the commercial MTi model of Xsens Motion Technologies. For the communication between IMU and control station a USB-serial data and power cable is used.

2.2.2 Digital Switch

This is the interface that manages the switching from manual to autonomous flight. Manual flight is controlled remotely by a human operator, while autonomous flight is supervised by a Central Processing Unit (CPU). Switching between manual and autonomous flight is an important operation because it allows the human tester to regain manual control at any time instant during experimentation, which is very useful in case of failure or insufficient controller behaviour.

2.2.3 Servo Driver/Controller

RC servos are the actuators used to control the motion of the helicopter. In manual operation, the onboard receiver forwards the transmitter commands to servos by sending appropriate PWM signals. In order to send such signals from the control station to the servos, a servo driver is needed. For that reason a PIC microcontroller is used, which translates control signals from the ground station to RC PWM servo signals and drives the servos. Further, the PIC reads the input from the localization system (x-y position, altitude) and transmits it to the control station.

2.2.4 Communication System

A wireless communication system has been established between the control station and the PIC microcontroller. Having 2 receiver/transmitter units (one on the helicopter and one on the ground station) and by using the Bluetooth protocol, we obtain two-way communication between the serial port of the PIC and the serial port of the control station.

2.2.5 Power System

The electric helicopter has high power consumption. During hovering, the electric motor needs about 50A current of 25V. Normally in these helicopters, LiPo batteries are used that have high capacity and the ability to sustain big currents. With this consumption and with a high capacity LiPo battery, the helicopter can perform hovering for about 15 minutes. To overcome this limitation in the duration of experiments, the test bed is provided with constant power supply of 24V that gives continuous current to the helicopter.

2.3 Ground Control Station

Since our test bed works indoor and we can have all the signals through wireless communication (except from the IMU), there is no need to put any processor unit onboard. For this reason we use portable CPU which serves as the “control station”. Because of this solution, the helicopter has fewer payloads to lift, while the control station has increased processing power able to run control algorithms at high speeds.

In Figure 4 a block diagram presents the connections of the equipment and the data transmission through these connections, for each subsystem (flying stand, helicopter and control station).

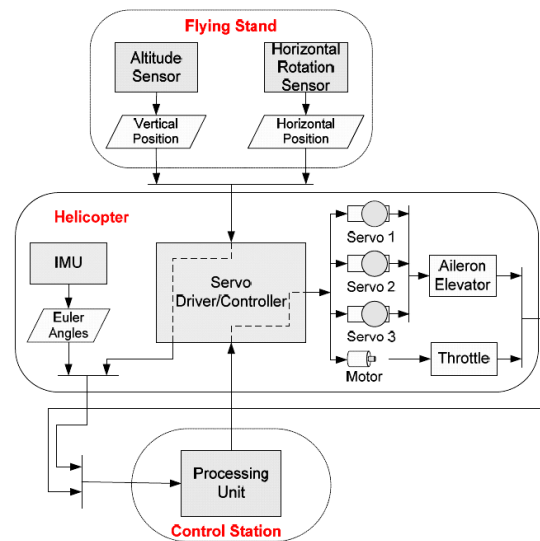


Figure 4: System Architecture.

3 ALTITUDE & HOVERING CONTROL

The controller developed and tested in the test bed is a fuzzy controller for altitude and hovering control. The objective of the controller is to hold stable the helicopter at a predefined horizontal position and altitude.

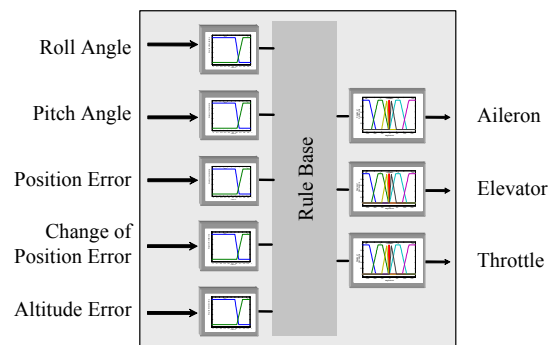


Figure 5: Hovering and Altitude Fuzzy controller.

3.1 Fuzzy Controller

A fuzzy controller of the Mamdani type has been designed and implemented (Figure 5) in the MATLAB environment. The objective of this controller is to keep the helicopter “hovering” at predefined positions subject to wind and other disturbances.

As shown in Figure 5, the inputs of the fuzzy controller are the *roll* and *pitch* angles of the

helicopter at every time instant, as well as the *position error*, the *change of position error* and the *altitude error*. In Figure 6 we show the representation of the *position error* input, which is defined as the difference between the current horizontal position and the target horizontal position.

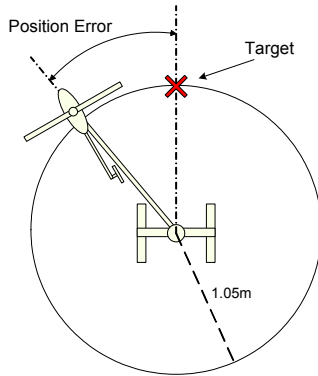


Figure 6: Position error representation.

As *position error* represents how far the helicopter is from the target point, the *change of position error* represents the way that position error changes and if the helicopter reaches the target point or moves away from it. The *altitude error* is also calculated as the difference between the current and the target altitude. The outputs of the controller are the change of the roll and pitch angles (*aileron* and *elevator* variables respectively), as well as the change in the *throttle* of the helicopter.

Roll angle is given by the IMU in real time. Although the flying stand permits roll angles from -30° to 30° , the flight control system takes as input degrees from -90° to 90° . The linguistic variables that represent the *roll angle* are: *left big* (LB), *left* (L), *zero* (ZERO), *right* (R), *right big* (RB), and their membership functions are shown in Figure 7.

The second input variable is the *pitch angle* of the helicopter. The linguistic variables for this input are: *back big* (BB), *back* (B), *zero* (ZERO), *front* (F), *front big* (FB), with membership functions also presented in Figure 7.

The third input variable is the *position error*, which is defined as the difference between the current and the desirable position. Since for safety reasons we do not want the stand to rotate out of its limits (-180° to 180° which corresponds to -30 to 30 in odometer units) we set the range of the *position error* variable to be between -30 to 30 (in odometer units). The linguistic variables for these inputs are: *negative big* (NB), *negative* (N), *zero* (ZERO), *positive* (P), *positive big* (PB) (Figure 7).

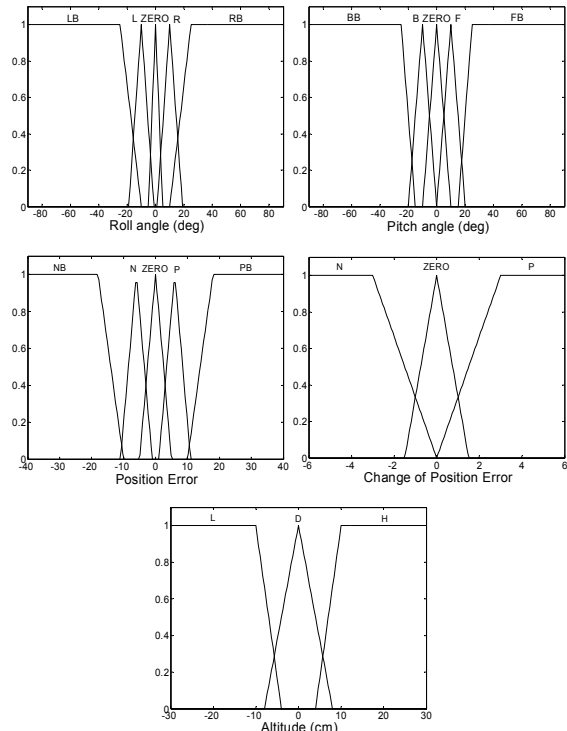


Figure 7: Membership functions for input variables.

The next input in the fuzzy controller is the *change of position error*. While position error shows how far the helicopter is from the desired position, *change of position error* shows how fast the vehicle is moving towards or away from the desired point. This input is defined as the difference (in odometer units) between the previous position error and the current position error, and it is represented by the linguistic variables: *negative* (N), *zero* (ZERO), *positive* (P) (Figure 7).

The last input is the *altitude error* input. This input represents the difference in cm between actual and desired altitude by counting if the helicopter is placed lower or higher than the desired position. The linguistic variables for this input are: *lower* (L), *desired* (D), *higher* (H) (Figure 7).

The outputs of the fuzzy controller are the changes of roll and pitch angles (Aileron and Elevator movements respectively) and Throttle change. The membership functions of *aileron*, *elevator* and *throttle*, are presented in Figure 8. The linguistic variables for *aileron* are *left big* (LB), *left* (L), *left small* (LS), *zero* (ZERO), *right small* (RS), *right* (R) and *right big* (RB). The linguistic variables for *elevator* are *back big* (BB), *back* (B), *zero* (ZERO), *front* (F) and *front big* (FB). Both *aileron* and *elevator* output values are presented in control signal units.

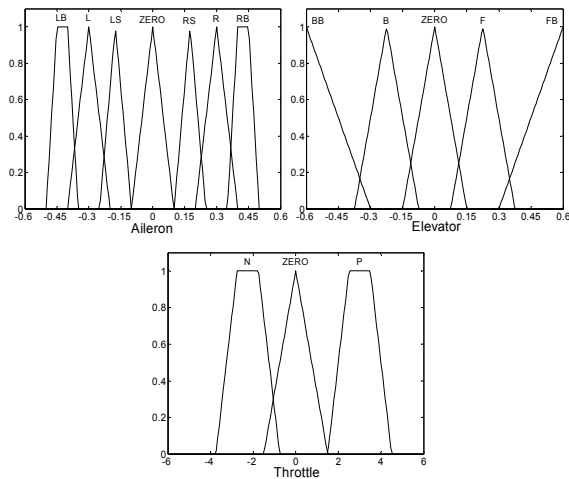


Figure 8: Membership functions for output variables.

The linguistic variables for *throttle* change are *negative (N)*, *zero (ZERO)* and *positive (P)*. The values of throttle output are also presented in control signal units. Negative output reduces throttle of the helicopter while positive output increases it.

It should be noted that in order to simplify the experimentation the yaw angle of the helicopter was set to zero. This is due to the fact that the yaw angle (tail movements) is usually stabilized in these helicopters by a gyro mechanism.

3.2 Control Rules

The control objective in the experiments performed was the stabilization of helicopter at a certain point (defined by horizontal and vertical target coordinates). The transition between the states of the controller is presented in Figure 9, while in Figure 10 the pseudo-code that describes the control scheme is shown. After take-off, the controller has as a target to hover the helicopter. Then checks actual horizontal position and drives the helicopter to the desired one. The next step is checking of actual altitude in order to drive the helicopter to the desired one. After some iterations where the helicopter hovers in the target point, the controller lands it.

For the implementation of this scheme, three sets (rule bases) of fuzzy IF-THEN rules were used. The one was responsible for the control of the pitch angle. The target was to keep the pitch angle always close to zero as this is what needs to be done when the helicopter hovers. This was achieved with simple rules of the form: <IF *Pitch* is *X* THEN *Elevator* is *Y*>, where *X*, *Y* represent the membership function of *pitch* and *elevator*, respectively.

The second rule base contains rules of the form: <IF *Roll* is *A* AND *position error* is *B* AND *change of position error* is *C* THEN *aileron* is *D*>. These rules lead the helicopter towards the desired point as they tend to minimize the distance between the helicopter's horizontal position at each moment and the desired one. This is a typical PD-like fuzzy controller with one extra input: the roll angle.

The third rule base is responsible for handling the throttle of the helicopter. The policy we follow here is that the changes in the throttle of the helicopter occur only when the helicopter is in stable hovering attitude on the desired horizontal position (roll and pitch angles are close to zero, change of position error is close to zero) or when the altitude becomes higher than a top safety limit. The rules of this rule base have the form <IF *Roll* is *A* AND *position error* is *B* AND *change of position error* is *C* AND *Altitude* is *D* THEN *throttle* is *E*>.

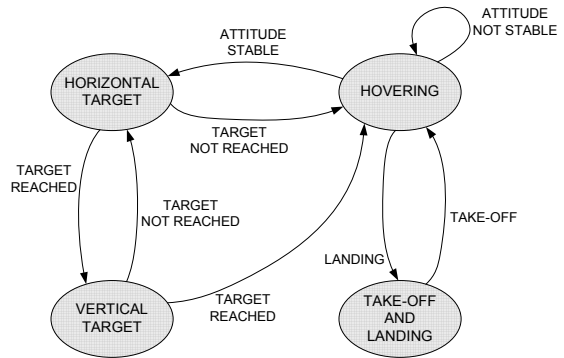


Figure 9: Controller state transition.

```

If attitude is not stable
    Stabilize helicopter to hovering
Else
    If current horizontal position is not the
    desired
        Drive helicopter to the desired
        horizontal position
    Else
        If Current Altitude is not the desired
            Change throttle in order to reach
            target altitude
        Else
            Hovering
        End IF
    End If
End If
    
```

Figure 10: Pseudo code of the hovering controller.

4 EXPERIMENTAL RESULTS

Experimental results for two test cases may be seen in Figures 11 and 12. In these figures *Roll* and *Pitch* values are measured in degrees, while *Position Error* and *Change of Position Error* are measured in odometer units (here, 1 odometer unit corresponds to 6 degrees) and *Altitude* is measured in centimetres. *Elevator*, *Aileron* and *Throttle* values are measured in control signals (values that PIC accepts as input and automatically translates into servo signals). The initial altitude of the helicopter (when the flying stand is on the ground) is 10cm, since in this altitude the infrared sensor is mounted to the stand.

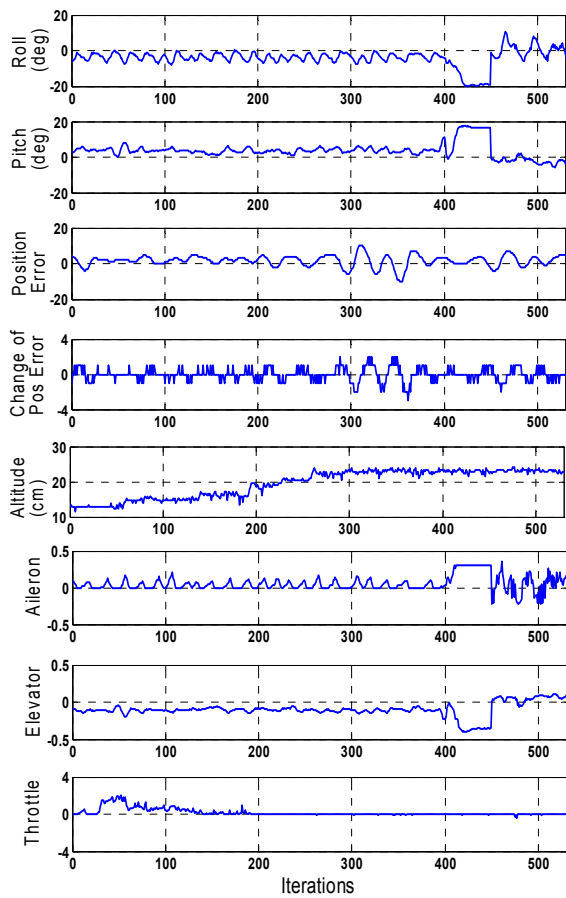


Figure 11: Experimental results for test case 1.

In **test case 1** (Figure 11) the ability of the controller to perform autonomous take-off and keep the helicopter in a hovering state, is evaluated. The helicopter is placed on the desired horizontal position by the human operator and then the autopilot takes over with a target altitude of about 22 cm. As it can be seen in Figure 11, the controller

keeps roll and pitch angles close to zero and gradually increases throttle in order to increase the altitude and reach the target one. When the target altitude is reached few oscillations around the target horizontal position occur but the controller manages to hold the helicopter in hovering in the desired position. In the beginning, it is clear that position error tends to be a small positive number, which means that the helicopter always drifts to the left of the desired position. This is explained by the position of the test bed area which is close to the walls of the building. Air flow from the main rotor of the helicopter circles through the walls and return as a disturbance to the helicopter. This air flow gives a small drift to helicopter to the left. The developed controller seems to recognize this disturbance and make corrections in order to hold stable the helicopter in the desired position.

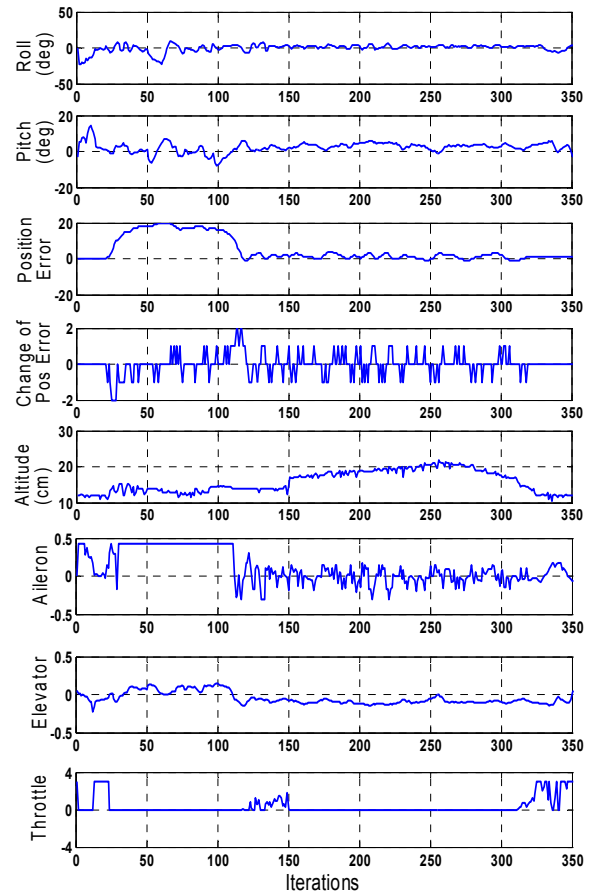


Figure 12: Experimental results for test case 2.

In Figure 12 we present the results of **test case 2**. In this test, the initial position of the helicopter is different from the desired one and the controller objective is to drive the helicopter to the desired

position and then land it autonomously. The helicopter is placed manually to a random position and then the fuzzy autopilot gains control of the helicopter. As one may see in the *Position Error* plot of Figure 12, the helicopter moves manually from its initial position to a random position. At time instant 50, the autopilot gains control of the vehicle. The target of the autopilot is to move the helicopter to the initial position and in 20 cm altitude. It is clear that the autopilot drives the helicopter to the target point by moving it to the desired horizontal position at first and then by raising the altitude until the targeted one has been reached. After a few iterations that the target position has been reached, the controller reduces the throttle and lands the helicopter. Small oscillations occur while the autopilot tries to keep the helicopter in stable position. It is also clear, as in test case 1, that we face the air disturbance that causes small drift in the helicopter in this test case too.

5 CONCLUSIONS

In this paper we presented a fuzzy controller for hovering and altitude control of a small-scale helicopter. The controller was developed and tested on a custom made laboratory experimental test bed, where tests on unmanned helicopters can be performed with safety. The test bed works indoors, is independent of power supply and can be used for continuous tests. The development of the controller is done on a real helicopter and not in simulation, so we can have direct and reliable results. The experimental results show that this setup works well. Experimental results from the evaluation of the altitude fuzzy controller were presented.

Future work, involves development of other kinds of controllers which will be tested and evaluated on the test bed. This work will lead to a comparison of controllers based on their efficiency and ability to control successfully an unmanned helicopter.

ACKNOWLEDGEMENTS

This paper is part of the 03ED365 research project, implemented within the framework of the "Reinforcement Programme of Human Research Manpower" (PENED) and co-financed by National and Community Funds (75% from E.U.-European Social Fund and 25% from the Greek Ministry of Development-General Secretariat of Research and Technology).

REFERENCES

- Andrievsky, B., Peaucelle, D., and Fradkov, A., 2007. Adaptive control of 3DOF motion for LAAS helicopter benchmark: Design and experiments. In *Proceedings of the 2007 American Control Conference*, New York City, USA.
- Dzul, A., Lozano, R., and Castillo, P., 2004. Adaptive control for a radio-controlled helicopter in a vertical flying stand. *International Journal of Adaptive Control and Signal Processing*, 18:473-485.
- Kim, H., and Shim, D., 2003. A flight control system for aerial robots: algorithms and experiments. *Control Engineering Practice*, 11:1389-1400.
- Kim, N., Calise, A., Corban, J. E., and Prasad, J. V. R., 2004. Adaptive output feedback for altitude control for an unmanned helicopter using rotor RPM. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Rhode Island, USA.
- Kutay, A., Calise, A., Idan, M., and Hovakimyan, N., 2005. Experimental results on adaptive output feedback control using a laboratory model helicopter. *IEEE Transactions on Control Systems Technology*, 13:196-202.
- Mancini, A., Caponetti, F., Monteriu, A., Frontoni, E., Zingaretti, P., and Longhi, S., 2007. Safe flying for an UAV helicopter. In *Proceedings of the 15th Mediterranean Conference on Control & Automation*, Athens, Greece.
- Merz, T., Duranti, S., and Conte, G., 2006. Autonomous landing of an unmanned helicopter based on vision and inertial sensing. *Experimental Robotics IX*, pp. 343-352.
- Sapiralli, S., Sukhatme, G., and Montgomery J., 2002. An experimental study of the autonomous helicopter landing problem. In *Proceedings of the International Symposium on Experimental Robotics*, Italy.
- Shin, J., Nonami, K., Fujiwara, D., and Hazawa, K., 2005. Model-based optimal attitude and positioning control of small-scale unmanned helicopter. *Robotica*, 23:51-63.
- Spanoudakis, P., Doitsidis, L., Tsourveloudis, N., and Valavanis, K., 2003. The market for VTOL UAVs. *Unmanned Systems Magazine*, Sept/Oct, pp. 14-18.
- Tanaka, K., Ohtake, H., and Wang, H., 2004. A practical design approach to stabilization of a 3-DOF RC helicopter. *IEEE Transactions on Control Systems Technology*, 12:315-325.

CLOCK SYNCHRONIZATION IN INDUSTRIAL AUTOMATION NETWORKS

Comparison of Different Syntonization Methods

Dragan Obradovic, Ruxandra Lupas Scheiterer, Chongning Na

Siemens AG, Corporate Technology, Information and Communications, Munich, Germany
dragan.obradovic@siemens.com, ruxandra.scheiterer@siemens.com, na.chongning.ext@siemens.com

Günter Steindl*, Franz-Josef Goetz**

Siemens AG, Automation and Drives, Industrial Automation Systems, Amberg, Germany
Siemens AG, Automation and Drives, Advanced Technologies and Standards, Nuremberg, Germany
guenter.steindl@siemens.com, franz-josef.goetz@siemens.com

Keywords: IEEE 1588, PTP, synchronization, syntonization, industrial automation network.

Abstract: Synchronization of distributed clocks is a critical task in many real time applications over Ethernet. The Ethernet protocol, due to its non-deterministic nature, is not suitable for real-time applications with very strict synchronicity requirements. However, the limit is continuously being pushed outwards by current research. The Precision Time Protocol (PTP), delivered by the IEEE 1588 standard, provides high synchronization accuracy and has been adopted in many real time applications in the areas of industrial automation, measurement & control, communications etc. This paper will discuss several issues aimed at improving the synchronization performance.

1 INTRODUCTION

Ethernet (IEEE 1997), due to its cheap cabling and infrastructure costs, high bandwidth, efficient switching technology and better interoperability, has been adopted in various areas to provide the basic networking solution. Many Ethernet-based applications require the networked clocks to be precisely synchronized. Typical examples include base station synchronization for handover or interference cancellation in telecommunication networks (Nieminen 2007), distribution of audio/video streams over Ethernet based networks (IEEE 2007a), and motion control in industrial Ethernet (Chen 2005). Standard Network Time Protocol (NTP) (Mills 1989, 1994) synchronization over Ethernet provides synchronization accuracy at the millisecond level, which is appropriate for processes that are not time critical. However, in many applications, for example base station synchronization or motion control, where only sub-microsecond level synchronization errors are allowed, a more accurate synchronization solution is needed. The Precision Time Protocol (PTP) of the

IEEE 1588 standard (IEEE 2002) published in 2002, is a promising Ethernet synchronization protocol, in which messages carrying precise timing information, obtained by hardware time stamping in the physical layer, are propagated in the network to synchronize the slave clocks to a master clock.

Factors that affect the synchronization quality achievable by PTP include the stability of oscillators, the resolution of message time stamping, the frequency of synchronization message transmission, and the propagation delay variation caused by the jitter in the intermediate elements. The synchronization error can be reduced by carefully studying the sources that contribute to the error, by choosing the most suitable implementation of PTP for a specific application and by designing efficient synchronization algorithms that make use of all available information provided by the PTP protocol.

Some work has been done to enhance the performance of IEEE 1588 taking the mentioned factors into consideration. The authors of (Jasperneite 2004) introduced the transparent clock (TC) concept to replace the so-called boundary clock (BC). BCs adjust their own clock to the master clock and then serve as master clocks for the next network

segment. Cascaded control loops are generated, which might lead to instabilities and deviation of the distributed clocks. Using TCs, intermediate bridges are treated as network components with known delay, which is compensated in the carried timing information. By doing this the synchronization at the time client is not dependent on the control loop design in the intermediate bridges. Hence performance is improved. The TC concept has been adopted in the new draft of IEEE 1588 published in 2007, and is used in this paper. In (Na 2007) we analyzed the influence of jitters and frequency drift and made suggestions for designing the parameters for higher synchronization accuracy. This study was extended in (Na 2008), where an algorithm to reduce the error was introduced.

In this paper, we discuss how to efficiently use the PTP messages to improve the synchronization performance, i.e. convergence speed and error. Problems arise when there is non-negligible frequency drift. In this case, clocks should first be syntonized, i.e. their frequency difference should be estimated and appropriate control applied to remove it. We study and compare different syntonization methods, and propose an improved solution for the syntonization and synchronization. Appropriate simulation results verify our analytic study.

The paper is organized as follows: Section 2 introduces the system model and briefly describes the PTP protocol. Section 3 introduces two methods for syntonization, master and peer frequency ratio estimation. In Section 4 we compare these methods and propose a synchronization algorithm which is based on both methods in section 5. Simulation results are presented in Section 6.

2 SYSTEM MODEL

Fig. 1 illustrates the time synchronization in a system with cascaded bridges. $N + 1$ elements are connected in a line topology. The first element is the time server, also called (grand)master, which provides the reference time to the other N elements, called slave elements, via time-aware bridges (TCs). The master element periodically sends Sync messages which carry the counter state of the master clock stamped at the time of transmission. The interval between two consecutive Sync messages is T . The i^{th} Sync message, generated by the master element at time t_i , consecutively passes through all slave elements. Quantities, certain or uncertain, linked with the Sync message transmitted by the

master at time t_i are labelled by the superscript i . We call the propagation time between the n^{th} slave and its preceding element line delay and denote by LD_n^i (also known as peer-to-peer delay in PTP).

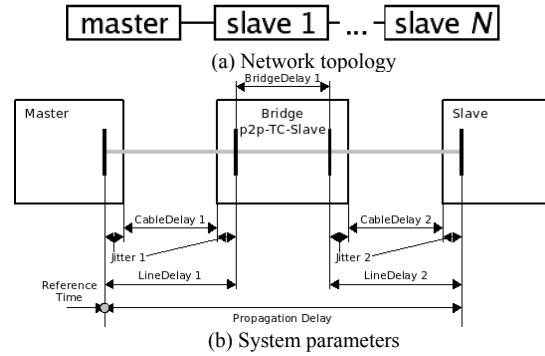


Figure 1: System Model.

The message will be forwarded to slave element $n + 1$ via a time-aware bridge after the bridge delay BD_n^i . We define LB_n^i to be the sum of line delay plus bridge delay of Sync message i at slave n . As the line delays and bridge delays are not necessarily constant in time, we define $\delta_{LB}^{i,n} = LB_n^i - LB_n^{i-1}$ to be the difference between the true LB value at slave n that affected Sync messages i and $i - 1$. All the delays we have mentioned up to now are defined in the absolute time. A delay D measured by a local clock takes the form $D \cdot f$ where f is the clock frequency (this product is replaced by an integral in the case of frequency drift).

The transparent clock synchronization protocol is depicted in Figure 2.

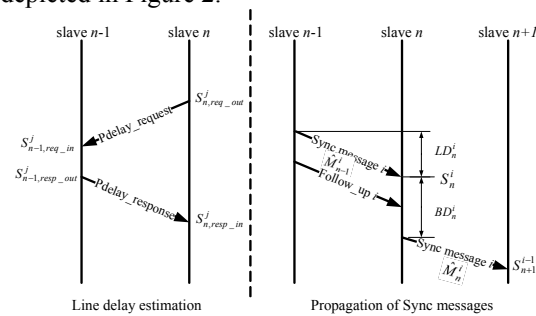


Figure 2: Illustration of PTP with transparent clocks.

The PTP has a master/slave structure. Timing information is packaged in special telegrams and propagated along the network. The synchronization relies on two processes, the delay estimation process and the timing propagation process. The delay

estimation process relies on 4 time-stamps, S_{n,req_out}^j , S_{n-1,req_in}^j , $S_{n-1,resp_out}^j$ and $S_{n,resp_in}^j$: slave n sends a delay request message to slave $n-1$ (which is the master in the case of slave 1) and records its time of departure (1st). Slave $n-1$ (or master) replies with a delay response message which reports the time-stamps of receiving the delay request message and sending the delay response message (2nd and 3rd).

Slave n records the time it receives the response message (4th). If slave n and $n-1$ have the same clock frequency or the frequency drift is negligible, the line delay can be calculated by:

$$\hat{S}_n^j(LD) = \frac{(S_{n,resp_in}^j - S_{n,req_out}^j) - (S_{n-1,resp_out}^j - S_{n-1,req_in}^j)}{2} \quad (1)$$

where $\hat{S}_n^j(LD)$ is the j^{th} estimated line delay using slave n 's local clock (and equal uplink and downlink line delays are assumed). The true line delay, measured in slave n 's local clock ticks, is $S_n^j(LD) = LD_n^j \cdot f_{S_n}$ for constant frequency.

In the timing propagation process each slave propagates the timing information of the master and uses that information to adjust its own clock. The master sends out a Sync message which contains the timestamp M when this message was sent. A more precise timestamp of the transmission of the Sync message will be sent by a so-called "follow-up message". Slave 1 forwards the Sync message to Slave 2, augmenting its content by the sum of its line and bridge delays (converted to master time – it will be explained presently how this is done), effectively transmitting its estimate of the master time for the time-instant of forwarding. This process is repeated in each slave until the message reaches the time client.

Consider for the moment that all the clocks have the same frequency. Then the updating of the content in slave n (i.e. his estimate of the master time) follows:

$$\hat{M}_n^i = \hat{M}_{n-1}^i + \hat{S}_n^i(LD) + \hat{S}_n^i(BD) \quad (2)$$

where $\hat{S}_n^i(LD)$ comes from the line delay estimation in (1). The bridge delay $\hat{S}_n^i(BD)$ is taken to be precisely known by using the time stamped at the reception and the forwarding of the Sync message.

Equation (2) can be used for proper time synchronization only if all clocks have the same frequency for all the time. If there is frequency difference between the clocks, the last two terms in (2), corresponding to the slave's counter increase during the two delays, are not equal to the counter increase during this time of the master clock.

Therefore, it is not suitable to use local time to update the master clock estimate, as shown in (2). To solve this problem, it is necessary to estimate the frequency offsets, i.e. syntonize the clocks.

3 SYNTONIZATION AND SYNCHRONIZATION IN PTP

As discussed in the previous section, if there are clock frequency drifts or the clocks have different frequencies, (1) and (2) are unsuitable for time synchronization. The problem with the line delay estimation in (1) is that $S_{n-1,resp_out}^j$ and

S_{n-1,req_in}^j are measured by the clock in slave $n-1$, whereas $S_{n,resp_in}^j$ and S_{n,req_out}^j are measured by the clock in slave n . To convert all into the same metric, the frequency difference between slave $n-1$ and n , i.e. neighboring slaves, needs to be known. And in (2), the last two terms should be translated into master time, i.e. the frequency difference of grandmaster and the slave needs to be known.

We define the rate compensation factor (**RCF**, also called rate ratio, (IEEE 2007b)) to be the ratio between the frequencies of two different clocks. We use $RCF_{X/Y}$ to denote the frequency ratio between X and Y , i.e. $RCF_{X/Y} = f_X / f_Y$. Then the correction of (1) is:

$$\hat{S}_n^j(LD) = \frac{(S_{n,resp_in}^j - S_{n,req_out}^j)}{2} - \frac{(S_{n-1,resp_out}^j - S_{n-1,req_in}^j) \cdot RCF_{S_n/S_{n-1}}}{2} \quad (3)$$

The master counter estimation equation of (2) should be changed to:

$$\begin{aligned} \hat{M}_n^i &= \hat{M}_{n-1}^i + (\hat{S}_n^i(LD) + \hat{S}_n^i(BD)) \cdot RCF_{M/S_n} = \\ &= \hat{M}_{n-1}^i + (\hat{S}_n^i(LD) + \hat{S}_n^i(BD)) \cdot \frac{1}{RCF_{S_n/M}} \end{aligned} \quad (4)$$

To compute RCF, observe that a time interval measured by two different clocks will result in different clock counter values. RCF can be calculated as the ratio of the clock counter values. The same time interval Δt is measured by clock 1 as $\Delta C_1 = \Delta t \cdot f_1$, and by clock 2 as $\Delta C_2 = \Delta t \cdot f_2$. Then, if the propagation time (latency) of messages was always the same, RCF could be precisely computed as $\Delta C_2 / \Delta C_1$ of two consecutive messages, since then their inter-departure and inter-arrival interval would be the same. In reality this is not the case, so a number of obtained RCF values have to be averaged, to remove as far as possible the zero-mean

error due to the latency variation. The effects of congestion are minimized by assigning highest priority to the IEEE 1588 messages.

In the rest of this section, we introduce two methods which estimate $RCF_{S_n/S_{n-1}}$ and $RCF_{S_n/M}$ respectively RCF_{M/S_n} . Both methods are based on the timing information carried in PTP messages, but use it in different ways.

3.1 Peer RCF Estimation

The RCF of neighboring elements can be estimated using two consecutive delay estimation messages, as depicted in Fig. 3.

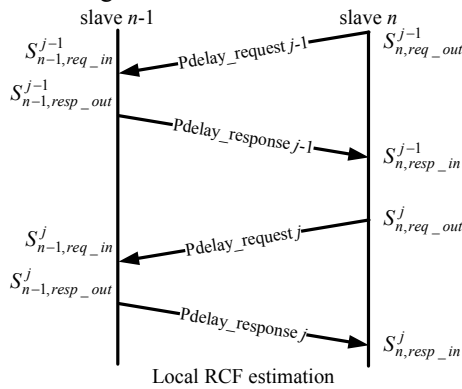


Figure 3: Peer RCF estimation.

$RCF_{S_n/S_{n-1}}$ can be calculated as:

$$RCF_{S_n/S_{n-1}} = \frac{S_n^{j, resp_in} - S_n^{j-1, resp_in}}{S_{n-1}^{j, resp_out} - S_{n-1}^{j-1, resp_out}} \quad (5)$$

Since the RCF calculated in (5) reflects the frequency difference of the neighboring elements and the estimation is only based on the message between neighboring elements, we call it peer RCF.

For the master time estimation, we need $RCF_{S_n/M}$ (or RCF_{M/S_n}), which can e.g. be calculated by using the peer RCFs calculated in the previous elements, i.e. slave 1 to $n-1$:

$$\begin{aligned} RCF_{S_n/M} &= \frac{f_{S_n}}{f_M} = \frac{f_{S_1}}{f_M} \cdot \frac{f_{S_2}}{f_{S_1}} \cdots \frac{f_{S_n}}{f_{S_{n-1}}} \\ &= RCF_{S_1/M} \cdot \prod_{i=2}^n RCF_{S_i/S_{i-1}} \end{aligned} \quad (6)$$

We call the RCF calculated this way cumulative RCF. To calculate $RCF_{S_n/M}$, slave n needs to collect all the peer RCFs in its uplink. This can be achieved recursively by modifying the Sync messages so that they contain not only the time information but also the cumulative RCF. So slave n calculates $RCF_{S_n/M}$ by multiplying the cumulative

RCF contained in the Sync message from slave $n-1$ with its peer RCF, i.e.:

$$RCF_{S_n/M} = RCF_{S_{n-1}/M} \cdot RCF_{S_n/S_{n-1}} \quad (7)$$

3.2 Master RCF Estimation

The RCF can also be estimated using exclusively the timing information contained in the Sync messages. This is illustrated in Fig. 4.

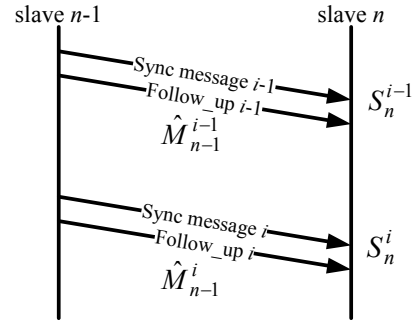


Figure 4: Master RCF estimation.

The estimation of RCF_{M/S_n} can be achieved by:

$$RCF_{M/S_n} = \frac{\hat{M}_{n-1}^i - \hat{M}_{n-1}^{i-1}}{S_n^i - S_n^{i-1}} \quad (8)$$

Since (8) calculates directly the ratio of the clock frequencies of the grand master and a slave, we call it master RCF calculation.

The frequency ratio of two neighboring slaves i.e. $RCF_{S_n/S_{n-1}}$ is then obtained as the quotient of the two master RCF values:

$$RCF_{S_n/S_{n-1}} = \frac{RCF_{M/S_{n-1}}}{RCF_{M/S_n}} \quad (9)$$

4 MASTER VERSUS PEER RCF

In this section, we will compare the two RCF calculation methods introduced in the previous section based on two criteria: convergence speed of the synchronization and the synchronization performance in the case of constant frequency drift.

4.1 Evaluation of Convergence Speed

Next we ask how much time a slave element needs to get the first correct timing information of the master since the start of the synchronization. Eq. (4) shows that an element has to have correct estimates of line delay and RCF_{M/S_n} in order to provide its downlink slave the correct master clock estimate.

Master RCF, RCF_{M/S_n} , is calculated using (8), and at least two Sync messages are needed. For correct line delay estimation, $RCF_{S_n/S_{n-1}}$ is necessary. It is calculated via (9) after RCF_{M/S_n} and $RCF_{M/S_{n-1}}$ are available. So, line delay estimates are correct after at least two Sync messages are received, and therefore a slave gets a correct master time estimate at the earliest after 3 Sync messages.

For peer RCF we first calculate $RCF_{S_n/S_{n-1}}$ using line delay estimation messages. Once $RCF_{S_n/S_{n-1}}$ is available, line delay can be calculated. Since $RCF_{S_n/S_{n-1}}$ and line delays are calculated locally, the estimation can be done in parallel, which accelerates the convergence speed. If the first Sync message is sent out when the first line delay is finished, it can carry all correct information to the slaves so that the slaves can estimate the master time correctly. Another advantage of peer RCF is its invariance to a change of (grand)master. The $RCF_{S_n/S_{n-1}}$ and line delay estimations are not affected, whereas in the master RCF calculation case, two Sync messages from the master are needed for the line delay estimation. So it always takes more time for synchronization via master RCF methods to converge if a new master is elected in the network.

4.2 Synchronization Performance for Constant Frequency Change

Next we compare the two RCF estimation methods' ability to track the frequency drift in the master.

We investigate the scenario where the master frequency is uniformly changing, e.g. due to heating, and the clock frequencies at the slaves stay constant. For analytic simplicity transmission and reception jitter is neglected, and hence the line delays can be perfectly determined. They are not neglected in our simulation in Section 6. Fig. 5 plots the frequency of each element as a function of the absolute time.

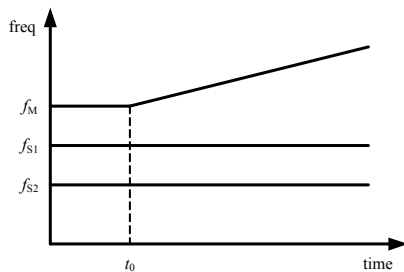


Figure 5: Frequency profile in master heating scenario.

The frequency of all elements is constant until t_0 , then the frequency of the master element increases linearly. In the case where the frequency change depends nonlinearly of the underlying cause, our analysis can be seen as a local first order approximation.

Let the slope of the frequency change of the master clock be Δ_M . So the master's frequency follows:

$$f_M(t_i) = f_M(t_{i-1}) + \Delta_M \cdot (t_i - t_{i-1}) \text{ with } t_i > t_{i-1} > t_0 \quad (10)$$

where t_i is the time when the i^{th} Sync message is transmitted by the master. The counter value increase of each element from time t_{i-1} to t_i is obtained by integrating the element's frequency over the interval (t_{i-1}, t_i) . For the slave element, whose frequency is constant, the counter value evolves as:

$$S(t_i) = S(t_{i-1}) + f_S \cdot (t_i - t_{i-1}) \quad (11)$$

For the master element, the counter value increase is calculated as:

$$\begin{aligned} M(t_i) - M(t_{i-1}) &= \int_{t_{i-1}}^{t_i} f_M(t) \cdot dt = \int_{t_{i-1}}^{t_i} [f_M(t_{i-1}) + \Delta_M \cdot (t - t_{i-1})] \cdot dt \\ &= f_M(t_{i-1}) \cdot (t_i - t_{i-1}) + \frac{\Delta_M}{2} \cdot (t_i - t_{i-1})^2 \end{aligned} \quad (12)$$

Due to the linearity of the frequency change, (12) can be alternatively expressed as the product of the frequency in the middle of the time interval times the interval length, which is sometimes a more useful form:

$$M(t_i) - M(t_{i-1}) = f_M \left(t_i - \frac{t_i - t_{i-1}}{2} \right) \cdot (t_i - t_{i-1}) \quad (13)$$

The error study for the synchronization with master RCF calculation can be found in (Na 2007, 2008), where we derive the general expression for the error in the master counter estimate of slave N , at the time when it forwards the Sync message to slave $N+1$. For simplicity of derivation, here we let all line delays and bridge delays be constant in time (the general expression can be found in (Na 2008)). Then in the time period of unchanged frequency gradient the error in the master counter estimate of slave N takes the form:

$$M - \hat{M}_{S_N, \text{out}} \Big|_{t_i + \sum_{n=1}^N LB_n^i} \approx \frac{\Delta_M}{2} \cdot \left[T \cdot \sum_{n=1}^N LB_n^i + \sum_{n=1}^N (LB_n^i)^2 \right] \quad (14)$$

where $M(t)$ is the true counter value at time t , and \hat{M} is the estimated one.

We use Fig. 6 to illustrate the error in (14). The area under the linearly rising master frequency f_M corresponds to the true master counter. The white portion thereof is the estimated master counter value at slave 2, which is the sum of the master counter value in the original Sync message plus the product of local delay times RCF estimate at each slave. It is

based only on the master frequency curve between t_{i-2} and t_i , shown solid, and holds regardless of the further gradient, shown dotted. The gray area is the estimation error in (14), which has two parts. The 1st is proportional to the time elapsed between Sync messages, and to the total delay (grey rectangles in Fig. 6); the 2nd is the sum of squares of local delays (grey triangles in Fig. 6). We see that the propagation of Sync messages let the slave elements partially follow the recent-past frequency change of the master. As the calculation of RCF uses two consecutive Sync messages, slave elements learn the trend of the frequency change of the master from the counters delivered in these two Sync messages.

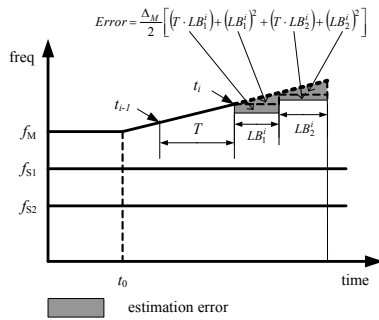


Figure 6: Sync error at the second slave (using master RCF).

For the synchronization with peer RCF, the Sync messages carry the cumulative RCF which is a product of the peer RCFs. Since the frequencies of all slaves stay constant, their peer RCFs, i.e. $RCF_{S_n/S_{n-1}}$ ($n=2 \dots N$) don't change and are:

$$RCF_{S_n/S_{n-1}} = \frac{f_{S_n}}{f_{S_{n-1}}} \quad (15)$$

A Sync message sent at time t_i by the grand master arrives at slave 1 at time $t_i + LD_1^i$. To estimate the master time, slave 1 needs the latest peer RCF and line delay estimates. Suppose that for some j with $t_0 < t_{j-1} < t_j \leq t_i$, the estimation was done based on the delay response message received at $t_{j-1} + LD_1^{j-1}$ and $t_j + LD_1^j$. Then the peer RCF between slave 1 and the grandmaster is estimated as in (11) and (13):

$$RCF_{S_1/M} = \frac{S_1(t_j + LD_1^j) - S_1(t_{j-1} + LD_1^{j-1})}{M(t_j) - M(t_{j-1})} \quad (16)$$

$$= \frac{f_{S_1} \cdot (t_j - t_{j-1} + \delta_{LD}^{j,j})}{f_M \left(t_j - \frac{t_j - t_{j-1}}{2} \right) \cdot (t_j - t_{j-1})} \approx \frac{f_{S_1}}{f_M \left(t_j - \frac{t_j - t_{j-1}}{2} \right)}$$

The error in this estimation is due to the small variation in line delays due to jitter. Inserting (15), (16) in (7), the cumulative RCF for each slave is:

$$RCF_{S_n/M} = \frac{f_{S_1}}{f_M \left(t_j - \frac{t_j - t_{j-1}}{2} \right)} \cdot \frac{f_{S_2}}{f_{S_1}} \dots \frac{f_{S_n}}{f_{S_{n-1}}} \quad (17)$$

$$= \frac{f_{S_n}}{f_M \left(t_j - \frac{t_j - t_{j-1}}{2} \right)} = (RCF_{M/S_n})^{-1}$$

The master counter value estimated at each slave when it forwards the Sync message according to (4) with the help of cumulative RCF is:

$$\hat{M}_{S_n, out} \Big|_{t_i + \sum_{n=1}^N LB_n^i} = M(t_i) + \sum_{n=1}^N \hat{S}(LB_n^i) \cdot RCF_{M/S_n}$$

$$= M(t_i) + \sum_{n=1}^N LB_n^i \cdot f_{S_n} \cdot \frac{f_M \left(t_j - \frac{t_j - t_{j-1}}{2} \right)}{f_{S_n}} \quad (18)$$

$$= M(t_i) + \sum_{n=1}^N LB_n^i \cdot f_M \left(t_j - \frac{t_j - t_{j-1}}{2} \right)$$

We have assumed that the line delay estimation is correct. The true master counter value corresponding to this time point is:

$$M \Big|_{t_i + \sum_{n=1}^N LB_n^i} = M(t_i + \sum_{n=1}^N LB_n^i) \quad (19)$$

$$= M(t_i) + f_M(t_i) \cdot \sum_{n=1}^N LB_n^i + \frac{\Delta_M}{2} \cdot \left(\sum_{n=1}^N LB_n^i \right)^2$$

Comparing (18) with (19), the estimation error using cumulative RCF is:

$$M - \hat{M}_{S_n, out} \Big|_{t_i + \sum_{n=1}^N LB_n^i} \approx \Delta_M \cdot \left(t_i - t_j + \frac{t_j - t_{j-1}}{2} \right) \cdot \sum_{n=1}^N LB_n^i + \frac{\Delta_M}{2} \cdot \left(\sum_{n=1}^N LB_n^i \right)^2 \quad (20)$$

and is shown in Fig. 7 (grey area) for the 2nd slave.

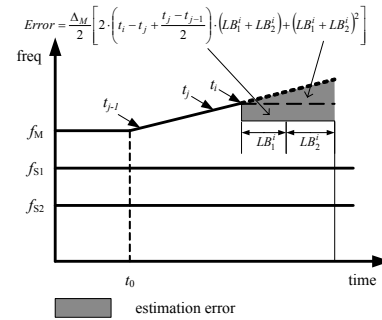


Figure 7: Sync error at the second slave (using peer RCF).

Compare the error expressions in (20) and (14): since $t_j - t_{j-1}$ (interval of delay messages) is usually greater than $T = t_i - t_{i-1}$ (interval of Sync messages), the 1st term in (20) is greater than the first term in (14). So are the 2nd terms. Our study shows that master RCF calculation performs better than peer

RCF calculation in estimating the master counter in the case of constant frequency drift in the master.

5 IMPROVED SYNTONIZATION AND SYNCHRONIZATION

In the previous section we have evaluated the performance of synchronization algorithms with peer RCF calculation and master RCF calculation. Peer RCF makes the convergence of synchronization faster, while master RCF tracks the frequency drift better. To improve the overall performance of PTP synchronization, we propose a method which combines both estimation methods.

The improved synchronization algorithm contains two phases: initial phase and steady phase. The initial phase starts at a restart. Each slave estimates peer RCF, i.e. $RCF_{S_n/S_{n-1}}$ and line delay locally using (5) and (3). The 1st Sync message is generated by the master.

Between the 1st and the 2nd Sync message there are 2 options. Either cumulative RCF is transmitted in the 1st Sync message, in which case the slave elements calculate the cumulative RCF using (7) and then estimate the master counter value using (4). This has the advantage of a convergence sped up by one Sync interval, at the cost of allowing for transmission of cumulative RCF, for which there is however enough free space in the Sync message. Or, nothing is done until the 2nd Sync message.

The steady phase begins with the 2nd Sync message. Since two Sync messages are now available, master RCF can be estimated as in (8). From the 2nd Sync message onward, master RCF will be used in (4) for the estimation of master counter value and the cumulative RCF will not be propagated any more. For the line delay estimation, we still use peer RCF calculation.

By using peer RCFs and possibly cumulative RCFs in the initial phase, the time for convergence is shortened. In the steady phase, using master RCF provides higher synchronization accuracy.

6 SIMULATION RESULTS

We have developed a MATLAB simulation tool to test and analyze the synchronization performance of IEEE 1588 in a line with cascaded bridges. We have used this tool to simulate PTP in PROFINET (Jasperneite 2005). The model parameters, summarized in Table 1, are given by the Siemens

Automation & Drive department. Comparative runs with other parameters have yielded similar results. In the simulation, the master temperature increases with a speed of 3K/s, resulting in a frequency drift of 3ppm/s. The temperature change starts at 20s, increases from 25°C to 85°C in the next 20s, then stays constant again. The frequency of slave elements never changes.

Table 1: Simulation settings.

Parameter	Value
Number of elements	80
Nominal Frequency	100MHz
Cable delay	100ns
Bridge delay	Uniform [5 15]ms
Temperature change	3K/s
Frequency Change	1ppm/K
Interval of Sync Message	32ms
Interval of Pdelay_request	8s
Interval of RCF calculation	200ms
Number of RCF averaging	7

In Fig. 8 we test the PTP synchronization with master RCF calculation, showing the synchronization errors for slaves 19, 39, 59 and 79. We observe large errors at the beginning of the synchronization. As discussed in Sect. 4.1 each element doesn't get the correct master counter value until the 3rd Sync message arrives. There is a biased error between 20 and 40s, which is caused by the constant frequency change in the master clock.

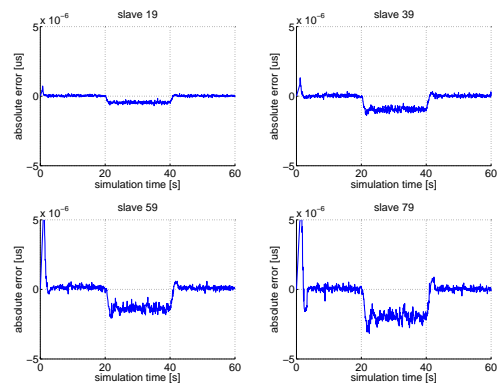


Figure 8: Synchronization error when using master RCF.

In Fig. 9 we repeat the simulation with peer RCF calculation. We see that the synchronization using peer RCF has a very smooth initial phase as the 1st Sync message already contains the correct information of RCF (by cumulative RCF) and line delay estimate. However, if we look at the time period between 20s and 40s when the frequency drift

in the master clock takes place, we observe a larger error (deviation from 0) than for the same slave in Fig. 8, which validates our analysis in Sect. 4.2.

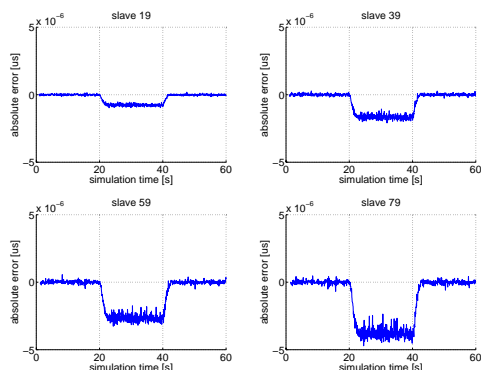


Figure 9: Synchronization error when using peer RCF.

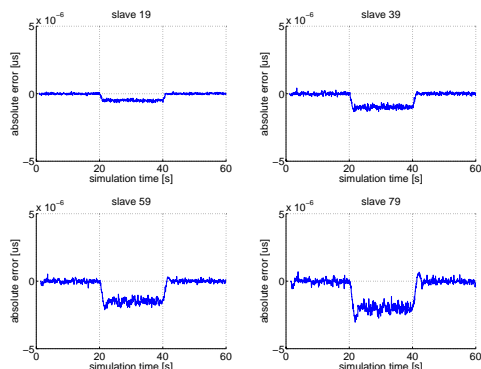


Figure 10: Synch. error with peer RCF and master RCF.

In Fig. 10 we simulate the algorithm where peer RCF and master RCF are combined. We see a better initialization compared with the result in Fig. 8 and smaller error during the frequency drift compared to Fig. 9. This confirms the improved performance we expect for the combination of peer and master RCF calculation.

7 CONCLUSIONS

In this paper, we have introduced two methods that calculate the frequency ratio of two elements based on the information contained in PTP messages. The *peer* RCF calculation utilizes delay messages locally and leads to fast convergence. The *master* RCF calculation use Sync messages to calculate the frequency ratio between the grandmaster and the slave. It performs better when there is constant frequency drift in the master clock. It has been

shown both through analysis and simulation results that a combination of both methods improves synchronization performance. Future work could illuminate the optimal combination of master RCF and peer RCF estimation for widely different system parameters or system requirements.

REFERENCES

Chen B., Chen Y.P., Xie J.M., Zhou Z.D., Sa J.M., 2005. Control methodologies in networked motion control systems. In: *Proc. of 2005 International Conference on Machine Learning and Cybernetics*, Guangzhou.

IEEE, 1997. *Standard for LAN/MAN CSMA/CD Access Method*, IEEE, New York.

IEEE, 2002. *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE, New York. ANSI/IEEE Std 1588-2002.

IEEE, 2007. *Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*, IEEE, New York.

IEEE, 2007. *IEEE P1588TM D2.2 Draft Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE, New York.

Jasperneite J., Shehab K., Weber K., 2004. Enhancements to the time synchronization standard IEEE-1588 for a system of cascaded bridges. In: *Proc. of 2004 IEEE International Workshop on Factory Communication Systems*, Vienna.

Jasperneite J., Neumann P., 2004. How to guarantee real-time behaviour using Ethernet. In: *Proc. of 11th IFAC Symposium on Information Control Problems in Manufacturing (IN-COM2004)*, Salvador-Bahia.

Jasperneite J., Feld J., 2005. PROFINET: an integration platform for heterogeneous industrial communication systems. In: *Proc. of ETFA 2005, 10th IEEE International Conference on*, Catania.

Mills, D.L., 1989. Internet time synchronization: The network time protocol. *Network Working Group Request for Comments*.

Mills, D.L., 1994. *Precision synchronization of computer network clocks*. Available: citeseer.nj.nec.com/mills94precision.html.

Na, C., Obradovic D., Scheiterer R.L., Steindl G. and Goetz F.J., 2007. Synchronization Performance of the Precision Time Protocol. In: *2007 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Vienna.

Na, C., Obradovic D., Scheiterer R.L., Steindl G. and Goetz F.J., 2008. Enhancement of the Precision Time Protocol in Automation Networks with a Line Topology. Submitted to: *IFAC 2008*, Seoul.

Nieminen, J., 2007. Synchronization of Next Generation Wireless Communication Systems. *Master thesis*, Helsinki University of Technology, Helsinki.

TEMPORAL SMOOTHING PARTICLE FILTER FOR VISION BASED AUTONOMOUS MOBILE ROBOT LOCALIZATION

Walter Nisticò and Matthias Hebbel

Robotics Research Institute (IRF), Technische Universität Dortmund, Otto-Hahn-Str. 8, Dortmund, Germany
{walter.nistico, matthias.hebbel}@uni-dortmund.de

Keywords: Particle Filters, Vision Based Localization, Real Time Systems.

Abstract: Particle filters based on the Sampling Importance Resampling (SIR) algorithm have been extensively and successfully used in the field of mobile robot localization, especially in the recent extensions (Mixture Monte Carlo) which sample a percentage of particles directly from the sensor model. However, in the context of vision based localization for mobile robots, the Markov assumption on which these methods rely is frequently violated, due to “ghost percepts” and undetected collisions, and this can be troublesome especially when working with small particle sets, due to limited computational resources and real-time constraints. In this paper we present an extension of Monte Carlo localization which relaxes the Markov assumption by tracking and smoothing the changes of the particles’ importance weights over time, and limits the speed at which the samples are redistributed after a single resampling step. We present the results of experiments conducted on vision based localization in an indoor environment for a legged-robot, in comparison with state of the art approaches.

1 INTRODUCTION

Vision-based localization is becoming very popular for autonomous mobile robots, and particle filters are a successful technique to integrate visual data for localization. Due to size, cost and power consumption limits, many robots use a camera as their only exteroceptive sensor; although range finders such as laser scanners would provide more accurate measurements, vision can often provide unique features such as *landmarks* to speed up global localization.

1.1 The Platform

This work has been developed on the Sony Aibo ERS-7 robot (Sony Corporation, 2004), which has been one of the most popular complete standard platforms adopted for robotic applications. 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 416X320 pixel. The camera is mounted on the robot head, with 3 degrees of freedom in the neck; it is severely limited in terms of resolution, field of view (56.9° horizontally), and is affected by a significant amount of noise. The Aibo production has been recently discontinued by

Sony, but several new commercially available robotic kits are being introduced, such as the humanoid robot “Nao” from Aldebaran Robotics¹, with similar characteristics in terms of size and power, often equipped with embedded RISC CPUs or PDAs and inexpensive compact flash cameras.

1.2 Related Work

Practical applications of particle filters for mobile robot localization began with the introduction of the Sampling Importance Resampling (SIR) filter proposed in (Gordon et al., 1993), (Dellaert et al., 1999) as an important extension of the Sequential Importance Sampling (SIS) filter (Geweke, 1989). The SIS filter requires a huge number of particles to work, since it does not make an efficient use of them, as the particle location is not influenced by sensor data in any way; the resampling step introduced in the SIR filter acts as a kind of “survival of the fittest” strategy that increases the particle density in the areas of highest probability of the posterior distribution. However, in the context of mobile robot localization the SIR filter suffers of two major problems: difficulty to

¹<http://www.aldebaran-robotics.com/>

recover in the “kidnapped robot” case (i.e. the robot is moved by an external agent to a new location); and *filter degeneracy*, where the filter performance starts to drop instead of improving as the sensors become “too” accurate, so that it becomes increasingly difficult to find particles close enough to the peaks of the likelihood function. Both problems can be (partially) eased by increasing the number of particles used, but this increases the computational cost; a much better solution has been proposed in (Lenser and Veloso, 2000) which samples a certain percentage of particles directly from the sensor model (“Sensor Resetting Localization” (SRL)), ensuring that no measurement gets lost due to a lack of particles. A more formal representation of the sensor resetting idea has been described in (Thrun et al., 2001) with the “Mixture Monte Carlo” (MMCL) algorithm, where a small percentage of particles are sampled from the sensor model and receive importance weights proportional to the process model, thus (unlike SRL localization) the belief converges to the true posterior for an infinite number of samples. In the context of vision based localization on autonomous mobile robots, in (Röfer and Jüngel, 2003) the authors presented an approach to deal with high levels of noise/uncertainty while using a small particle set (100 samples). Their approach keeps track of the individual importance weights of different landmark classes, and uses this information to constrain the maximum change of likelihood of the particle set in a single iteration of the algorithm, thus easily rejecting outliers in the measured data. The proposed approach is not formally correct, as we will show, however it provides surprisingly good results in practice and it is very popular on the Aibo platform, thanks also to the availability of its source code².

2 PARTICLE FILTERS

The *Particle Filter* (Fox et al., 2003) is a non-parametric implementation of the general *Bayes Filter* (Thrun et al., 2005). Given a time series of *measurements* $z_{1:t}$, *control actions* $u_{1:t}$ and an *initial belief* $p(x_0)$, the Bayes Filter is a recursive algorithm that calculates the belief or *posterior* $bel(x_t)$ at time t of the state x_t of a certain process, by integrating sensor observations z_t and control actions u_t over the belief of the state at time $t - 1$. The Bayes Filter is based on the Markov assumption or *complete state* assumption which postulates the conditional independence of past and future data given the current state x_t . This can be expressed in terms of conditional independence as

follows:

$$\begin{aligned} p(z_t|x_t, z_{1:t}, u_{1:t}) &= p(z_t|x_t) \\ p(x_t|x_{t-1}, z_{1:t-1}, u_{1:t}) &= p(x_t|u_t, x_{t-1}) \end{aligned} \quad (1)$$

A Particle Filter represents an approximation of the posterior $bel(x_t)$ in the form of a set of samples randomly drawn from the posterior itself; such a representation has the advantage, compared to closed form solutions of the Bayes Filter such as the Kalman Filter (Kalman, 1960), of being able to represent a broad range of distributions and model non-linear processes, whereas parametric representations are usually constrained to simple functions such as Gaussians. In the context of robot localization and object tracking, particle filters are often referred to as *Monte Carlo Localization*. Given a set of N samples or *particles* $\chi_t := x_t^1, x_t^2, \dots, x_t^N$, at time t each particle represents an hypothesis of the state of the observed system; in case of robot localization, the state space is usually represented by the (x, y) cartesian coordinates in the plane, and the heading θ . Obviously, the higher the number of samples N , the better the approximation, however (Fox, 2003) has shown how to dynamically adjust N . In this work we do not consider dynamic adjustments in the number of samples used, since we are interested in keeping the run-time of the algorithm approximately constant, due to the real-time constraints. An estimate of $p(x_t|u_t, x_{t-1})$

Algorithm 1 : SIR Particle Filter.

Require: particle distribution χ_{t-1} , control action u_t , measurement observation z_t

for $i = 1$ to N **do**

1. **Process update:** update the particles’ state as the result of the control action u_t : $\bar{x}_t^i \sim p(x_t|u_t, x_{t-1}^i)$

2. **Measurement update:** calculate the particle importance factors $w_t^i = p(z_t|\bar{x}_t^i)$ from the latest observation

Add $\langle \bar{x}_t^i, w_t^i \rangle$ to the temporary set $\bar{\chi}_t$

end for

3. **Resampling:** create χ_t from $\bar{\chi}_t$ by drawing the particles x_t^i in number proportional to their importance w_t^i . All the importance factors w_t in χ_t are reset to 1.

is called *Process Model*, while $p(z_t|x_t)$ is known as *Sensor Model*. $\bar{\chi}_t$ before the measurement update step is a sampled representation of the *prior distribution* $bel(x_t)$. If we omit the Resampling step in Algorithm 1 we obtain the SIS filter. Under the Markov assumption, current measurements are statistically independent from past measurements, as such the importance factors during the Measurement update step

²<http://www.germanteam.org/GT2005.zip>

can be calculated as:

$$w_t^i = p(z_t|x_t^i) \cdot w_{t-1}^i \quad (2)$$

The SIS filter for localization generally performs poorly, in fact since the importance weights are updated multiplicatively, a single percept is sufficient to set the importance of a given particle close to zero. It has been formally demonstrated that the variance of the importance weights in the SIS filter can only increase (stochastically) over time (Doucet et al., 2000). The SIR filter performs much better, because after the resampling step, the particles of low importance are discarded while the particles of high importance are replicated and all the importance weights are reset to 1, i. e. all particles carry the same importance. Consequently the distribution approximates the true posterior $bel(x_t) = \eta p(z_t|x_t) bel(x_t)$.

2.1 Sampling from Observations: The Kidnapped Robot Problem

The situation when a mobile robot is physically moved (“teleported”) by an external agent in a new location is known as the “Kidnapped Robot” problem. This situation is challenging because the robot has no information about such external action and its sensor and control data is not in accordance with the new state. The SIR filter as it has been presented has serious problems dealing with this situation, because it is unlikely to find a particle in the proximity of the location where the robot has been teleported, due to the effect of resampling which tends to concentrate the particle distribution in the area of high likelihood of the posterior at time $t - 1$, which in this case corresponds to the robot’s location prior to the teleporting action. An efficient solution to such problem is to reverse the Monte Carlo process (“*Dual Monte Carlo*”) by drawing a certain percentage of the samples directly from the measurement model instead of the prior belief distribution :

$$\bar{x}_t^i \sim p(z_t|x_t) \quad (3)$$

and calculate the importance weights from the prior distribution (integrating the process model):

$$w_t^i = \int p(\bar{x}_t^i|u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \quad (4)$$

Dual Monte Carlo performs better than normal MCL with very precise sensors, and can quickly recover in case the robot is teleported, since the samples are drawn directly from the last observations. Conversely, it is extremely sensitive in case of high sensor noise and performs poorly especially with ghost percepts; for these reasons, Mixture Monte Carlo uses a small percentage of samples which are updated using the

Dual MCL approach, and the rest are following the normal MCL algorithm, thus combining the benefits of both approaches. The main implementation problems of the Dual MCL approach is that it is often difficult to sample from the observation model (Equation 3), when such model does not have a closed form, and it is both difficult and computationally expensive to calculate the importance weights from the prior distribution (Equation 4).

3 TEMPORAL SMOOTHING PARTICLE FILTER

Particle filters can generally deal efficiently with sensor noise, provided that there is a sufficient amount of particles (most authors suggest to use at least 1000 particles) and that the measurement model is accurate enough. However a limit of this approach is that in practice the Markov assumption can often be violated: unmodeled dynamics in the environment, such as the presence of other mobile agents, “ghost” percepts (for example, echoes in sonar data, or incorrectly classified objects in a vision system), inaccuracies in the probabilistic sensor and process model (slippage, collisions), approximation errors due to the sampled representation, etc.

3.1 Sensor Smoothing

In (Röfer and Jünger, 2003) the authors proposed an approach which can deal with highly noisy visual measurements while using a sample set of just 100 particles, which can then be executed in real-time on severely constrained (in terms of processing power) mobile robots. The idea is to limit the effect of a single measurement (which might be an outlier) on the particle population; this is achieved by constraining the measured importance weight \bar{w}_t^i for a given particle at time t to differ at most by a fixed Δ from the value that the particle had at time $t - 1$.

$$w_t^i = \begin{cases} (w_{t-1}^i + \Delta_{up}) & \text{if } (\bar{w}_t^i > w_{t-1}^i + \Delta_{up}) \\ (w_{t-1}^i - \Delta_{dn}) & \text{if } (\bar{w}_t^i < w_{t-1}^i - \Delta_{dn}) \\ \bar{w}_t^i & \text{else} \end{cases} \quad (\Delta_{up} = 0.01, \Delta_{dn} = 0.005) \quad (5)$$

The visual features used for localization are separated in a set of different percept classes (field lines, borders and goals are the recognized objects in (Röfer and Jünger, 2003)), the perceived measurements of different classes are assumed to be statistically independent, and each particle carries a different importance weight per percept class. Consequently, the im-

portance weight of a particle is calculated as the product of the weights associated to each percept class. From a theoretical point of view, the smoothing performed in Equation 5 is a violation of the Markov assumption, in fact past and current measurements for objects in the same class are correlated by this update method. This can be seen as an incremental estimate of the importance weight of a given particle with respect to a series of measurements of a certain object, rather than considering each measurement independently. For this reason, this method can easily filter noise in the measurements, as well as limit the effects of outliers, since it takes a series of **consistent** measurements to affect significantly the distribution. However, this approach as presented in (Röfer and Jüngel, 2003) has some problems:

1. having $\Delta_{up} \neq \Delta_{dn}$ introduces an unnecessary bias in the importance weights;
2. since measurements relative to different objects are put in the same class, and the particles keep their “memory” indefinitely, this mechanism incorrectly correlates measurements which should be independent.

While fixing the first problem is trivial, the same is not true about the second; the correct solution would be to assign a different percept class to each perceived object, so that they can be treated as statistically independent. This however is not always possible, because in some cases the robot is observing different objects which are undistinguishable, like walls in vision based localization, or range measurements with laser scanners. Thus, here we introduce an “aging law” which fades out the memory of past measurements in the importance weights, thus reducing the incorrect correlations among percepts of the same class. This corresponds to an assumption of *temporal coherence* of a series of measurements, i. e. we expect “bursts” of measurements relative to a certain object to appear for a short time. So, if we define a set Γ of percept classes, let $w_t^{i,j}$ be the importance weight for the percept class $j \in \Gamma$ relative to the i -th particle at time t , $\alpha^j \in [0..1]$ an aging factor for class j , then the result of the aging process $\hat{w}_t^{i,j}$ can be calculated as:

$$\hat{w}_t^{i,j} = w_t^{i,j} + (1 - w_t^{i,j}) \cdot \alpha^j \quad (6)$$

This function asymptotically fades an importance factor toward 1 (i. e. that percept class does not affect the global importance factor of that particle), with a speed controlled by the parameter α^j , which can be set according to the frequency of independent observations of a given percept class. For α tending to 0, the particles keep “memory” of old percepts for a long time, while for α tending to 1 the particle weights are reset in each iteration like in the SIR filter. This is also

beneficial in case a series of past measurements for a given class resulted in a very low weight, which without aging would affect negatively the global weight of the particle, regardless of the values of the other percept classes. Finally, after the *aging step* (Equation 6), given a $\bar{w}_t^{i,j}$ calculated from the measurement model, we can apply the *temporal smoothing step*:

$$w_t^{i,j} = \begin{cases} (\hat{w}_{t-1}^{i,j} + \Delta^j) & \text{if } (\bar{w}_t^{i,j} > \hat{w}_{t-1}^{i,j} + \Delta^j) \\ (\hat{w}_{t-1}^{i,j} - \Delta^j) & \text{if } (\bar{w}_t^{i,j} < \hat{w}_{t-1}^{i,j} - \Delta^j) \\ \bar{w}_t^{i,j} & \text{else} \end{cases}$$

$$w_t^i = \prod_{j \in \Gamma} w_t^{i,j} \quad (7)$$

3.2 “lazy” Resampling

The main problem of the approach described in (Röfer and Jüngel, 2003) is the resampling step: particles are copied in the new distribution χ with a frequency proportional to their importance weight w_t^i , as in the Step 3 of Algorithm 1, but the weights $w_t^{i,j}$ are not reset to 1 afterward, this because otherwise it would not be possible to use them to filter the measurements at the following iterations of the algorithm. So, supposing that a particle with importance w_t^i is copied $n^i > 1$ times in the target distribution χ , the probability density in that location will increase by a factor of n^i . Consequently, after the resampling step the particle distribution does not approximate the true posterior anymore: the probability density will artificially increase in areas of high likelihood, while it will further decrease in areas of low likelihood. The result is that this filter converges very quickly and tends to focus most of the particles around the main peak of the probability distribution; in practice the filter works very well as long as the noise can be filtered by its sensor smoothing mechanism, but when there is a high level of ambiguity in the sensor data, for example if no unique landmark is observed in a long time, then the particles will start to quickly oscillate among different peaks of the true posterior distribution, rather than assume a multi-modal distribution. This problem can be fixed however, if we know the number of times n^i that a certain particle i has been copied. This can be used to normalize the weights in the target distribution, so that it again approximates the posterior:

$$\tilde{w}_t^i = \frac{w_t^i}{n_t^i} \Rightarrow \frac{\prod_{j \in \Gamma} w_t^{i,j}}{\prod_{j \in \Gamma} \nu_t^i} \Rightarrow \prod_{j \in \Gamma} \frac{w_t^{i,j}}{\nu_t^i} \quad (8)$$

$$\nu_t^i = \sqrt[c]{n_t^i}; c = \|\Gamma\|$$

where, given n^i and c (the number of percept classes) we can normalize the global weight of the particle \tilde{w}_t^i dividing its importance weights by the coefficient ν_t^i . As the basis for our new resampling algorithm, we use the *low variance sampler* (Thrun et al., 2005). The low variance sampler has several advantages over

Algorithm 2 : Low Variance Sampling.

Require: proposal particle distribution $\bar{\chi}_t$, importance weights W_t , number of particles N

```

 $\chi_t = \emptyset$ 
 $r = \text{rand}(0; N^{-1})$ 
 $c = w_t^1$ 
 $i = 1$ 
for  $m = 1$  to  $N$  do
     $U = r + (m - 1) \cdot N^{-1}$ 
    while  $U > c$  do
         $i = i + 1$ 
         $c = c + w_t^i$ 
    end while
    add  $x_t^i$  to  $\chi_t$ 
end for
return target distribution  $\chi_t$ 
    
```

the independent random sampler described in the basic SIR filter algorithm: it has $O(N)$ complexity (instead of $O(N \log(N))$), it guarantees that if all particles have the same importance, χ_t is the same as $\bar{\chi}_t$, and most importantly for us, it covers the set of samples systematically, making it much easier to track the values n^i to use for normalization. At this point, we can introduce a new technique, that here we will call “*lazy resampling*”, to enforce temporal coherence in the particle distribution: we can clip the number of new samples in the target distribution χ which can be generated from a single particle in the proposal distribution $\bar{\chi}$ to be at most a chosen $\bar{n}_{max} \in [1..N]$. The idea is to reduce the effect of a single resampling step on the particle distribution: if a robot is well localized and it is not teleported, we would expect that the peaks of the posterior distribution at time t are close to the position they had at time $t - 1$, factoring in the odometry data in the process update step. Especially when drawing some samples from the measurement model, as in the Sensor Resetting / Mixture Monte Carlo approaches described, a few bad measurements would move many particles away from the area of high likelihood, degrading the performance of the filter; by setting a limit to the number of times that a single particle can be copied as a consequence of resampling, we are reducing the effects of outliers on the particle distribution. The downside is that, in case the robot is really teleported, our filter would be slower in relocating most of the particles in the new

area of high likelihood: the robot would still relocate its estimate quickly, because of the update in the particle importance weights, but there would be initially less resolution in the area where the robot has been teleported. Thus, it is necessary to set a value of \bar{n}_{max} as a compromise dependent on the frequency of robot teleports and the amount of sensor noise and ambiguity. For $\bar{n}_{max} = 1$, a particle in $\bar{\chi}$ is copied unaltered in χ , as such the SIS filter can be seen as a special case of our lazy resampling; at the other end of the spectrum, setting $\bar{n}_{max} = N$ results in no limitations in the effects of resampling, so our filter behaves exactly like a SIR filter. Algorithm 3 describes a possible implementation, r is used to calculate how many times a certain particle should be copied in the target distribution based on its importance weight; such number is clipped to a maximum resample limit of \bar{n}_{max} . Particles of low weight ($n < 1$) are copied in the target distribution starting from the end of the list (indexed by j), so that they can be overwritten by the more important particles, which are copied starting from the beginning of the list (indexed by i).

Algorithm 3 : “Lazy” Resampling.

Require: proposal particle distribution $\bar{\chi}_t$, importance weights W_t , number of particles N , resample limit \bar{n}_{max}

```

 $\chi_t = \emptyset$ 
 $i = 1$ 
 $j = N$ 
 $r = \frac{N}{\sum_i w_t^i}$ 
for  $m = 1$  to  $N$  do
     $n = \min(\lfloor r \cdot w_t^m \rfloor; \bar{n}_{max})$ 
    if  $n > 0$  then
        normalize  $\tilde{w}_t^m = \frac{w_t^m}{n}$ 
        for  $z = 1$  to  $n$  do
            add  $x_t^m, \tilde{w}_t^m$  to  $\langle \chi_t, \tilde{W}_t \rangle$  in position  $i$ 
             $i = i + 1$ 
        end for
    else
        if  $j > i$  then
            add  $x_t^m, w_t^m$  to  $\langle \chi_t, \tilde{W}_t \rangle$  in position  $j$ 
             $j = j - 1$ 
        end if
    end if
end for
return target distribution  $\langle \chi_t, \tilde{W}_t \rangle$ 
    
```

4 EXPERIMENTS AND RESULTS

In the following experiments, we will evaluate the performance of our new approach (from now on referred to as Temporal Smoothing Monte Carlo “TSMCL”) which uses our *sensor smoothing with aging* (Equations 6,7) and the *lazy resampling* strategy described in Algorithm 3. We compare it with a SIR filter with Sensor Resetting, and the approach described in (Röfer and Jüngel, 2003) (here for convenience referred to as Sensor Smoothing Monte Carlo “SSMCL”).

4.1 Test Environment

The robot is placed in a soccer field whose size is $6m \times 4m$, where it can recognize 4 unique landmarks: 2 goals (1 yellow, 1 blue) and 2 cylindrical beacons (yellow on blue, and blue on yellow). The position of the landmarks on the map is known to the robot. Additionally, the robot can recognize non unique features such as the white lines on the field, and the intersections where the lines meet (see Figure 1). Since the

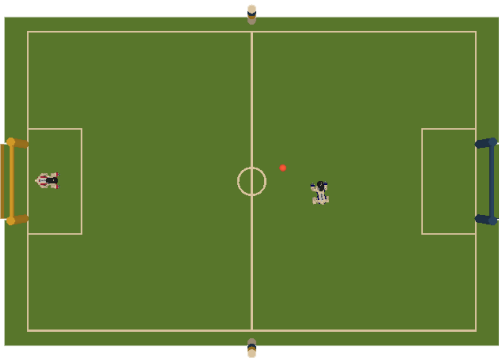
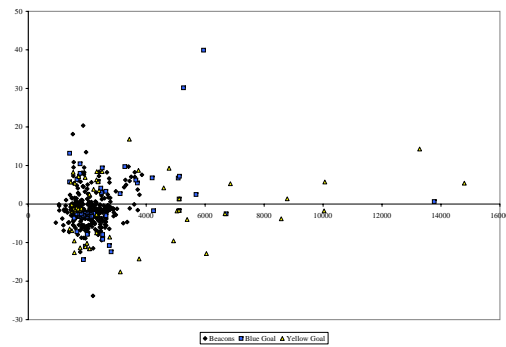


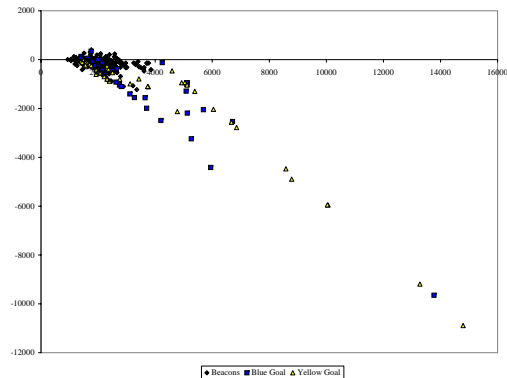
Figure 1: Test environment: a yellow and a blue goal, two colored beacons (on the middle line), two robots and an orange ball.

environment is color coded, the vision system, similar to the approach described in (Nisticò and Röfer, 2006), uses color based image segmentation. The vision and localization systems run at 30 frames per second.

As can be seen in Figure 2, the measurement error is high (especially the measured distance error grows almost exponentially with the distance to the observed object), due to the low resolution of the camera and the fact that the camera pose in space can only be roughly estimated. In fact the camera is mounted on the head of the robot, it has 3 degrees of freedom and due to the narrow field of view, it has to constantly scan around at high angular velocity. Moreover, the



(a) Bearing Error



(b) Distance Error

Figure 2: Landmark Measurement error. The bearing (degrees) and distance (mm) errors are plotted as a function of the perceived distance.

height of the neck on which the camera is mounted is not constant since the robot uses legged locomotion, and it can only be approximately estimated through kinematic calculations from the servos’ encoders in the robot legs. For these reasons our sensor model calculates the measurement error in terms of horizontal angle (*bearing*) and vertical angle (*depression*) to the observed object. This has the advantage that, while the distance error grows more than linearly with the real distance to the object, the vertical angle error variance is approximately constant. We assume the horizontal and vertical angle error components to be independent, and use gaussian likelihood functions to model each component, with different constants $\sigma_{horizontal}^j, \sigma_{vertical}^j, j \in \Gamma$ for each percept class. Field lines and their intersections (crossings) due to their small size can only be recognized reliably up to a maximum distance of approximately $800mm$; for these non-unique features we adopt a closest point matching model (with pre-computed lookup tables) as described in (Röfer and Jüngel, 2003). Due to the different measurement errors (see Table 1) and frequency of detection, we distinguish our percepts in 3 different classes: $\Gamma = \{Beacons, Goals, Lines\}$.

Table 1: Measurement error. ρ represents the distance error, in expressed in mm ; α the bearing error, in $deg.$.

Percept Class	μ_ρ	σ_ρ	μ_α	σ_α
Beacons	162.7	178.5	3.2	3.1
Goals	708.5	1294.6	4.6	4.8
Lines	122.6	215.8	3.1	3.5

All three localization algorithms use overall 100 particles, draw a small percentage of them directly from observations, and are executed in parallel in real-time on our test platform. The locations for such samples are calculated from landmarks via triangulation or using two bearings and two distances; for this purpose, the location of seen landmarks is retained for 5 seconds and updated using odometry information. Additionally, line crossings are classified as T-shaped or L-shaped, and samples are drawn from all possible locations which match the seen shape (8 at most, for L-shaped crossings, hence we use a mixture with 8% of particles drawn from observations), and two further possible poses are found if the center circle is seen. To provide ground truth to evaluate the localization performance, we use an external vision system based on a ceiling-mounted camera which is able to track the robot position at 25 frames per second with a maximum error of 40mm (position) and 2° degrees (heading). The *process model* updates the particle positions based on odometry data which is provided by the robot’s motion module; the odometry error is modeled as a bi-dimensional gaussian with the major axis parallel to the direction of movement, and an independent gaussian represents the heading error.

4.2 Experiments

In the static localization case, with the robot having unlimited time to reach a specified position on the field, all three approaches are able to localize the robot with a position error below 7 cm and heading error below 3 degrees, and in this situation it is not possible to make an analysis of the relative performance, due to measurement errors in the ceiling camera system providing ground truth and the fluctuations derived from the randomized nature of the localization algorithms employed. So in the following experiments, we will analyze the accuracy of localization while the robot runs around the field: in the first half of each experiment the robot will be allowed to look around to see unique landmarks, in the second half it will be chasing a ball looking only at it, hence seeing landmarks only occasionally, having to rely on non-unique features such as lines and crossings for localization. In the first experiment, we want to il-

lustrate the effects of our *lazy resampling*, by comparing the performance of our algorithm for different values of the *resample limit* \bar{n}_{max} : 8 (the experimentally derived optimum for our application), 15 and 30 (for all practical purposes, equivalent to “unlimited”, given a set of 100 particles). As can be seen in Figure 3, all three versions localize about at the same time (≈ 180 frames or 6 seconds), then they perform equally well, as long as the robot receives reasonably good percepts. However, in the situations where the robot looks only at the ball for a long time, the version with $\bar{n}_{max} = 8$ performs much better, being more effective in smoothing the noise and resolving ambiguities by keeping more particles in the areas where the robot was previously localized. In the sec-

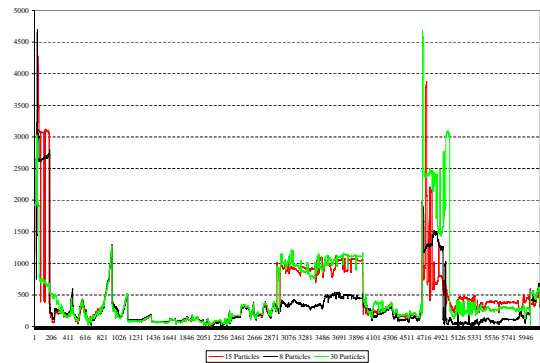
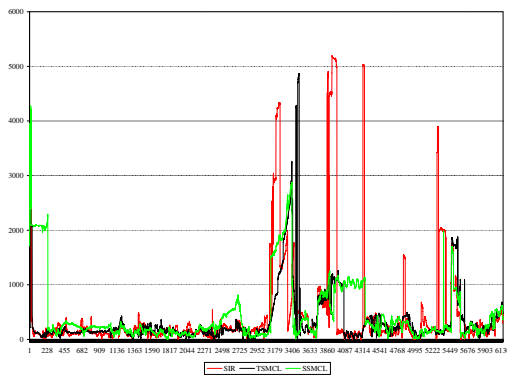


Figure 3: Resample limit comparison (Test 1). x -axis: time (frames); y -axis: position error. At frames 2900-3950 and 4700-5100 the scarcity of unique features and the odometry errors make the localization jump to a wrong location.

ond experiment, we compare our TSMCL algorithm with $\bar{n}_{max} = 8$ to the state of the art approaches. In the first half of the experiment, all systems perform equally well, with TSMCL and SIR nearly identical and SSMCL slightly worse but within the limits due to random factors. When the robot starts to chase the ball however (after frame 3000), SIR and SSMCL start to oscillate much more (Figure 4). Our third test is similar to the second, but it represents a worse scenario since this time the beacons (which are the best landmark, in terms of measurement error and ease of detection) are removed. The results presented in Figure 5 show our approach clearly outperforming the others. The run-time of all 3 algorithms is 3–5ms per frame, depending on the number and type of percepts seen. We could not measure significant differences in the time taken to globally localize and recover from kidnapping (2 – 6s in all cases).

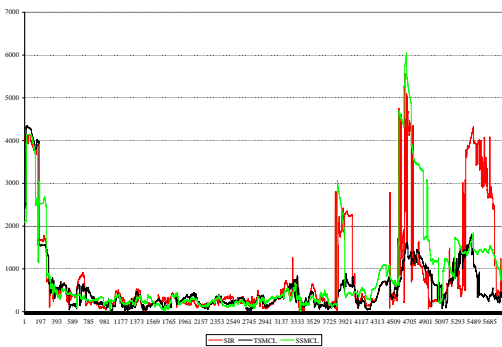


(a)

	First Half			Overall		
	μ	σ	max	μ	σ	max
TSMCL	144.8	62.8	430	323.6	503.0	4864
SSMCL	201.9	130.3	813	444.1	553.1	2875
SIR	146.2	76.1	548	436.7	897.1	5195

(b)

Figure 4: Localization algorithm comparison (Test 2). x -axis: time (frames); y -axis: position error.



(a)

	First Half			Overall		
	μ	σ	max	μ	σ	max
TSMCL	277.5	160.1	847	432.7	355.0	1811
SSMCL	276.3	128.6	867	745.8	998.1	6051
SIR	294.2	169.0	1261	706.9	993.7	5567

(b)

Figure 5: Localization algorithm comparison (Test 3). x -axis: time (frames); y -axis: position error.

5 FUTURE WORK

We have presented an extension of Monte Carlo Localization which exploits temporal smoothing to improve localization accuracy in presence of high amounts of noise and sensor ambiguity. In the future we would like to apply our algorithm to multi-object tracking, where its use of temporal coherence could allow it to “keep memory” of multiple modes in the distribution more effectively than normal parti-

cle filters, making it a good candidate to compete with banks of multiple filters with degrees of association.

REFERENCES

- Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA99)*.
- Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–208.
- Fox, D. (2003). Adapting the sample size in particle filters through kld-sampling. *I. J. Robotic Res.*, 22(12):985–1004.
- Fox, D., Hightower, J., Liao, L., Schulz, D., and Borriello, G. (2003). Bayesian Filtering for Location Estimation. *PERVASIVE computing*, pages 10–19.
- Geweke, J. (1989). Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 57(6):1317–39.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45.
- Lenser, S. and Veloso, M. (2000). Sensor resetting localization for poorly modelled mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2000*.
- 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. and Jünger, M. (2003). Vision-Based Fast and Reactive Monte-Carlo Localization. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA 2003, September 14-19, 2003, Taipei, Taiwan*, pages 856–861. IEEE.
- Sony Corporation (2004). *OPEN-R SDK Model Information for ERS-7*. Sony Corporation.
- Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. Number ISBN 0-262-20162-3. MIT Press.
- Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141.

SLAM AND MULTI-FEATURE MAP BY FUSING 3D LASER AND CAMERA DATA

Ayman Zureiki, Michel Devy and Raja Chatila

CNRS, LAAS, 7 avenue du Colonel Roche, F-31077 Toulouse, France

Université de Toulouse, UPS, INSA, INP, ISAE, LAAS-CNRS, F-31077 Toulouse, France

{ayman.zureiki, michel.devy, raja.chatila}@laas.fr

Keywords: Data Fusion, SLAM, Heterogeneous Maps.

Abstract: Indoor structured environments contain an important number of planar surfaces and line segments. Using these both features in a unique map gives a simplified way to represent man-made environments. Extracting planes and lines by a mobile robot requires more than one sensor: a 3D laser scanner and a camera can be a good equipment. The incremental construction of such a model is a Simultaneous Localisation And Mapping (SLAM) problem: while exploring the environment, the robot executes motions; from each position, it acquires sensory data, extracts perceptual features, and simultaneously, performs self-localisation and model update. First, the 3D range image is segmented into a set of planar faces which are used as landmarks. Next, we describe how to extract 2D line landmarks by fusing data from both sensors. Our stochastic map is of heterogeneous type and contains plane and 2D line landmarks. At first, The SLAM formalism is used to build a stochastic planar map, and results on the incremental construction of such a map are presented, further on, heterogeneous map will be constructed.

1 INTRODUCTION

Simultaneous Localisation and Mapping is a fundamental technology for autonomous mobile robots. A robot needs a description of his environment. Maps are required for self-localisation, for motion planning, etc. In this article, we deal with the on line learning of such maps for a structured (man-made) environment supposed unknown.

Using the SLAM algorithm, the robot performs a complex process, including the execution of motions, the acquisition of sensory data, data association between these sensory data and the current world model, estimation of the robot pose using these associations and finally, the incremental construction of the map. It has to take into account many geometric constraints, and many sources of errors. Essentially, the robustness to achieve this task depends on the robot capabilities to extract pertinent information (called Landmarks) from sensory data coming from embedded sensors. The robot starts up from an initial position without any a priori knowledge about landmarks: by use of relative measurements on landmarks, the robot estimates its pose and poses of the landmarks in an absolute frame, generally selected as the initial pose of the robot. When moving, the robot updates the landmark map and exploits it to produce an estimate of its pose. The delivered map can be of

intuitive representation for humans or not. In the literature we can find three main types of maps. Topological, metric and hybrid maps. A *topological* map can be seen as an abstract representation describing relations between environment areas (typically, rooms or corridors). Such maps are well adapted for route planning, the selection of the best strategy for motions between areas. Their main drawback is the absence of geometric information. On the contrary, a *metric* map provides a (detailed) geometric representation of the environment; it gives explicit metric information (lengths, widths, positions, etc.), generally expressed with respect to a global reference frame. The third class is the *Atlas* which is a *Hybrid* metric/topological approach to SLAM capable to achieve efficient mapping of large-scale environments (Bosse et al., 2003).

SLAM has been an active research topic for more than twenty years; many works from Durrant-White, Tardos, Nebot, Dissanayake, Feder, Leonard, Newman, Rencken... aim to develop generic tools, based on the formalism of **stochastic maps** proposed by (Smith et al., 1990). The majority of these works have focused on the estimation methods required in order to maintain estimates of the robot pose and of landmark attributes in a consistent stochastic map. The extended Kalman Filter (EKF) was initially proposed as a mechanism that allows the incremental fusion of

information acquired by the robot; later, other methods have been exploited successfully (information filter, particle filter etc.), especially in the FastSLAM method, proposed by (Thrun et al., 1998). A well detailed state of the art can be found in (Durrant-Whyte and Bailey, 2006).

These approaches have been validated mainly by constructing 2D representations (2D segment maps etc.) of indoor environment from laser data acquired typically by SICK range finders. Recently, 3D SLAM draws attention. (Takezawa et al., 2004) describes a SLAM framework based on 3D landmarks. (Jung, 2004) constructs a 3D map from interest points in outer environment using stereo vision data; (Sola et al., 2005) builds such maps using only monocular vision. These sparse representations allow essentially the robot to locate itself. Our work is focused on the construction of surface model in indoor environment, where many planar surfaces (ceiling, floor, walls, doors etc.) can be used as landmarks. Our goal is to produce a geometric stochastic map made of 3D planar features. In the same area, let us cite the preliminary contribution of (Nashashibi and Devy, 1993), with an off line validation from a limited number of range images, and the works of (Thrun et al., 2000) based on the exploitation of two laser ranger finders to acquire measurements on horizontal and vertical planes and to produce a dense model of 3D points, from which a mesh can be constructed a posteriori. (Abuhadrous et al., 2004) developed a similar approach to model urban sites using GPS to localise the vehicle. Finally using only monocular vision, planes are extracted by using homographs and fused by a SLAM approach in (Silveira et al., 2006).

While the algorithm of SLAM is well known and studied, using new sensors and robust features extraction remains an open topic. Sensors' data fusion is an interesting approach to overcome the deficiency of each sensor and to obtain more sophisticated and accurate results. In this paper, we present a novel type of *Heterogeneous* multi-feature metric maps. Our map contains two types of features: 3D Planes and 2D lines attached to these planes. While planes are extracted from 3D point cloud issue from 3D laser scanner, the 2D lines are extracted by fusing data from the laser scanner with image data from a camera.

In the section 3 we give details about features extraction: planar features from range images, and 2D line segments from both laser and camera data. Then in section 4, we define our heterogeneous map which contains plane and 2D line landmarks. Next in section 5 we adapt the slam algorithm for both of used features. Finally in section 6, experimental results using our mobile robot (Jido) are discussed, before summa-

rizing our contribution and presenting current works in section 7.

2 NOTATION

As we use many reference frames and two features, it is useful to summarise used notations. Let $\mathbf{R}_{1,2}$ and $\mathbf{t}_{1,2}$ be the rotation matrix and the translation vector from reference frame 1 to frame 2. For a 3D point represented by \mathbf{P}_1 in the frame 1 and by \mathbf{P}_2 in the frame 2, we have:

$$\mathbf{P}_1 = [x_1 \quad y_1 \quad z_1]^T \quad (1)$$

$$\mathbf{P}_1 = \mathbf{R}_{1,2} \mathbf{P}_2 + \mathbf{t}_{1,2} \quad (2)$$

Used Frames.

- \mathcal{R}_{sk} : SICK frame.
- \mathcal{R}_c : camera frame.
- \mathcal{R}_r : robot frame.
- \mathcal{R}_w : global (world) frame.
- \mathcal{R}_p : plane landmark local frame.

The transformations between these frames are given by the following matrices and vectors:

- $\mathbf{R}_{r,sk}$ and $\mathbf{t}_{r,sk}$: from robot to SICK frames.
- $\mathbf{R}_{r,c}$ and $\mathbf{t}_{r,c}$: from robot to camera frames.
- $\mathbf{R}_{w,r}$ and $\mathbf{t}_{w,r}$: from world to robot frames.
- $\mathbf{R}_{w,p}$ and $\mathbf{t}_{w,p}$: from world to plane landmark frames.

The robot pose is defined by $(x_v, y_v, \theta_v)^T$ in the world frame.

3 FEATURES EXTRACTION

We detail in this section the extraction of used landmarks from sensory data.

3.1 Plane Extraction

3D laser scanner provides range images with thousands of 3D points. Segmenting the range image means how to divide it into features, i.e. how to bind each point with a label identifying to which feature it belongs, so that points of the same plane have all the same label. For a mobile robot, segmenting range images is a difficult topic, because the robot does not know a priori what is seen in the scene; moreover segmentation process must be robust in presence of

non-planar or non static objects and in spite of measurements' noises. The planar segmentation has been well studied in computer graphics in order to perform real-time rendering of complex models (Heckbert and Garland, 1997). A major difference exists between robotics and computer graphics. Data in robotics are issued from sensors and hence they are erroneous, while models in computer graphic are supposed to be without errors.

(Hähnel et al., 2003) proposed a simplification algorithm adapted to robotic context. They extract planes by using an approach of type *region-growing* by starting from an arbitrary point, then try to enlarge the region in all directions. (Weingarten, 2006) proposed some improvement to this algorithm by starting *region seed* from the most flat point in the cloud (minimum local error), and by profiting from the structure of the range image to simplify the research of neighbour points. Our approach is based on these two works, with some differences in the choice of plane's parameters and the method of their estimation.

3.1.1 Plane Equation

A plane can be represented by three parameters: the distance from the origin ρ and two angles. Let ϕ be the angle between the projection of the plane normal on the OXY plane and the axis \overrightarrow{OX} , and let ψ be the angle between the plane normal with the axis \overrightarrow{OZ} . The plane equation is then:

$$\cos \phi \sin \psi x + \sin \phi \sin \psi y + \cos \psi z + \rho = 0 \quad (3)$$

The vector $(\rho \ \phi \ \psi)^t$ will be used as the minimal parametric representation of a plane.

3.1.2 Estimation Process

Kalman Filter is a recursive estimator : to estimate the current state, only the previous state and the actual measurements are required. The observation history is not needed. In the Extended Kalman Filter (EKF), the dynamic and observation models could be non-linear functions. To estimate the parameters of a plane we use and EKF. We consider that each point belonging to a plane as an observation of this plane. We detail the estimation process in (Zureiki and Devy, 2008).

3.1.3 Choice of Plane Landmark Local Reference

Let \mathcal{P} be a plane landmark defined by its parameters (ρ_w, ϕ_w, ψ_w) in the global frame \mathcal{R}_w . We are looking for a orthonormal frame of this plane. We choose the projection of the origin O_w on the plane \mathcal{P} as an origin

O_p of local frame, and the axis Z_p to be parallel to the normal vector \mathbf{n} . We need to choose the axis X_p . Let $\vec{i}_w, \vec{j}_w, \vec{k}_w$ be the unit vectors of axes O_wX, O_wY, O_wZ respectively, and $\vec{i}_p, \vec{j}_p, \vec{k}_p$ be unit vectors of wanted axes O_pX_p, O_pY_p, O_pZ_p respectively.

$$\vec{i}_p = [\sin \phi_w \quad -\cos \phi_w \quad 0]^T \quad (4)$$

This vector can be interpreted as the unit vector of direction of the intersection line between the plane \mathcal{P} and the plane $Z = 0$ (if they are not parallel). The rotation matrix from global to the local references is:

$$\mathbf{R}_{w,p} = \begin{bmatrix} \sin \phi_w & \cos \phi_w \cos \psi_w & \cos \phi_w \sin \psi_w \\ -\cos \phi_w & \sin \phi_w \cos \psi_w & \sin \phi_w \sin \psi_w \\ 0 & -\sin \psi_w & \cos \psi_w \end{bmatrix} \quad (5)$$

and the translation vector is:

$$\mathbf{t}_{w,p} = \rho_w \begin{bmatrix} \cos \phi_w \sin \psi_w \\ \sin \phi_w \sin \psi_w \\ \cos \psi_w \end{bmatrix} \quad (6)$$

3.2 2D Line Landmark Extraction

By mean of a camera we extract 2D segments in the image. These segments can be interpreted as the projection of 3D Lines (or more generally Planes) onto the image plane. The first idea to come is to use the 3D Lines as a second type of landmarks in the stochastic map. To define a 3D line we need to define two planes. Using the camera, we can obtain one of them, so we need to use the 3D laser to define the other plane. By fusing the data of both sensors we can extract 3D lines in the scene. For representation reasons, we will consider the 3D line as 2D line attached to a holding plane. The holding plane is define by the laser data (as describe in 3.1). This representation seems reasonable, because the 3D line may be either a corner (intersection of two planes) or the borders of a poster fixed on a wall for example, and in both cases a 2D line in the local plane frame is sufficient to totally define it. As a result, we need to add to the map only the parameters defining a 2D line in the plane landmark frame. Therefore, the second type of Landmarks for us is a **2D Line attached to a Plane Landmark**.

3.2.1 Advantages of Data Fusion

Data fusion is the technique of combining data from multiple sensors or information from different sources to achieve more specific inferences than could be

reached by using a single independent sensor. Fused data provide several advantages over single sensor data (Hall and Llinas, 2001). First, if several identical sensors are used, combining the observations will result in an improved estimate of the observed quantity. A statistical advantage is gained by adding N independent observations. The same result could be obtained also by combining N observations from an individual sensor. A second advantage involves using the relative placement of multiple sensors to improve the observation process. For example, two sensors (camera) that observe the same object can coordinate to determine the 3D position of the object by triangulation (stereo vision). A third advantage of using multiple sensors is improved observability. For a robot equipped by a 3D laser scanner and a camera, the laser scanner can accurately determine the range of an obstacle (wall for example), camera can determine the visual properties of the obstacle but can not determine its range. By using the camera we can recognise whether the robot is in front of a wall or a closed door. If these two observations are correctly associated, the combination of the two sensors provides a better localisation than could be obtained by either of the two independent sensors.

In our work, we use the 3D laser scanner to extract planes from the 3D range images. A camera is used to extract 2D lines in images. By combining the camera-laser data, we define a new landmark as : 2D line attached to a 3D plane landmark. This 2D line landmark could be seen as graduations on a ruler, while the ruler defines the plane in the 3D space, the graduations on it define more information with respect to the ruler plane. So with these two landmarks, a robot can be localised with respect to a plane and with respect to the graduations (2D line landmarks) on this plane. The importance of such landmarks can be more illustrated in a long corridor formed with two walls (and eventually with closed doors). Using only plane landmarks will lead to only two parallel planes. Using a camera to extract 2D lines in the image (may be borders of a poster fixed on the wall, or the borders of a door), and fusion laser-camera data will provide a 2D line fixed on the wall in a precise position. The robot will be localised with respect to both plane and line: the plane will help to find latitude information, while the 2D line will add longitude information.

3.2.2 2D Line Extraction in Images

We use a traditional method of line extraction in images. It begins by a Canny filter to extract the contour, then we use a polygonal approximation to estimate the line passing through adjacent contour points. A phase of post processing is necessary to merge simi-

lar segments and to remove very small ones.

3.2.3 Interpretation Plane

For a line segment l_I in the image, the associated *Interpretation plane* is the plane passing through this 2D line and the centre of projection (viewpoint) of the camera. The normal vector of this plane can be calculated only based on intrinsic parameters of the camera $(\alpha_u, \alpha_v, u_0, v_0)$ and the data image of the segment. In fact, let (δ_I, γ_I) be the 2D line parameters of the infinite line holding the 2D segment l_I , where γ_I is the angle with the axis u and δ_I is the distance from the origin. The 2D line equation is in the image reference frame:

$$\cos \gamma_I u + \sin \gamma_I v - \delta_I = 0 \quad (7)$$

Then using camera coordinates:

$$\cos \gamma_I \left(\alpha_u \frac{x_c}{z_c} + u_0 \right) + \sin \gamma_I \left(\alpha_v \frac{y_c}{z_c} + v_0 \right) - \delta_I = 0 \quad (8)$$

we obtain:

$$\alpha_u \cos \gamma_I x_c + \alpha_v \sin \gamma_I y_c + (-\delta_I + u_0 \cos \gamma_I + v_0 \sin \gamma_I) z_c = 0 \quad (9)$$

The normal vector in the camera reference frame is given by:

$$\mathbf{n}_c = \begin{bmatrix} \alpha_u \cos \gamma_I \\ \alpha_v \sin \gamma_I \\ -\delta_I + u_0 \cos \gamma_I + v_0 \sin \gamma_I \end{bmatrix} \quad (10)$$

and the distance to the origin in camera frame $d_c = 0$.

The interpretation plane in the robot, world, plane landmark frames is noted respectively by: (\mathbf{n}_r, d_r) , (\mathbf{n}_w, d_w) and (\mathbf{n}_p, d_p) . We note also:

$$\begin{cases} \mathbf{n}_c &= [n_{c,x} \ n_{c,y} \ n_{c,z}]^T \\ \mathbf{n}_r &= [n_{r,x} \ n_{r,y} \ n_{r,z}]^T \\ \mathbf{n}_w &= [n_{w,x} \ n_{w,y} \ n_{w,z}]^T \\ \mathbf{n}_p &= [n_{p,x} \ n_{p,y} \ n_{p,z}]^T \end{cases} \quad (11)$$

3.2.4 The 2D Line in the Plane Landmark Frame

The interpretation plane in the plane landmark frame is given by:

$$\begin{cases} \mathbf{n}_p &= \mathbf{R}_{w,p}^T \mathbf{R}_{w,r} \mathbf{R}_{r,c} \mathbf{n}_c \\ d_p &= d_c - \mathbf{t}_{r,c}^T \mathbf{R}_{r,c} \mathbf{n}_c - \mathbf{t}_{w,r}^T \mathbf{R}_{w,r} \mathbf{R}_{r,c} \mathbf{n}_c \\ &+ \mathbf{t}_{w,p}^T \mathbf{R}_{w,r} \mathbf{R}_{r,c} \mathbf{n}_c \end{cases} \quad (12)$$

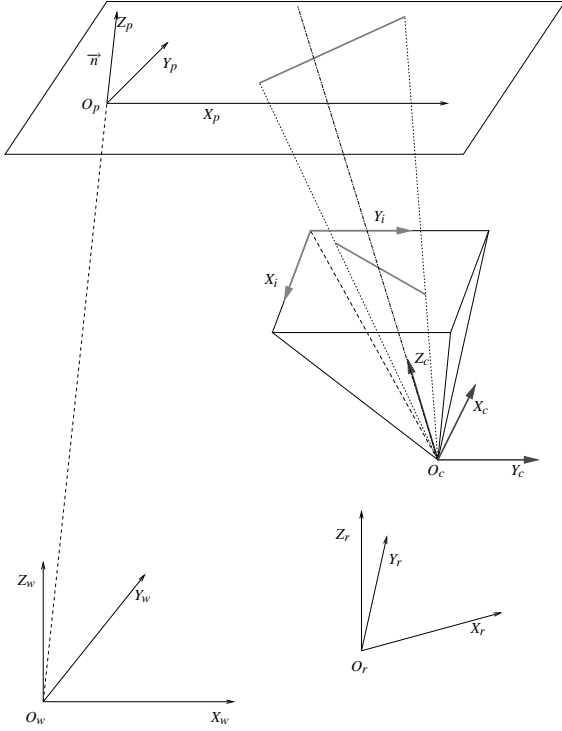


Figure 1: A 2D Segment in image and its corresponding 2D Line Landmark attached to the Plane Landmark.

Hence, we obtain the normal vector and the distance from the origin (\mathbf{n}_p, d_p) of the interpretation plane in the plane landmark frame. The 3D line of the intersection between the interpretation plane with the plane $z_p = 0$ can be seen as 2D line in the plane $O_p X_p Y_p$, therefore, the 2D line landmark equation in the plane landmark frame is:

$$n_{x,p} x_p + n_{y,p} y_p + d_p = 0 \quad (13)$$

4 THE STOCHASTIC MAP

Indoor environment can be considered (in a simplified way) as a set of planar surfaces which we choose as landmarks for the SLAM algorithm. Attached to these plane surfaces we consider another type of landmarks: the 2D lines.

The SLAM algorithm maintains a representation of both the environment state and the robot state. During the robot displacement, it uses its sensors to observe the surrounding landmarks. The system state at time k , $\mathbf{X}(k)$, is composed of the robot state \mathbf{X}_v , and of n_f vectors describing the observed landmarks, $\mathbf{X}_i(k)$, $i = 1, \dots, n_f$.

$$\mathbf{X}(k) = [\mathbf{X}_v^W \quad \mathbf{X}_1 \quad \dots \quad \mathbf{X}_{n_f}]^T \quad (14)$$

where \mathbf{X}_i is the state of a landmark i either in the global frame \mathcal{R}_W if it is a plane landmark or in holding plane frame if it is a 2D line landmark. We can rearrange the system state vector so that we group the states of landmarks in one term $\mathbf{X}_m(k)$:

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{X}_v^W \\ \mathbf{X}_m^W \end{bmatrix} \quad (15)$$

The robot state at time k can be determined by its position and orientation in the space. The robot state vector is defined by: $\mathbf{X}_v(k) = [x_v(k), y_v(k), \theta_v(k)]^T$. Each planar surface (wall, ceiling, floor etc.), is considered as an infinite plane defined by three parameters $\mathbf{X}_{\pi,j}(k) = [\rho_j(k), \phi_j(k), \psi_j(k)]^T$. Each 2D Segment is considered as an infinite line in the holding plane landmark and is defined by means of two parameters $\mathbf{X}_{L,i}(k) = [\delta_i(k), \gamma_i(k)]^T$. Of course, a plane landmark can contain many 2D line landmarks, but a 2D line landmark can not exist alone without a holding plane landmark. Our stochastic map is then an heterogeneous map, as it has two types of landmarks.

5 THE SLAM ALGORITHM

We detail the main steps in the SLAM algorithm adapted to the used landmarks.

5.1 Prediction

The prediction phase of the extended Kalman filter uses the dynamic model of the robot to produce an estimate of the robot motion $\hat{\mathbf{X}}_v(k|k-1)$, at time k knowing all the information until time $k-1$, and the control input $u(k)$:

$$\hat{\mathbf{X}}_v(k|k-1) = f(\hat{\mathbf{X}}_v(k-1|k-1), u(k)) \quad (16)$$

We can write the prediction phase of the filter as:

$$\begin{bmatrix} \hat{\mathbf{X}}_v(k|k-1) \\ \hat{\mathbf{X}}_m(k|k-1) \end{bmatrix} = \begin{bmatrix} f(\hat{\mathbf{X}}_v(k-1|k-1), u(k)) \\ \hat{\mathbf{X}}_m(k-1|k-1) \end{bmatrix} \quad (17)$$

The covariance matrix must propagate through the robot model in this phase. The extended Kalman filter linearise the propagation of the uncertainty around the current estimate $\hat{\mathbf{X}}(k-1|k-1)$ by using the Jacobean $\nabla_{\mathbf{X}} f(k)$ of f at $\hat{\mathbf{X}}(k-1|k-1)$. $\mathbf{Q}(k)$ is the covariance of the error.

$$\mathbf{P}(k|k-1) = \nabla_{\mathbf{X}} f(k) \mathbf{P}(k-1|k-1) \nabla_{\mathbf{X}} f^T(k) + \mathbf{Q}(k) \quad (18)$$

For the SLAM algorithm, this phase can be simplified thanks to the hypothesis that the landmarks are fix. This let us reduce the calculation complexity of the prediction covariance to only the calculation of covariance of robot pose and cross-covariance between the robot and the map (Williams, 2001).

5.2 Observation of 3D Plane Landmarks

For plane landmarks, the innovation function is:

$$\mathbf{v} = \mathbf{Z}(k) - \hat{\mathbf{Z}}(k|k-1) \quad (19)$$

where:

- $\mathbf{Z}(k)$: the current measurement, i.e. the extracted planes from the 3D range image in the robot frame, with a covariance matrix $\mathbf{R}(k)$.
- $\hat{\mathbf{Z}}(k|k-1)$: estimation of measurement, i.e. how the plane landmarks in the stochastic map are positioned with respect to the current robot pose in robot frame. Hence the measurement estimation is a function of the predicted state of the system:

$$\hat{\mathbf{Z}}(k|k-1) = h(\hat{\mathbf{X}}(k|k-1)) \quad (20)$$

Measurement Estimation. In the case of plane/plane fusion: Let $(\rho_w, \varphi_w, \psi_w)$ and $(\rho_r, \varphi_r, \psi_r)$ be the parameters of a plane in the global and robot frame respectively. For a robot moving on horizontal floor, the relation between a plane parameters in the global and robot frame is:

$$\begin{cases} \rho_r &= \rho_w - \cos \varphi_w \sin \psi_w x_v \\ &\quad - \sin \varphi_w \sin \psi_w y_v \\ \varphi_r &= \varphi_w - \theta_v \\ \psi_r &= \psi_w \end{cases} \quad (21)$$

Then, the observation prediction of a plane landmark relatively to the robot frame:

$$\hat{\mathbf{Z}}(k|k-1) = \begin{bmatrix} \hat{\rho}_r(k|k-1) \\ \hat{\varphi}_r(k|k-1) \\ \hat{\psi}_r(k|k-1) \end{bmatrix} \quad (22)$$

5.3 Observation of 2D Line Landmarks

The procedure of updating the stochastic map is the following: First we update the plane landmarks using only the 3D laser data. Then, we update the 2D line landmarks on each plane.

Innovation Function. With 2D line landmarks attached to plane landmarks, the innovation function is written:

$$\mathbf{v} = \mathbf{Z}(k) - \hat{\mathbf{Z}}(k|k-1) \quad (23)$$

where:

- $\mathbf{Z}(k)$: the current measurement of the 2D line landmark attached to a plane. In reality, we obtain this measurement by fusing 3D laser and camera data. Hence the current measurement is a function of the system state. This measurement is in plane landmark local frame.

$$\mathbf{Z}(k) = h(\hat{\mathbf{X}}(k|k-1), Image) \quad (24)$$

- $\hat{\mathbf{Z}}(k|k-1)$: estimation of measurement, i.e. how the 2D line landmarks attached to a plane landmark are positioned in the local plane frame. In fact, the 2D line landmarks are fix with respect to the plane holding them

$$\hat{\mathbf{Z}}(k|k-1) = \hat{\mathbf{Z}}(k-1|k-1) \quad (25)$$

Innovation Covariance Matrix.

$$\mathbf{S}(k) = \mathbf{P}_{ii}(k|k-1) + \mathbf{R}(k) \quad (26)$$

where $\mathbf{P}_{ii}(k|k-1)$ is the covariance matrix of the 2D line landmark i , and $\mathbf{R}(k)$ is the covariance matrix of the current measurement. See the previous note, we write:

$$\mathbf{Z}(k) = h(\hat{\mathbf{X}}_v(k|k-1), \hat{\mathbf{X}}_j(k|k-1), \mathbf{X}_I(k)) \quad (27)$$

where $\hat{\mathbf{X}}_j(k|k-1)$ is the predicted state of holding plane landmark, and $\mathbf{X}_I(k)$ is the camera data (image 2D line parameters) with a covariance matrix Λ_I .

With $\nabla_v h$, $\nabla_j h$ and $\nabla_I h$ are the Jacobians of the function h with respect to the robot, plane landmark j and 2D line in Image I respectively, we have:

$$\mathbf{R}(k) = \begin{aligned} &\nabla_v h \mathbf{P}_{vv} \nabla_v h^T + \nabla_v h \mathbf{P}_{vj} \nabla_j h^T \\ &+ \nabla_j h \mathbf{P}_{jj} \nabla_j h^T + \nabla_j h \mathbf{P}_{vj}^T \nabla_v h^T \\ &+ \nabla_I h \Lambda_I \nabla_I h^T \end{aligned} \quad (28)$$

In this last equation, we identify clearly the role of camera data which are not correlated to laser data. This allows us to justify our choice for 2D Line Landmarks attached to plane landmark. In spite of the intuitive idea that the 2D line landmark will be correlated to the plane landmark holding it, a part of data (camera data) defining the 2D line is not correlated with the data defining the holding plane. This proves that the two landmarks are not correlated.

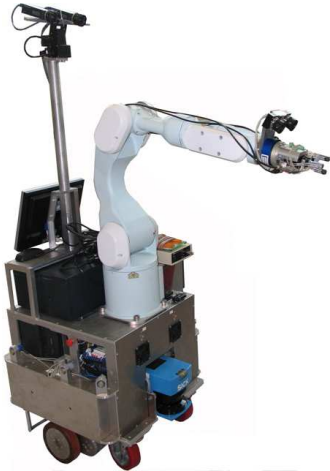


Figure 2: The mobile robot Jido.

5.4 Update

Once an observation is associated to a landmark in the map, the estimate of system state can be updated using the gain matrix $\mathbf{W}(k)$. The gain matrix provides a weighted sum of the prediction and observation, and is calculated based on the innovation covariance matrix $\mathbf{S}(k)$, and the prediction of covariance matrix, $\mathbf{P}(k|k-1)$. The weighting factor is proportional to $\mathbf{P}(k|k-1)$ and inversely proportional to innovation covariance (Smith et al., 1990). This can be used to update the system state vector $\hat{\mathbf{X}}(k|k)$ and its covariance matrix $\mathbf{P}(k|k)$.

$$\hat{\mathbf{X}}(k|k) = \hat{\mathbf{X}}(k|k-1) + \mathbf{W}(k)\mathbf{v}(k) \quad (29)$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k) \quad (30)$$

where

$$\mathbf{W}(k) = \mathbf{P}(k|k-1)\nabla_x h^T \mathbf{S}^{-1}(k) \quad (31)$$

6 IMPLEMENTATION AND RESULTS

We used in the experiments our robot JIDO (figure 2). It has (among other sensors) a SICK LMS-200 Range finder fixed on a rotating axis installed ahead, a stereo rig on a pan/tilt.

The 3D scanner laser has an angular resolution of 0.5° , with a field of view of 180° which gives 361 points per scan. For the rotation of scanner around the horizontal axis, we choose to make steps of 0.01 Rad ($\approx 0.57^\circ$) and to rotate the scanner between -0.3 Rad ($\approx -17^\circ$) and 1.4 Rad ($\approx 80^\circ$), which



Figure 3: Experimental Results.

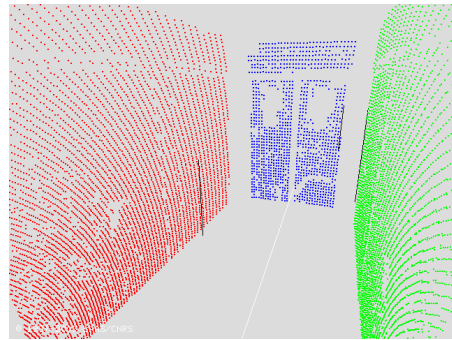


Figure 4: Plane Landmarks and some 2D Line Landmarks attached to them.

includes 171 scans. The produced range image is composed of $171 * 361 = 61731$ points. The left camera of the stereo rig is used to acquire images.

The robot did a tour in our laboratory, it moves and halts, takes measurements, then it advances again. It has made a tour in a corridor and did a half turn to return to the point of departure, making in all 12 displacements. We implement a classical EKF SLAM algorithm. The incremental construction of the map of the corridor is illustrated (partially) in the figure 3, where we choose to print only the points belonging to each planar facet in the stochastic map. The points of each plane are assembled from the successive fused planes' points. The texture was mapped onto the planes using a homography transformation from initial image and a virtual image placed on the plane.

Figure 4 represent an image with the extracted 2D line segments, and figure 5 represents 3D planes extracted from laser scanner and on which are attached to some 2D line landmarks coming from the fusion algorithm explained in this article. For now only plane landmarks are added to the stochastic map, the addition of 2D line landmarks is under construction.



Figure 5: Image with the extracted 2D line Segments.

7 CONCLUSIONS

This paper has described an heterogeneous 3D stochastic map building using a SLAM method. The map has 3D plane landmarks and 2D line landmarks. Features extraction is detailed with the emphasis on the fusion of laser and camera data to obtain 2D line landmarks. Preliminary results on 2D line landmarks was presented, as well as a map reconstructed only with planar landmarks. The current work is to achieve the building of the heterogeneous map.

Adding 2D lines to planes has two major benefits: make the map more rich for navigation, and at the same time enforce the phase of data association of plane landmarks.

Due to the acquisition time using the laser scanner sensor, the robot must stop during the scanning: the same method will be applied with continuous acquisition made from a PMD sensor (Swiss Ranger from the CSEM company) mounted on the mast of our robot.

REFERENCES

- Abuhadrous, I., Ammoun, S., Nashashibi, F., Goulette, F., and Laugeau, C. (2004). Digitizing and 3d modelling of urban environments using vehicle-borne laser scanner system. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W., and Teller, S. (2003). An atlas framework for scalable mapping. In *International Conference on Robotics and Automation (ICRA03)*, pages 1899–1906.
- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous Localization and Mapping (SLAM): Part I & II. *IEEE Robotics & Automation Magazine*.
- Hähnel, D., Burgard, W., and Thrun, S. (2003). Learning compact 3d models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*.
- Hall, D. L. and Llinas, J., editors (2001). *Handbook of Multisensor Data Fusion*. CRC Press LLC.
- Heckbert, P. S. and Garland, M. (1997). Survey of polygonal surface simplification algorithms. Technical report, Carnegie-Mellon Univ.
- Jung, I. K. (2004). *Simultaneous localization and mapping in 3D environments with stereovision*. PhD thesis, Institut National Polytechnique de Toulouse, France.
- Nashashibi, F. and Devy, M. (1993). 3d incremental modeling and robot localization in a structured environment using a laser range finder. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.
- Silveira, G., Malis, E., and Rives, P. (2006). Real-time robust detection of planar regions in a pair of images. In *Proc. IEEE/RSJ International Conference on Intelligent Robots Systems, Beijing, China*.
- Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. *Autonomous robot vehicles*, pages 167–193.
- Sola, J., Monin, A., Devy, M., and Lemaire, T. (2005). Undelayed initialization in bearing only slam. In *Proc. IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS)*, pages 2751–2756.
- Takezawa, A., Herath, D. C., and Dissanayake, G. (2004). Slam in indoor environments with stereo vision. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Thrun, S., Burgard, W., and Fox, D. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3):29–53.
- Thrun, S., Fox, D., and Burgard, W. (2000). A real-time algorithm for mobile robot mapping with application to multi robot and 3d mapping. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*.
- Weingarten, J. (2006). *Feature-based 3D SLAM*. PhD thesis, École Polytechnique Fédérale de Lausanne.
- Williams, S. (2001). *Efficient Solutions to Autonomous Mapping and Navigation Problems*. PhD thesis, The University of Sydney.
- Zureiki, A. and Devy, M. (2008). Slam and data fusion from visual landmarks and 3d planes. In *Proc. the 17th IFAC World Congress*.

TORQUE CONTROL WITH RECURRENT NEURAL NETWORKS

Guillaume Jouffroy

Artificial Intelligence Laboratory, University Paris 8, France

gj@ai.univ-paris8.fr

Keywords: Joint constraint method, oscillatory recurrent neural network, generalized teacher forcing, feedback, adaptive systems.

Abstract: In the robotics field, a lot of attention is given to the complexity of the mechanics and particularly to the number of degrees of freedom. Also, the oscillatory recurrent neural network architecture is only considered as a black box, which prevents from carefully studying the interesting features of the network's dynamics. In this paper we describe a generalized teacher forcing algorithm, and we build a default oscillatory recurrent neural network controller for a vehicle of one degree of freedom. We then build a feedback system as a constraint method for the joint. We show that with the default oscillatory controller the vehicle can however behave correctly, even in its transient time from standing to moving, and is robust to the oscillatory controller's own transient period and its initial conditions. We finally discuss how the default oscillator can be modified, thus reducing the local feedback adaptation amplitude.

1 INTRODUCTION

Central Pattern Generators (CPG) are biological periodic oscillatory neural networks responsible for a wide range of rhythmic functions. They can be made of endogeneous oscillatory neurons connected to non oscillatory ones or from the sole interaction between non oscillatory neurons.

Particularly, they are a great source of inspiration in the robotics field, for the control of joints in locomotion. In general, an oscillatory network controls a joint angle in both directions, and the phase relationships needed between all joints arise from the coupling between the different networks.

The needed parameters for an artificial Recurrent Neural Network (RNN) to have a periodic oscillatory behavior cannot be measured experimentally. This network is most of the time a relatively simplified model of its biological counterpart when available. Only clinical temporal data of joints kinetics and kinematics can be of use, where however it is difficult to isolate the real control of a particular joint from the influence of the others.

In the case of non endogeneous oscillatory neurons, in the literature, parameters are thus mainly determined empirically or with genetic algorithms (Buono and M.Golubitsky, 2001), (Ghigliazza and P.Holmes, 2004), (Ishiguro et al., 2000), (Kamimura

et al., 2003), (Taga, 1994), (Ijspeert, 2001), comparatively to relatively few learning methods (Mori et al., 2004), (Tsung and Cottrell, 1993), (Weiss, 1997). Though this is useful with large networks in complex mechanical models, there are two drawbacks. It is very difficult in general to isolate the resulting dynamics of the different networks, and to understand their interaction to each other and with the mechanical system dynamics. Also, it is not clear how to modify such networks in an adaptive context, e.g. in the case of a permanent constraint change on a joint due to injury.

Based on this considerations, we apply a *generalized* formulation of the so called *teacher forcing* gradient descent-based learning algorithm, to create an oscillatory RNN as a *torque controller* for an interesting vehicle with one single degree of freedom, the Roller Racer. The RNN is put in a closed loop with the Roller Racer, such that the vehicle can be freely controlled, where the RNN can be modified permanently.

The paper is structured as follows. In section 2, we briefly present the Roller Racer model and we show how it can be controlled with a torque input. In section 3 we describe the control system. The subsection 3.1 presents the generalized formulation of the teacher forcing learning algorithm with which we build the oscillatory RNN as a basic torque controller

for the Roller Racer vehicle. In subsection 3.2 we describe the local feedback control that can be built so that the vehicle direction can be controlled, limiting the effect of its transient state. We discuss how the basic oscillatory system can be In section 4, we give concluding remarks and discuss how the basic oscillatory system can be modified to better fit the needs of the Roller Racer, thus reducing the local feedback adaptation amplitude.

2 VEHICLE MODEL

The Roller Racer is a toy-vehicle with one single degree of freedom which is the handlebar. The direction wheels are shifted back from the the axis. Thus, oscillating the handlebar from side to side, a component of the reaction force on the ground which points backward is created, moving forward the vehicle.

In (Jouffroy and Jouffroy, 2006), we revisited and synthesized a mathematical model of the Roller Racer from the original work of Krishnaprasad and Tsakiris (Krishnaprasad and Tsakiris, 1995). The input control was the angle of the axis. Here, we will describe the torque input control formalization.

Recall the state of the Roller Racer vehicle is $\mathbf{x} \triangleq (\theta_r, x_r, y_r, p, \theta_c, \dot{\theta}_c)^T \in \mathbb{R}^6$, and its dynamics is $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u)$, where

$$\mathbf{f}(\mathbf{x}, u) \triangleq \begin{pmatrix} \frac{1}{\Delta(\theta_c)} (\sin \theta_c p - \delta(\theta_c) \dot{\theta}_c) \\ \frac{\cos \theta_r}{\Delta(\theta_c)} (\chi(\theta_c) p - \gamma(\theta_c) \sin \theta_c \dot{\theta}_c) \\ \frac{\sin \theta_r}{\Delta(\theta_c)} (\chi(\theta_c) p - \gamma(\theta_c) \sin \theta_c \dot{\theta}_c) \\ [A_1(\theta_c) \dot{\theta}_c - C_1(\theta_c)] p + \\ [A_2(\theta_c) \dot{\theta}_c + C_2(\theta_c)] \dot{\theta}_c \\ \dot{\theta}_c \\ u \end{pmatrix}, \quad (1)$$

θ_c is the angle of the handlebar, p is the momentum of the vehicle and (x_r, y_r) are the rear coordinates respectively to the global reference space. Friction constraint is built in the model through the functions $C_1(\theta_c)$ and $C_2(\theta_c)$. Thus one does not need to deal with mechanical aspects, leaving focus on the control strategy, and on the learning aspects of the RNN.

Here we control the Roller Racer using the torque input control T_c with the following equation

$$\ddot{\theta}_c = u = B_1(\theta_c) \theta_c p + B_2(\theta_c) \dot{\theta}_c^2 + B_3(\theta_c) T_c \quad (2)$$

The right hand side of the equation replaces u in (1). The parameters are defined as

$$\begin{aligned} B_1 &\triangleq -\frac{A_2(\theta_c)}{\Delta_1(\theta_c)} \\ B_2 &\triangleq \frac{m_1 \gamma(\theta_c) \sin \theta_c}{\Delta(\theta_c) \Delta_1(\theta_c)} [\gamma(\theta_c) \cos \theta_c + d_1 \delta(\theta_c)] \\ B_3 &\triangleq \frac{\Delta(\theta_c)}{\Delta_1(\theta_c)}, \end{aligned}$$

where

$$\Delta_1 \triangleq I_1 I_2 \sin^2 \theta_c + m_1 (I_1 d_2^2 + I_2 d_1^2 \cos^2 \theta_c),$$

and the other parameters are as defined in (Jouffroy and Jouffroy, 2006).

3 DESIGN OF THE OSCILLATORY CONTROLLER

3.1 The Oscillatory Recurrent Neural Network Torque Controller

Let us consider the RNN system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{W}), \quad (3)$$

with $\mathbf{x} \in \mathbb{R}^n$ is the state vector of the network, $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the matrix of the weight connexions, w_{ij} to be considered as the weight from the neuron i to the neuron j . We consider a *fully* connected RNN, which means all neurons are interconnected and self-connected ($w_{ii} \neq 0$). For the neuron model we use the rate based neuron model of the simplest form

$$\mathbf{f}(\mathbf{x}, \mathbf{W}) = (\mathbf{I} \tau^{-1}) (-\mathbf{x} + \mathbf{W} s(\mathbf{x})), \quad (4)$$

with $s(x)$ a squashing function such as $\tanh(x)$. \mathbf{I} is the identity matrix and $\tau \in \mathbb{R}^n$ the time constant vector of the system.

Each component x_i^* of the teacher vector \mathbf{x}^* is of the form

$$x_i^* = \sin(t + \phi_i) \quad (5)$$

The learning is achieved when an error criterion E , $E \in \mathbb{R}^n$ is less or equal than a minimum $\varepsilon \in \mathbb{R}$, $\varepsilon \simeq 0$

$$E = \frac{1}{2} (\mathbf{x} - \mathbf{x}^*) \circ (\mathbf{x} - \mathbf{x}^*) < \varepsilon, \quad (6)$$

the operator \circ being the Hadamard product.

The weight matrix \mathbf{W} is ajusted according to the following gradient rule

$$w_{pq}^* = -\eta \sum_{i=1}^n \frac{\partial E}{\partial x_i} z_{pq}^i, \quad (7)$$

with $\eta \in \mathbb{R}$ is the learning rate. $\mathbf{z} \in \mathbb{R}^{n \times n^2}$ is the sensitivity of the state of the system with respect to a weight w_{pq} , which can be written in the matrix form

$$\frac{d\mathbf{z}}{dt} = (\mathbf{I}\tau^{-1})(\mathbf{J}_{\mathbf{f}_x}(M)\mathbf{z} + \mathbf{J}_{\mathbf{f}_w}(M)), \quad (8)$$

where $\mathbf{J}_{\mathbf{f}_x}(M)$ and $\mathbf{J}_{\mathbf{f}_w}(M)$ are the jacobian matrices of the function \mathbf{f} respectively to \mathbf{x} and \mathbf{w} at the point M . In the teacher forcing case (8) reduces to

$$\frac{d\mathbf{z}}{dt} = (\mathbf{I}\tau^{-1})(-\mathbf{I}\mathbf{z} + \mathbf{J}_{\mathbf{f}_w}(M)), \quad (9)$$

For convenience \mathbf{z} is of the form

$$\begin{bmatrix} z_{11}^1 & \cdots & z_{nn}^1 \\ \vdots & \ddots & \vdots \\ z_{11}^n & \cdots & z_{nn}^n \end{bmatrix} \quad (10)$$

Providing a target signal(s) x_i^* only for some neuron(s) i , letting $\mathbf{J}_{\mathbf{f}_x} = \mathbf{A}$, the sensitivity equation (8) can be written as

$$A_{ij} = -a_{ij} + b_{ij}w_{ji} \frac{\partial s(x_j)}{\partial x_j}, \quad (11)$$

with $a_{ij} = 1$ when $i = j$, 0 otherwise, $b_{ij} = 1$ when i is not a forced neuron, 0 otherwise. $\mathbf{J}_{\mathbf{f}_w} = \mathbf{B}$ is of the same form as \mathbf{z} and its elements are such that $B_{pq}^i = 0$ when $i \neq q$, $B_{ij} = s(x_p^*)$ if p is a forced neuron, $B_{ij} = s(x_p)$ otherwise.

With this algorithm we build a 4 neurons fully connected RNN. Teacher signals have an amplitude of 1, and we choose the phase difference vector ϕ^* as $\{0; \pi/3; 2\pi/3; \pi\}$. To generate the torque control we use the output of the neurons 1 and 4 which are in opposite phase, to control each direction of the handlebar angle using the transformation

$$T_c = \text{pos}(x_1) - \text{pos}(x_4) \quad (12)$$

The use of 3 neurons could have been the minimum acceptable to solve the phase difference of π between the output neurons. But in the scope of recover, with at least 4 of them if one break, we have the opportunity to start the algorithm again and obtain the needed oscillator. The data obtained for the weight matrix \mathbf{W} of the estimated oscillator needed is

$$\mathbf{W} = \begin{bmatrix} 0.493 & -0.18 & -0.673 & -0.493 \\ 0.958 & 0.882 & -0.076 & -0.958 \\ 0.465 & 1.062 & 0.597 & -0.465 \\ -0.493 & 0.18 & 0.673 & 0.493 \end{bmatrix}$$

Convergence is reached in at most 300 timesteps, with $\eta = 0.1$.

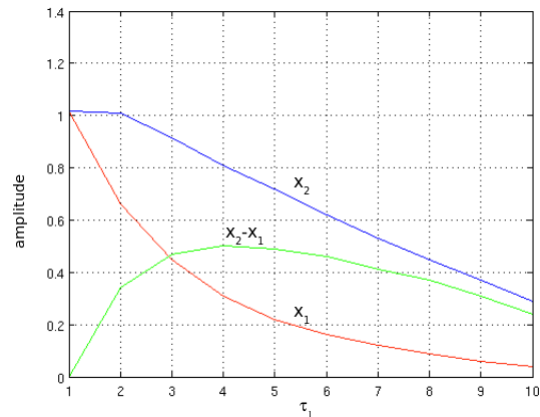


Figure 1: Effect of the modification of the time constant τ_1 on the state x_1 in the oscillatory neural network. τ_2 is kept fixed ($\tau_2 = 1$).

3.2 Feedback Design of the System

The torque amplitude of the oscillatory controller, considering its frequency, may be too large for the needs of the Roller Racer. Beside, the starting energy for a vehicle is generally different from when it is at full speed. So is the Roller Racer with the torque control. It needs a transient oscillatory input which might depend on friction biases and inertia. Thus we create an angle feedback from the Roller Racer to the oscillatory controller, which purpose is to constrain the angle within some limits. We control the vehicle in the forward direction which means the average angle of the handlebar should be π (see (Jouffroy and Jouffroy, 2006)).

To apply a correction to the torque generated, we use the feedback to modify the time constants τ_i of the oscillatory controller's output neurons. In Figure 1 we illustrate the effect on the amplitude of the output neurons state when changing only one time constant (τ_1 on the figure). As might be expected, the state of each neuron output decreases. However the correction applies more to x_1 , and the amplitude difference is maximum at $\tau \approx 4$. It is not really really desirable to have one output which becomes zero as it prevents the Roller Racer to get energy. Therefore we should not have τ_i being set too high.

We now define the transfer function, that constrains the angle within the boundaries $[\pi - 1; \pi + 1]$ and with a correction applied when $\tau_i < 5$, considering the effect on the amplitude reduced above this limit. Note that the frequency is relatively not modified in this range (a frequency modification would take place if all τ_i where equally changed). The transfer function \mathbf{g} for the feedback is defined as

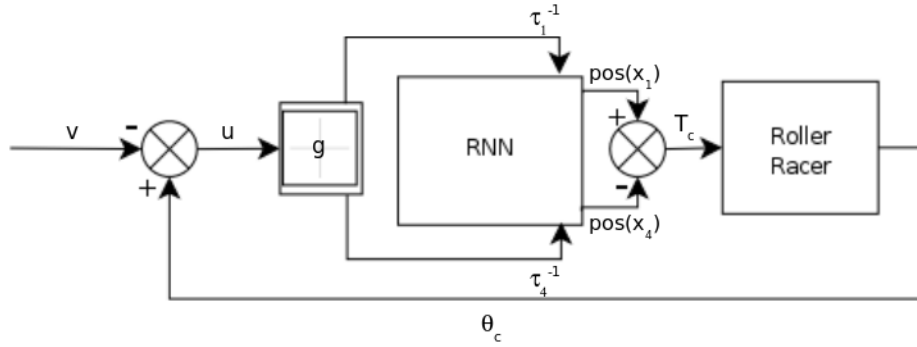


Figure 2: Control architecture of the Roller Racer. T_c is the torque control applied to the vehicle. v is the angular input which is obtained from the user direction control δ .

$$\mathbf{g}(u) = \begin{bmatrix} g_1(u) \\ g_4(u) \end{bmatrix}, \quad (13)$$

with

$$g_1(u) = \frac{6}{1 + e^{-(6u-4.5)}}, \quad (14)$$

g_1 being the transfer function for the feedback to the neuron 1, and $g_4(u) = -g_1(u)$ to the neuron 4.

The formalization of these transfer functions has been chosen so that they can be used as “feedback” neurons, with the same neuron model as in (3), replacing f_i by g_i .

The signal $u \in \mathbb{R}$ is the actual feedback signal which is the difference between the angle of the handlebar θ_c (in radians) and the desired average angle control $v \in \mathbb{R}$. For the forward direction u is actually set as

$$u = \theta_c - v = \theta_c - (\pi + \delta), \quad (15)$$

where $\delta \in \mathbb{R}$ is the user direction input, which is actually the continuous component control of the RNN.

The feedback information thus obtained is used to modify the time constants τ_1 and τ_4

$$\tau_1^{-1} = \frac{1}{1 + g_1(u)} \text{ and } \tau_4^{-1} = \frac{1}{1 + g_4(u)} \quad (16)$$

The whole architecture is summarized in Figure 2.

3.3 Results

We present here the results of two different trials. In both of them the purpose is to have a straight trajectory along the x axis of a physical space reference. The architecture is of course able to freely control the vehicle in all directions but the simulations are not shown.

In the first trial we start the RNN with a little energy given to a neuron. The neuron 3 in the following simulation has the initial condition $x_3 = 0.1$.

In Figure 3 left, is plotted the angle of the handlebar θ_c which continuous component is π , for the forward direction. It clearly shows the transient time when the system is extracting itself from reaction forces, until it reaches its permanent speed at around 40 timesteps. The transient period has an amplifying oscillation around π because the RNN is also in its transient state with little energy.

This transient activity of the RNN, is shown by the very weak correction applied from the feedback $g(u)$ in Figure 3 right, and the low speed along the x axis in Figure 3 bottom. One can clearly see on this graphics of the right side of the figure, a little deviation during this time. This is only the drawback of the transient state of the RNN which does not provide a symmetric gain, even if the angle is within the boundaries $[\pi - 1; \pi + 1]$.

The correction applied once the vehicle is in its permanent state shows that the RNN’s own oscillation is not optimal and that a relearning could fix this. The trajectory however becomes quite straight.

In the second trial we initialize the RNN with a strong gain to see how the feedback behaves during the transient period. We set $x_3 = 1$.

In Figure 4 left we can see that the amplitude of the transient state of the RNN has been pushed too high. The angle of the handlebar θ_c does not show anymore an amplifying oscillatory behavior, and reach the boundaries we have specified. The correction from the feedback apply a high gain correction (see Figure 4 right).

After $t \approx 40$, as in the first trial, the symmetric oscillations are recovered, and the correction reduces to the steady-state time in Figure 3 right. Interestingly the correction has constrained the deviation well, which is not higher than in the first trial, except in the transient period. The vehicle also gets speed earlier and the trajectory is also straight (Figure 4 bottom).

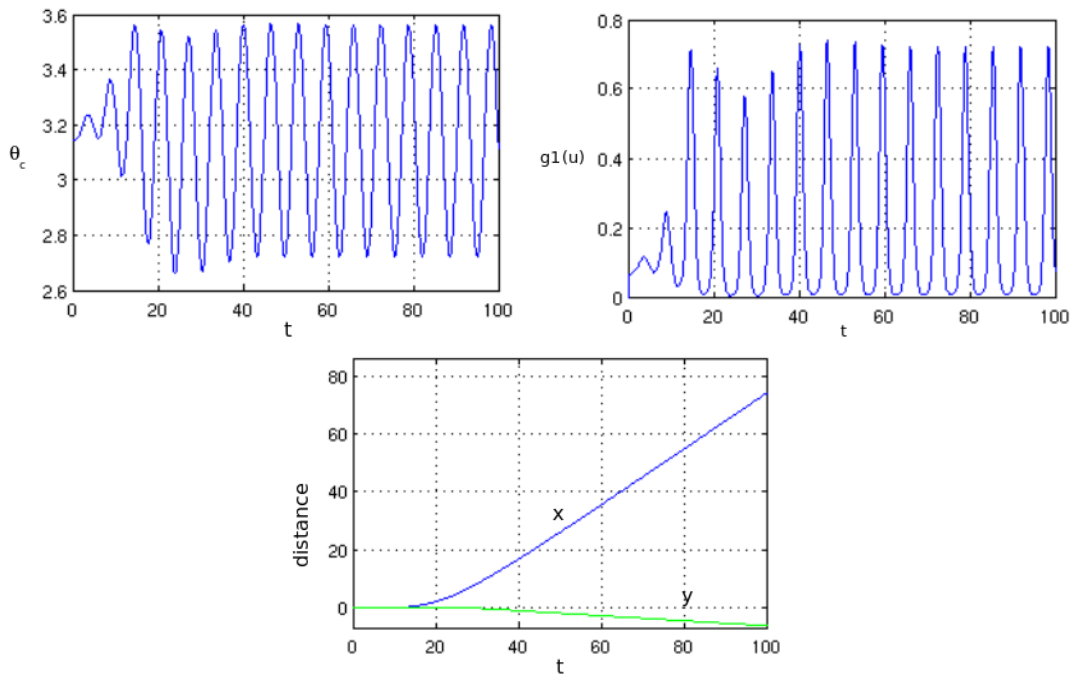


Figure 3: Simulation results when the RNN is initialized with a weak energy. Left: angle of the handlebar θ_c . Right: feedback correction $g_1(u)$. Bottom: trajectory evolution. During and after the transient times, little correction is applied.

4 CONCLUSIONS AND DISCUSSION

Nature is a great source of inspiration for engineers who deal with autonomous robots. Evolution has found optimally-designed solutions for robustness and adaptability in a changing environment, which are exciting to discover. However, most of the research in the control aspects of robots with neural networks tackle the question of complex mechanics with many degrees of freedom, and massive neural architectures, which appear as “black boxes” designed by genetics algorithms. This hides the dynamics of the neural system and correlatively the opportunity to constitute adaptive strategies.

In this work, we present a generalized version of the teacher forcing learning algorithm, to build up an estimated oscillatory controller for a vehicle with one degree of freedom, the Roller Racer. We create an angular feedback such that the degree of freedom is constrained within some boundaries. The purpose is to prevent the vehicle to go out of control during its transient state when it starts moving, as a consequence of the oscillator being not adapted to this particular moment.

Our simulation results show that the feedback makes the vehicle to behave relatively well during transient state, when the oscillator is initialized with a

weak energy or even a strong one. The deviation also stays little. When the steady-state period is reached, the vehicle moves in a straight line as expected.

From a design point of view, the correction applied by the feedback system to the RNN, never completely vanishes. This shows that a more optimal oscillatory behavior can be obtained, though the “default” one does not critically affect the system with the help of a *mutual entrainment* between the vehicle dynamics and the controller, as described first by (Taga, 1994).

We are currently studying how an adaptive process or observer, can modify permanently the RNN when the average correction is too high. The outputs of the network, with the help of the feedback, could be the desired targets for a second network which could thus be trained in parallel, with a partial teacher forcing. However this is highly computationally expensive, and not biologically viable.

The teacher forcing principle is made such that an oscillatory behavior can be obtained with a gradient descent algorithm. However, forcing the outputs of the network means disconnecting it, and thus loosing the interesting desired target obtained with feedback. Algorithms without direct gradient descent evaluation techniques may be more appropriate (for e.g. (Kailath, 1990)).

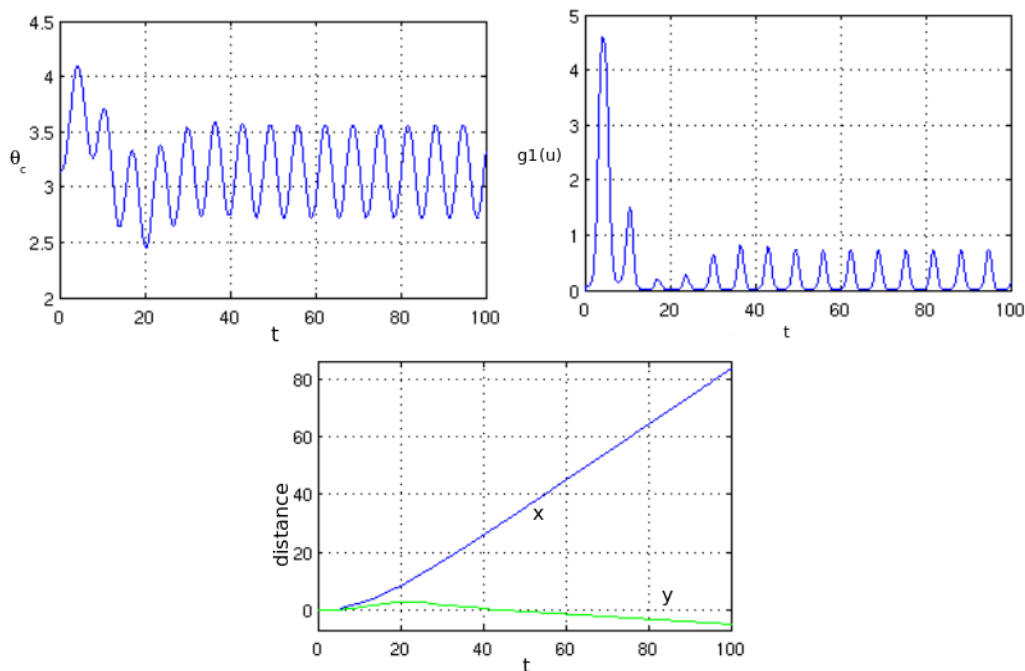


Figure 4: Simulation results when the RNN is initialized with a stronger energy. The correction is high during the transient state. The deviation is not higher than in the first trial which shows the effective action of the feedback.

Beside, the *constraint method* we used in this article can have some interest when studying the coupling of oscillatory neural networks, for e.g. to synchronize different joints. When we attach an oscillatory RNN to another one which has a constraining feedback, we can find coupling parameters which do not yield an increase of the correction in the feedback. This constraint method thus helps to reduce the space to search for suitable coupling parameters, and to better match the desired phase relationship.

REFERENCES

- Buono, P. and M. Golubitsky (2001). Models of central pattern generators for quadruped locomotion. *Journal of Mathematical Biology*, 42:291–326.
- Ghigliazza, R. and P. Holmes (2004). A minimal model of a central pattern generator and motoneurons for insects locomotion. *SIAM Journal on Applied Dynamical Systems*, 3(4):671–700.
- Ijspeert, A. (2001). A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, 84:331–348.
- Ishiguro, A., Otsu, K., Fujii, A., Uchikawa, Y., Aoki, T., and Eggenberger, P. (2000). Evolving and adaptive controller for a legged-robot with dynamically-rearranging neural networks. In *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, Cambridge, MA. MIT Press.
- Jouffroy, G. and Jouffroy, J. (2006). A simple mechanical system for studying adaptive oscillatory neural networks. *IEEE International Conference on Systems, Man and Cybernetics*, pages 2584–2589.
- Kailath, A. D. . T. (1990). Model-free distributed learning. *IEEE Trans. Neural Networks*, 1(1):58–70.
- Kamimura, A., Kurokawa, H., Yoshida, E., Tomita, K., Murata, S., and Kokaji, S. (2003). Automatic locomotion pattern generation for modular robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 714–720.
- Krishnaprasad, P. and Tsakiris, D. (1995). Oscillations, se(2)-snakes and motion control. New Orleans, Louisiana.
- Mori, T., Nakamura, Y., Sato, M., and Ishii, S. (2004). Reinforcement learning for cpg-driven biped robot. *Nineteenth National Conference on Artificial Intelligence*, pages 623–630.
- Taga, G. (1994). Emergence of bipedal locomotion through entrainment among the neuro-musculo-skeletal system and the environment. *Physica D*, 75:190–208.
- Tsung, F. and Cottrell, G. (1993). Phase-space learning for recurrent networks. Technical Report CS93-285, Dept. Computer Science and Engineering, University of California, San Diego.
- Weiss, M. (1997). Learning oscillations using adaptive control. *International Conference on artificial Neural Networks*, pages 331–336.

INTELLWHEELS

A Development Platform for Intelligent Wheelchairs for Disabled People

Rodrigo A. M. Braga^{1,2}, Marcelo Petry², Antonio Paulo Moreira² and Luis Paulo Reis^{1,2}

¹*Artificial Intelligence and Computer Science Lab-LIACC*

²*Faculty of Engineering of University of Porto, Rua Dr. Roberto Frias, s/n 4200-465, Porto, Portugal*
rodrigo.braga@fe.up.pt, marcelo.petry@gmail.com, amoreira@fe.up.pt, lpreis@fe.up.pt

Keywords: Intelligent Wheelchair, Intelligent Robotics, Human-Robots Interfaces.

Abstract: Many people with disabilities find it difficult or even impossible to use traditional powered wheelchairs independently by manually controlling the devices. Intelligent wheelchairs are a very good solution to assist severely handicapped people who are unable to operate classical electrical wheelchair by themselves in their daily activities. This paper describes a development platform for intelligent wheelchairs called IntellWheels. The intelligent system developed may be added to commercial powered wheelchairs with minimal modifications in a very straightforward manner. The paper describes the concept and design of the platform and also the intelligent wheelchair prototype developed to validate the approach. Preliminary results concerning automatic movement of the IntellWheels prototype are also presented.

1 INTRODUCTION

Wheelchairs are important locomotion devices for handicapped and senior people. With the increase in the number of senior citizens and the increment of people bearing physical deficiencies in the social activities, there is a growing demand for safer and more comfortable Intelligent Wheelchairs (IW) for practical applications. The main functions of IWs are (Jia, 2005) (Faria, 2007a) (Faria, 2007b):

- Interaction with the user, including hand based control (such as, joystick, keyboard, mouse, touch screen); voice based control; vision based control and other sensor based control (such as pressure sensors);
- Autonomous navigation (with safety, flexibility and robust obstacle avoidance);
- Communication with other devices (like automatic doors and other Wheelchairs).

This paper discusses the concept and the design of a development platform for intelligent wheelchairs. The project, called IntellWheels, is composed of a control software, simulator/supervisor and a real prototype of the intelligent wheelchair. In the study, shared control and high-level planning algorithms applied in an IW operating in a hospital environment were developed and tested. A shared control algorithm was tested,

allowing the IW to automatically avoid dangerous situations. Also, typical algorithms used in most intelligent robotics applications were applied in the control of the IW and simulated in a hospital scenery. Blended with the control, a motion planner was developed capable of generating the behavior/path commands according to an a-priori created map of the world. This motion planner is capable of instructing the low-level motion controller module to achieve the high-level commands desired by the user (Luo, 1999).

The rest of the paper is subdivided as following different sections: Section 2 presents some related work; section 3 explains the hardware desing of our development plataform; section 4 and 5 contain a complete description of the software design and control system; section 6 provides experimental tests and result discussion and section 7 presents the final conclusions and points out some future research topics.

2 RELATED WORK

This section presents a brief to the state of the art about Intelligent Wheelchairs.

In recent years, many intelligent wheelchair have been developed (Simpson, 2005). Only in the year of

2006 more than 30 publications in IEEE, about IW, may be found.

One of the first concept projects of an autonomous wheelchair for handicapped physicists was proposed by Madarasz (Madarasz, 1986). He presented a wheelchair equipped with a microcomputer, a digital camera, and ultrasound scanner. His objective was to develop a vehicle capable to operate without human intervention in populated environments with little or no collision with the objects or people contained in it.

Hoyer and Holper (Hoyer, 1993) presented an architecture of a modular control for a omni-directional wheelchair. NavChair is described in (Simpson, 1998), (Bell, 1994), (Levine, 1997) and has some interesting characteristic such as wall following, automatic obstacle avoidance, and doorways passing capabilities.

Miller and Slak (Miller, 1995) (Miller, 1998) projected the Tin Man I system, initially with three ways of operation: human guided with obstacle avoidance, move forward along a heading, move to a point (x, y). Afterwards, the project Tin Man I evolved, resulting in several new functions, extended in the Tin Man II, by including new capabilities such as: backup, backtracking, wall following, doorway passing, and docking.

Wellman (Wellman, 1994) proposed a hybrid wheelchair which is equipped with two legs additionally to the four wheels, which enable the wheelchair to climb over steps and move through rough terrain.

Some projects presented solutions for people with tetraplegia, by using the recognition of facial expressions as the main input to guide the wheelchair (Jia, 2006), (Pei, 2001), (Adachi, 1998). Another method of control is by using the user "thoughts". This technology typically uses sensors that measure the electromagnetic waves of the brain (Lakany, 2005) (Rebsamen, 2007).

ACCoMo (Hamagami, 2004) is a prototype of an IW that allows for safe movement in indoor environments for the handicapped physicists. ACCoMo is an agent based prototype with simple autonomous, cooperative and collaborative behaviors.

Although several prototypes have been developed and different approaches have been proposed for IWs, there is not, at the moment, a proposal for a platform enabling easy development of Intelligent Wheelchairs using common electric powered wheelchairs with minor modifications.

3 HARDWARE DESIGN

The hardware architecture of IntellWheels prototype is shown in Figure 1. The IntellWheels_chair1 is based on a commercial electrical wheelchair model Powertec, manufactured by Sunrise in England (Sunrise, 2007). The Powertec wheelchair has following features: Two differentially driven rear wheels; Two passive castor front; Two 12V batteries (45Ah); Traditional Joystick; Power Module.

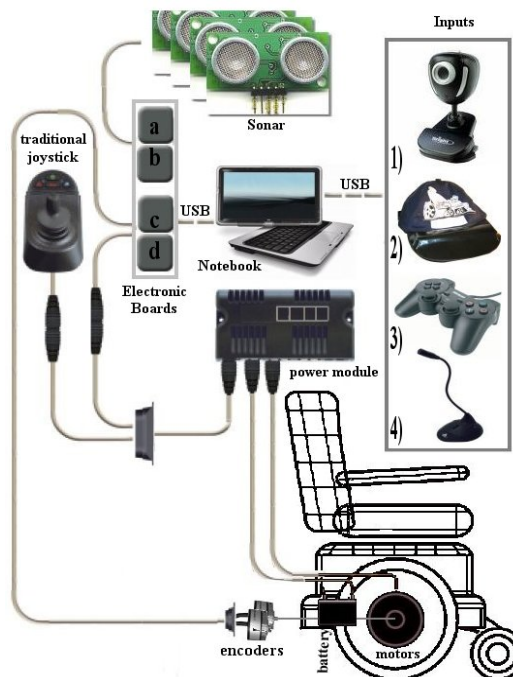


Figure 1: Hardware Architecture of IntellWheels.

The IntellWheels hardware parts are divided in three functional blocks: user inputs, IW sensor, hardware devices. This blocks are depicted in Fig. 2.

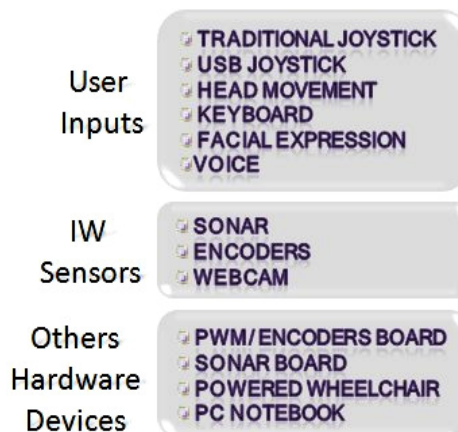


Figure 2: Hardware Functional Blocks.

3.1 User Inputs

To enable people with different kinds of disabilities to drive the intelligent wheelchair, this project has incorporated several sorts of user inputs. The idea is to give options to the patients, and let them choose what control is more comfortable and safer. Besides that, multiple inputs makes possible the IntellWheels integration with an intelligent input decision control, which is responsible to cancel inputs in the case its perception recognizes conflicts, noise and danger. For example, in a noisy environment, voice recognition would have low rate or even would be canceled. The inputs implemented go from traditional joysticks until head movement control, and are explained below:

- Traditional Joysticks. These inputs present in ordinary wheelchairs are a robust way to drive a wheelchair. However they may not be accessible to paraplegic or cerebral palsy people. They are present in the prototype due to its simplicity;
 - USB Joystick. The USB joysticks are a little bit more sophisticated than traditional joysticks. This game joystick has many configurable buttons that makes the navigation easier;
 - Head Movement. This input device is mounted in a cap making it possible to drive the wheelchair just with head movement;
 - Keyboard. This device enables that the wheelchair control can be made just pushing some keys in the keyboard;
 - Facial Expressions. By using a simple webcam, present in most of the notebooks, this software recognize simple facial expressions of the patient, using them as inputs to execute since basic commands (like: go forward, right and left) to high level command (like: go to nursery, go to bedroom);
 - Voice. Using commercial software of voice recognition we developed the necessary conditions and applications to command the wheelchair using the voice as an input.
- The use of a vast set of input options enables the prototype to be easily controlled by patients suffering from distinct disabilities.

3.2 Sensors

The purpose of this project is to develop an intelligent wheelchair. The distinction between an IW and a robotic wheelchair is not just semantic. It means that we want to keep the wheelchair appearance, reducing the visual impact that the sensors mounted on the device produce but, at the same time, increasing the wheelchair regular

functionalities. This statement limits the number and the kind of sensors we are able to use due to size, appearance and assembly constraints. To compose the intelligent wheelchair ten sonar sensors were mounted (giving the ability to avoid obstacles, follow walls and perceive unevenness in the ground), two encoders were assembled on the wheels (providing the tools to measure distance, speed, position) and a webcam were placed directed to the ground (capable to read ground marks and refine the odometry).

3.3 Hardware Devices

The hardware devices block is composed of (Fig. 3):

- 2 sonar board (Electronic board 'a' and 'b' illustrated in the Figure 1 and Figure 2), the function 'a' and 'b' boards are receiving information of the ten sonars and sending it to the PC;
- 2 PWM/Encoders board (Electronic board 'c' and 'd'), these boards have speed control function, as well as sending to the PC the displacement information to enable the odometry;
- Commercial electrical wheelchair and commercial notebook.

The core of IntellWheels prototype is a PC notebook (HP Pavilion tx1270EP,AMD Turion 64 X2 TI60), although other notebooks could be used without any loss of capabilities.

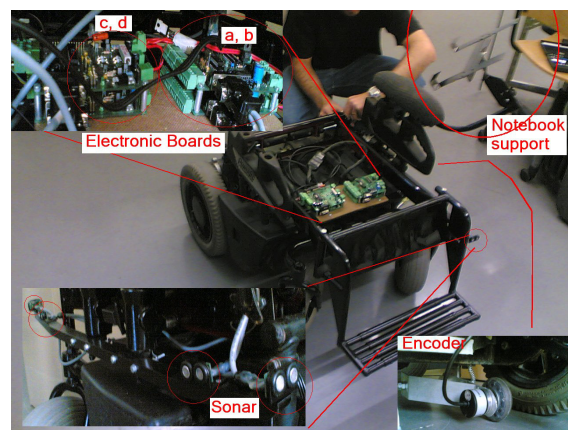


Figure 3: Devices installed in IW.

4 SOFTWARE DESIGN

The software in this project is composed by a modular system, where each group of tasks interact with the main application. Figure 4 shows the specific functions and the software responsible to

convert data from hardware to be used in the control level.

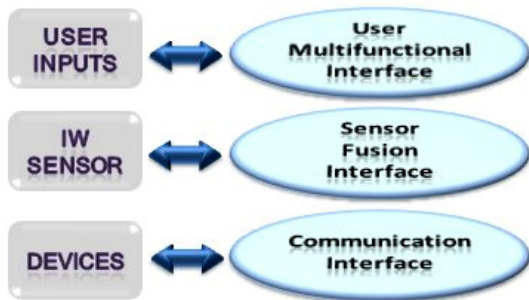


Figure 4: Intelligent Wheelchair Software.

The main application has the function to gather all information from the system, communicate with the control module and set the appropriate output. Moreover, it has other different functions depending on what is the control interface mode set:

- Real. If this option is selected, all data comes from the real world. The main application collects real information from sensors through the Sensor User Interface, calculate the output through the Control module and send these parameters to the PWM Boards;
- Simulated. In this mode, the system works just with virtual information and is used for two purposes: generate the same behavior of a real wheelchair and to test control routines. The main application collects virtual information directly from the simulator, calculates the output through the Control module and send back these parameters to the simulator;
- Augmented Reality. This mode creates an interaction between real and virtual objects, changing a real wheelchair's path to avoid collision with a virtual wall (or with a virtual wheelchair) for example. The objective of the augmented reality is to test the interaction between wheelchairs and environment, reducing its costs once major agents can be simulated. The main application collects real information from the sensors through the Sensor User Interface and mix it with virtual information collected from the simulator, calculate the output through the Control module and send the parameters to the simulator and the PWM Boards.

Due to its capabilities to interact with real, simulated and augmented reality worlds we call this whole system: IW Platform (Figure 5). In other words, the Platform is the fusion of the Simulator, Software modules, Real Wheelchair and Hardware Devices, to test, preview, understand and simulate the behaviour of Intelligent Wheelchairs.

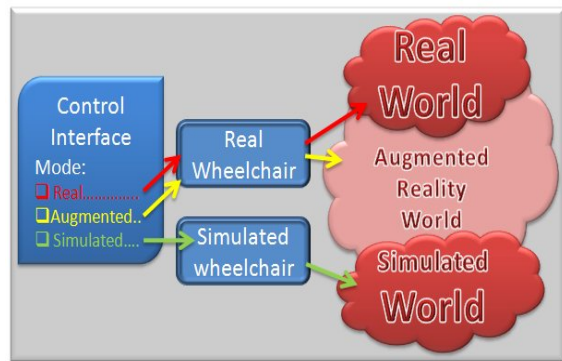


Figure 5: Intelligent Wheelchair Platform Architecture.

5 CONTROL

IntellWheels has a multi-level control architecture subdivided in three levels: a basic control level, a tactical level and strategy level, as shown in Fig. 6.

As focus is primarily testing the Platform, classic algorithms were chosen to validate the system, some other issues are left outside the scope of this paper, such as analyses of its limitations and its performance. High Level strategy plan is responsible to create a sequence of high level actions needed to achieve the global goal. Actually the algorithm implemented to fulfill this task is based on the STRIPS planning algorithm (Fikes, 1971).



Figure 6: Control architecture.

In the Generation of Action Plans, we order the system to generate a sequence of basic actions aiming to satisfy the objectives proposed previously.

To find a path from a given initial node to a given goal node the system has a simple A* Algorithm implemented (Shapiro, 2000). Once the path is calculated, it is subdivided into basic forms (lines, circles, points) that are afterwards followed by the wheelchair.

Each basic form represents a basic action to be executed on time in the Basic Action Control module. Following that the system executes the Generation of References, which is responsible to estimate the wheelchair linear and angular speeds, putting the wheelchair into motion.

The lowest level of control is designed through a Digital PID implemented in the PWM/Encoders Boards. References from Basic Action are transferred by serial communication to the boards and then contrasted with real time data to control the speed.

6 EXPERIMENTS AND RESULTS

This section presents the prototype implementation and a simple experiment and the results achieved. Figure 7 shows the mechanical structure and the hardware implementation. As we can see, the prototype is a commercial electric powered wheelchair with minimal modifications.



Figure 7: IntellWheels_Chair1 Prototype.

User interface software was designed to be as friendly as possible and it is shown in Figure 8. In

the main window it contains a webcam window, which is used to recognize landmarks, and the results of its localization. A schematic figure with the position of the sonar mounted in the wheelchair easily shows the distance to nearby objects. By the side, we have a panel with the information provided by the odometry. The application still displays a scrollbar indicating the speed of each wheel as well as buttons to choose the operation mode.



Figure 8: Interface Control of IntellWheels.

Some basic tests were performed to validate odometry, and consisted in moving the wheelchair around a rectangle path, starting and stopping in the same point. In Figure 9 the results of an automatic test (red line) and a manual test (blue line) are presented. In the manual test the user had all control of the wheelchair.

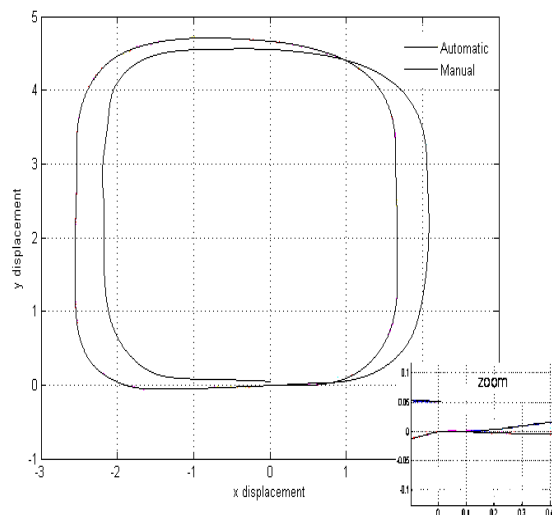


Figure 9: Path Followed by Wheelchair in the Tests.

In the automatic mode the final point of the path of the wheelchair presented a little displacement regarding its start point (physical mark point on the ground). As this error is not displayed in the odometry graph, a Manual test was executed to evaluate the results. In this test, the user drove the wheelchair following the same path, but at this time stopping in the same physical mark point on the ground. Results of this test presented the error displayed in the odometry graph.

The error presented is admissible since it is just 5 cm in a total amount of 1500 cm of displacement, and can be explained by the integration of the odometry systematic error.

Figure 10 shows the results of distinct tests of displacement in straight line. The objective was evaluate odometry dispersion error for different distances. In this test was valued displacement of 5, 10 and 15 meters, with approximately 20 samples for each path.

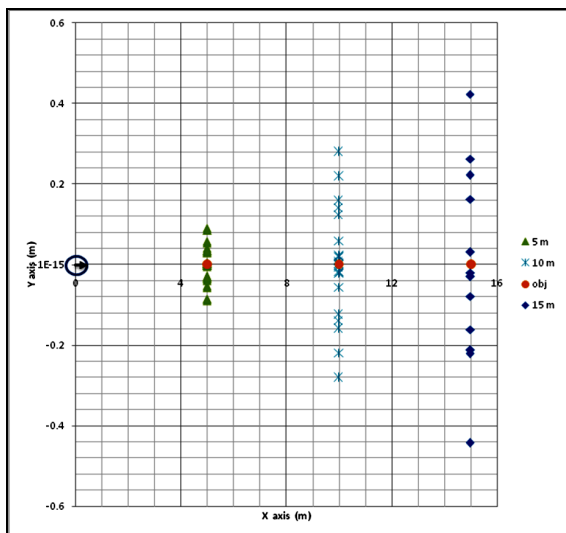


Figure 10: Odometry dispersion error for 5m, 10m and 15m of displacement in straight line.

7 DISCUSSION AND CONCLUSIONS

This paper presented the design and implementation of a development platform for Intelligent Wheelchairs called IntellWheels.

This platform facilitates the development and test of new methodologies and techniques concerning Intelligent Wheelchairs. We believe that this new techniques can bring to the wheelchairs real capacities of intelligent action planning, autonomous

navigation, and mechanisms to allow the execution in a semi-autonomous way of the user desires, expressed in a high-level language of command.

Future research will aim the test and validation for the other sensors mounted in the wheelchair. Moreover, like in many other systems, cooperative and collaborative behaviours are desired to be present in the IW and need to be incorporated in the Platform. Another important improvement to be pursued includes a comparative study of the classic implemented algorithms and the new proposal to solve these issues.

The platform will allow that real and virtual IWs interact with each other. These interactions make possible high complexity tests with a substantial number of devices and wheelchairs, representing a reduction in the project costs, once there is not the necessity to build a large number of real IW.

ACKNOWLEDGEMENTS

The first author would like to thank for CAPES for doctoral course financing.

REFERENCES

Adachi, Y., Kuno, Y., Shimada, N., Shirai, Y., 1998. Intelligent wheelchair using visual information on human faces. *International Conference in Intelligent Robots and Systems*. Oct. Vol. 1. pp. 354-359.

Bell, D.A., Borenstein, J., Levine, S.P., Koren, Y., and Jaros, L., 1994. An assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance. *IEEE Conf. on Robotics and Automation*.

Faria, P. M., Braga, R. A. M., Valgôde, E., Reis L. P., 2007a. Platform to Drive an Intelligent Wheelchair using Facial Expressions. *Proc. 9th International Conference on Enterprise Information Systems - Human-Computer Interaction (ICEIS 2007)*. pp. 164-169. Funchal Madeira, Portugal. June 12-16. ISBN: 978-972-8865-92-4

Faria, P. M., Braga, R. A. M., Valgôde, E., Reis, L. P., 2007b. Interface Framework to Drive an Intelligent Wheelchair Using Facial Expressions. *IEEE International Symposium on Industrial Electronics*. 4-7 de June de 2007, pp. 1791-1796, ISBN: 1-4244-0755-9.

Fikes, R., Nilsson, N. J., 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *IJCAI 1971*. pp.608-620.

Hamagami, T., Hirata, H., 2004. Development of intelligent wheelchair acquiring autonomous, cooperative, and collaborative behavior. *IEEE*

- International Conference on Systems, Man and Cybernetics. Oct. Vol. 4. pp. 3525 - 3530.
- Hoyer, H. and Hölper, R., 1993. Open control architecture for an intelligent omnidirectional wheelchair. Proc. 1st TIDE Congress, Brussels. pp. 93-97.
- Jia, P. and Hu, H., 2005. "Head Gesture based Control of an Intelligent Wheelchair", CACSUK - 11th Annual Conference of the Chinese Automation and Computing Society in the UK, Sheffield, UK, Sept. 10.
- Jia, P., Hu, H., Lu, T. and Yuan, K., 2006. Head Gesture Recognition for Hands-free Control of an Intelligent Wheelchair. *Journal of Industrial Robot*.
- Lakany, H. 2005. Steering a wheelchair by thought. The IEEE International Workshop on Intelligent Environments. June. pp. 199 - 202.
- Levine, S.P., Bell, D.A., Jaros, L.A., Simpson, R.C., Koren, Y., Borenstein, J., 1997. The NavChair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering* pp. 443-451.
- Luo, R. C., Chen, T. M. and Lin, M. H., 1999. Automatic Guided Intelligent Wheelchair System Using Hierarchical Grey-Fuzzy Motion Decision-Making Algorithms, Proc. of IEEVRSJ'99 - International Conference on Intelligent Robots and Systems.
- Madarasz, R.L., Heiny, L.C., Cromp, R.F. and Mazur, N.M., 1986. The design of an autonomous vehicle for the disabled, *IEEE J. Robotics and Automat.* September, pp. 117 - 126.
- Miller, D. and Slack, M., 1995. Design and testing of a low-cost robotic wheelchair prototype. *Autonomous Robots*. Vol. 2, pp. 77-88.
- Miller, D., 1998. Assistive Robotics: An Overview. *Assistive Technology and AI*. pp. 126-136.
- Pei Chi Ng; DeSilva, L.C., 2001. Head gestures recognition. *Proceedings International Conference on Image Processing*. Oct. Vol. 3. pp. 266-269. ISBN: 0-7803-6725-1.
- Rebsamen, B., Burdet, E., Guan, C., Zhang, H., Teo, C. L., Zeng, Q., Laugier, C., Ang Jr., M. H., 2007. Controlling a Wheelchair Indoors Using Thought. *IEEE Intelligent Systems and Their Applications*. Vol. 22, pp. 18 - 24.
- Shapiro S.C., 2000. *Encyclopedia of Artificial Intelligence*. Wiley-Interscience. May.
- Simpson, R. C., 2005. Smart wheelchairs: A literature review. *Journal of Rehabilitation Research & Development*. August. Vol. 42. pp. 423-436.
- Simpson, R. et al. 1998. NavChair: An Assistive Wheelchair Navigation System with Automatic Adaptation. [ed.] Mittal et al. *Assistive Technology and AI*. pp. 235-255.
- Sunrise, 2007. Sunrise Medical manufacturers www.sunrisemedical.co.uk
- Wellman, P., Krovi, V. and Kumar, V., 1994. An adaptive mobility system for the disabled. *Proc. IEEE Int. Conf. on Robotics and Automation*.

HOW TO ASSESS RELIABILITY OF INDUSTRIAL WIRELESS SOLUTIONS

Lutz Rauchhaupt and Marko Krätzig

*Institut für Automation und Kommunikation e.V. Magdeburg, Steinfeldstr. 3, Barleben, Germany
lutz.rauchhaupt@ifak.eu, marko.kraetzig@ifak.eu*

Keywords: Wireless Communication, Industrial Communication, Performance Investigation.

Abstract: Wireless communication is an emerging technology for industrial automation applications. Many solutions are available which more or less consider industrial related requirements. One of the main concerns of industrial automation system users is the reliability of wireless communication. The subject of this paper is a method to assess reliability of wireless communication from the point of view of industrial automation applications. Characteristic parameters are introduced which can be used in analytical studies, in network simulations or measurements to assess reliability with respect to intended industrial control processes. In particular the different use cases for the characteristic parameters are stated as well as the stochastic nature of these parameters. Finally the influences are mentioned which have to be taken into account while assessing wireless industrial communication systems.

1 INTRODUCTION

Wireless communication technologies are widely spread in daily life. The price of wireless products is thereby the main design aspect with respect to the consumer market. Reliability is one of the minor design goals. Therefore, almost everyone has had negative experiences with such technologies and has developed concerns regarding the usage of wireless in industrial communication.

Indeed a number of measures are applied to make industrial fit products from cheap solutions of the consumer market (Dzung, 2005), (Weczerek, 2005), (Siemens, 2007). However, how can users be convinced that wireless solutions meet the requested reliability of industrial automation applications?

This paper starts with a definition of reliability with respect to the application area - industrial automation. Thereafter a model is introduced from which relevant characteristic parameters are derived which are used to assess reliability. Some examples follow which show how to assess wireless solutions with the described approach.

2 MODEL

2.1 Requirements

First we would like to clarify what reliability means in context of wireless industrial communication. A user of an industrial communication system expects a certain process value, e.g. position or temperature, at a certain interface within a defined time frame without any errors under defined conditions. This is an informal definition. In order to be able to assess the degree of fulfilment of this requirement by means of simulation or measurement, a formal model is required.

First of all this model has to take into account the application field - the industrial automation. The parameters to be investigated have to be in line with the design criteria of industrial automation systems. Parameters such as Data Throughput or Bit Error Rate are normally not useful to design a particular automation application which e.g. shall manufacture a product in a certain time frame or with a certain cycle.

Furthermore, the model has to consider that there is no general wireless standard available for industrial automation which fits to all communication tasks. Several different technologies are used for industrial automation. A unified interface between communication and application is

not available. Therefore the model has to be independent of a certain wireless technology and even more it has to be open for future developments in wireless communication such as Ultra-Wideband.

Last but not least the model should represent the conditions of reality as accurate and complete as possible and necessary. The following section introduces an approach which fulfils the mentioned requirements.

2.2 Approach

The abstraction of a distributed automation application using wireless communication is shown in Figure 1. Wireless communication modules are seen as an internal or external part of automation devices. The automation devices have to fulfil certain functions in a distributed automation system and for that they have to communicate in our case using a wireless communication media. From the point of view of the automation system, the communication characteristics at the interface provided by the wireless solution are important. These communication characteristics have to fit to the time and error categories used in the industrial automation area as introduced later in this document.

It must be clearly defined as to what the communication interface is, upon which the characteristics are related to. This interface consists of a hardware part such as Ethernet or Dual Ported RAM and a software part such as a communication protocol or a driver. Besides a clear statement concerning the communication interface and the communication characteristics, the conditions have to be described under which the characteristic values are valid. The conditions can be described by a number of influencing values which have different origins. It is obvious that the communication system itself affects the characteristics concerning e.g. topology or data rate. It is also evident that the communication media has influence because of other users of the spectrum or because of the effects of multi path fading. Furthermore, the characteristics depend on the options chosen in the devices, which means on its configuration. It is sometimes forgotten that also the application affects the characteristic values in the sense of the size of a packet or the cycle of requests on the communication system.

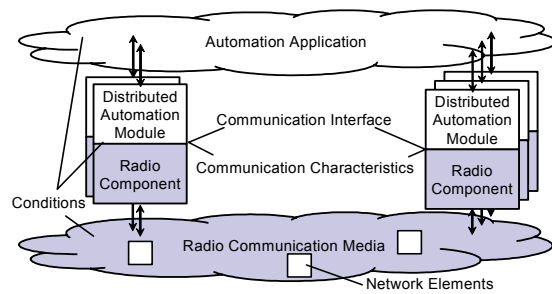


Figure 1: Model approach for the assessment of wireless industrial communication systems.

2.3 Characteristic and Influencing Parameters

The analysis of literature concerning the usage of characteristic parameters to describe and assess communication behaviour has shown that there are remarkable differences. Moreover, the definitions come mostly from the application field of Ethernet, Internet or telecommunication which does not fit to the application field of industrial automation (e.g. in (DIN EN 61491, 1999), (DIN EN 61209, 200)). That is why it was necessary to find appropriate definitions. The following characteristic parameters are proposed to assess wireless communication systems with respect to industrial automation applications:

- Transmission delay
- Response time
- Update time
- Data throughput
- Packet loss rate
- Residual error rate
- Activation time after energy saving mode
- Energy requirements

It has to be mentioned that it is not required nor recommended to use all parameters at the same time to characterise a communication solution for a certain application. The following sections show exemplary which parameters fit to which kind of use cases. The definitions of the listed characteristic parameters can be found in (VDI/VDE 2185, 2007).

It is obvious that the values of the characteristics are influenced by several parameters. That is why it is important to know these parameters and their values. Some of the parameters can be set with certain values but it is also possible that the parameters can not be influenced. In this case it is important to determine the value of the parameter to be able to assess the determined characteristic value.

The first set of influencing values is related to the application. This includes

- A background communication load, which exists in addition to the communication under investigation
- A user data length (packet size)
- A distance between the radio components
- An application period
- A relative moving speed between the radio components
- A relative moving direction between the radio components

The second set of influencing values is related to the radio technology and the radio devices. It includes

- A topology
- A frequency band
- A security functionality
- A safety functionality
- A type, direction and gain of antenna
- A transmission power
- A data rate via the physical media
- A media access control method
- A retry limit in case of errors
- A data rate at the communication interface
- A communication cycle

The third set of influencing values is related to the environment in which the communication will take place. It includes

- An application area
- Electromagnetic disturber
- Other frequency users
- Environmental conditions

Taking into account the listed influencing parameters while determining target-oriented relevant characteristic parameters, these can be used to assess the time and error behaviour of a wireless communication solution with respect to automation applications.

2.4 Use on Reliability Definition

Now we can define the term reliability more specific and we can describe how to assess reliability. In line with the definition of chapter 2.1, reliability can be seen as the degree in which you can expect that a wireless communication solution meets the limits of relevant characteristic parameters. With this definition it is obvious that the assessment of reliability needs stochastic measures. The characteristic parameters are random variables. Their behaviour follow probability density functions. The reliability is the probability that a value of a characteristic parameter is less or equal to

the limit defined by the automation application.

3 ASSESSMENT OF RELIABILITY

3.1 Event Driven Data Transmission

Event driven data transmission is relevant for process variables which indicate that a certain state is assumed. For example when a work piece reaches a specified position that it can be machined or when a fluid reaches a defined level in a tank. In these cases it is of interest as to how long it takes to transfer the information from sensor to the control unit e.g. programmable control logic (PLC). The appropriate characteristic parameter to assess the behaviour of a communication system is the transmission delay.

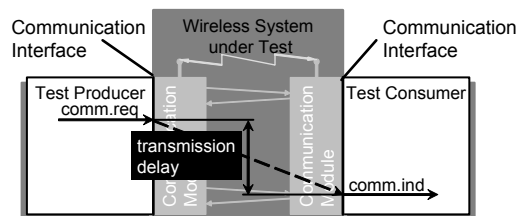


Figure 2: Definition of the transmission delay.

The definition of the transmission delay is based on a producer consumer model (see Figure 2). It is the time duration from the beginning of the handing over of the first user data byte of a packet at the communication interface in the test producer, up to the handing over of the last user data byte of the same packet at the communication interface at the test consumer. It may be necessary to transmit several telegrams between the communication modules e.g. for acknowledgment. Furthermore, network elements such as base stations may be involved in the communication producing additional delays. All these delays are covered by the transmission delay.

As already mentioned, the characteristic parameters and therefore the transmission delay are random variables. Next it is shown which parts of the transmission are randomly distributed and which are constant. Furthermore, it is shown which parts are specific for wireless transmission and what makes these special considerations necessary.

To get a deeper understanding of enlarged transmission delays, the most important segments of a transmission delay value are listed in Table 1.

Table 1: Time segments which influence the transmission delay value.

Time Segments		Remark
Latency of application interface	T_{ai}	The data transfer between application module and communication module may influence the transmission delay value remarkably.
Latency of implementation	T_i	The implementation of the communication module influences the transmission delay value remarkably.
User data length	L_{ud}	The user data length is related to the data which is generated or consumed by the automation application.
Data rate	Bd_{ud}	This rate is the radio transmission rate of the user data. Sometimes a symbol rate is given. In this case a symbol may consist of more than one bit. The header of a packet containing the user data may be transmitted with another data rate.
Technology constant	T_{tc}	The technology constant contains all technology relevant protocol overheads which are the same for each transmission such as fixed idle times or the time to transmit headers or tails.
Technology variable	T_{tv}	The technology variable contains all technology relevant protocol overheads which may vary for different transmissions such as the time to get a clear channel or the back-off time. Depending on the technology, acknowledgments are required to complete a transmission.
Number of retries	N_r	If a transmission is disturbed, the packet is usually retransmitted. This may be possible at different layers.
Transmission deadline	DL	In some cases the transmission is terminated when a deadline is exceeded.
Time allocation of additional connections	T_{ac}	If there is more than one connection established, the time allocated to the other connections within the same system has to be taken into account.
Global time slot	T_{GTS}	In systems with TDMA the maximum transmission delay can be calculated considering the global time slot.

The random nature of the transmission delay is being caused by the latency of the application interface and implementation, by the technology variable, the number of retries and the time allocation for additional connections. In contrast to wired communication, the wireless transmission is affected much more by environmental influences. Therefore, the random behaviour of the technology variable together with the number of retries and the

time allocation for additional connections may influence the transmission delay remarkably

Taking into account the time segments listed in Table 1, the dependency of the transmission delay can be described in different ways. The first way is the given formula (1) and is illustrated in Figure 3.

$$T_{td} = f(T_{ai}(p), T_i(p), T_{ai}(c), T_i(c), L_{ud}, Bd_{ud}, T_{tc}, T_{tv}, N_r, T_{ac}) \quad (1)$$

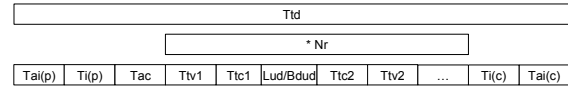


Figure 3: Time segments of transmission delay depending on re-transmissions, data rate and data length.

The second way to describe the transmission is given in formula (2) and Figure 4. The maximum transmission delay is fundamentally influenced by the maximum allowed deadline which covers the random behaviour of the media related time segments.

$$T_{tdmax} = f(T_{ai}(p), T_i(p), T_{ai}(c), T_i(c), DL) \quad (2)$$

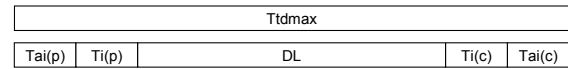


Figure 4: Time segments of transmission delay depending on a transmission deadline.

The third way to describe the transmission delay is given in formula (3) and shown in Figure 5. The maximum transmission delay is fundamental depending on the number of retries and the global time slot.

$$T_{tdmax} = f(T_{ai}(p), T_i(p), T_{ai}(c), T_i(c), N_r, T_{GTS}) \quad (3)$$

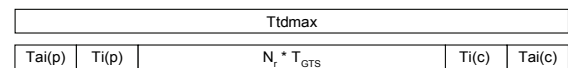


Figure 5: Time segments of transmission delay depending on a global time slot.

As an example typical results of transmission delay measurements are depicted in Figure 6. The lower part of the figure shows the number of packets, relative to the sample size, with certain transmission delay values. The above described random nature of the transmission delay can be observed. The reasons for the different values are

mainly transmission retries because of disturbances and delays due to an occupied media. The curve follows a Beta probability density function.

The probability distribution function of the measurement is depicted in the upper part of Figure 6. It shows how many packets are transmitted by a certain point of time.

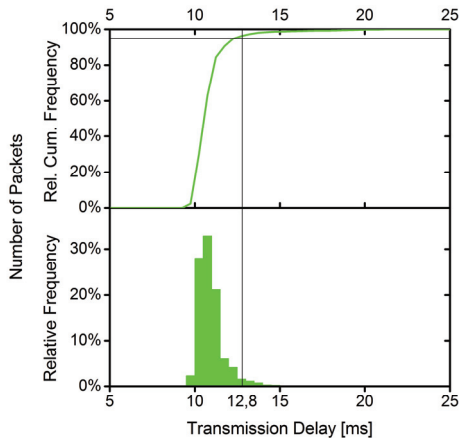


Figure 6: Probability Density Function and Probability Distribution Function (Cumulative Density Function) of packets' transmission delay.

To assess the time behaviour of a wireless solution, the well known statistical parameters for the centre (e.g. mean value) and for the variation (e.g. standard deviation) can be used. Our experience is that the 95th percentile value P95 is the best indicator for relevant changes in the time behaviour e.g. because of disturbances. It is a trade off between a feasible sample size (e.g. one million packets) and an adequate significance.

The maximum value is not qualified for assessment since it is a single value of a series of measurements and it is not sure that the real maximum value is captured. An infinite measurement of the transmission delay would be necessary or an inference to a larger population using methods of inferential statistics. However, the maximum value is considered so far as it influences the value of the 95th percentile P95.

The assessment of the reliability of an event driven data transmission means a comparison of a limit for a statistical parameter given by the application with the statistical parameters of a measurement.

3.2 Cyclic Data Transmission

Most of the control processes in industrial automation are cyclic. A process image is taken

cyclically via input devices or process interfaces. It is processed by a controller and the result is output via output devices or interfaces. One example is the control of an overhead monorail system. The position is acquired cyclically and as a result the control information is transferred to the drive.

Also for these cases it is of interest as to how long it takes to transfer data e.g. from the position sensor to the controller. However, using the transmission delay to assess the time behaviour could be misleading. The problem is that in these cases different cyclic processes are involved which are not synchronised. Taking as an example a rotary encoder sensor in which the communication buffer is cyclically updated with the position information which is cyclically transferred to a controller. The effect of the asynchronism is shown in Figure 7.

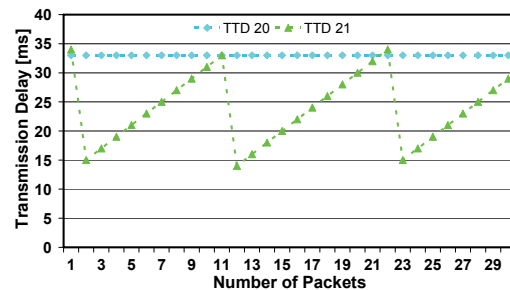


Figure 7: Effect of asynchronous cyclic processes on transmission delay.

Depending on the initial situation and the period of the cyclic processes, the transmission delay may have very different statistical parameters. In the case TDD20, the communication cycle is an integral multiple of the update time and therefore the transmission delay value is constant. In the case TDD21, first the transmission has been started just before the buffer update (maximum value) and next the transmission starts just after the buffer update (minimum value). Thus, the mean value may decrease even when the communication cycle increases as shown in Table 2. The variation on the other hand becomes much higher.

Since this behaviour is random and in reality more than two cyclic processes are involved, it is possible that the influence on the wireless transmission is overlaid by the effect shown in Figure 7 and can therefore possibly not be assessed. This behaviour must at least be considered.

In most cases the update time is the appropriate characteristic parameter to assess the time behaviour of communication systems with cyclic data transfer. The update time is ascertained according to the

producer-consumer-model (see Figure 8). This means the period of time is from the delivery of a packets' last user data byte, from the communication interface of a consumer to the application, until the delivery of the last user data byte of the following packet of the same producer. Therefore the update time is at least as long as the transmission delay between producer and consumer, prolonged by the time of the application update within the producer.

Table 2: Transmission delay values for different communication cycles.

	TTD20	TTD21
Buffer update	40 ms	40 ms
Communication cycle	20 ms	21 ms
$N_r * (T_{tc} + T_{tv} + L_{ud} / Bd_{ud})$	13 ms	13 ms
Transmission Delay (Mean Value)	33 ms	23,8 ms

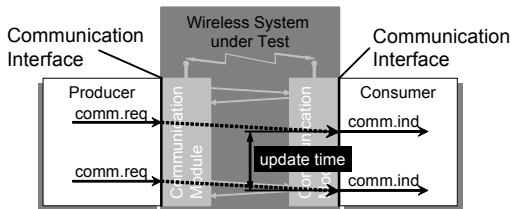


Figure 8: Definition of cycle time at producer-consumer model.

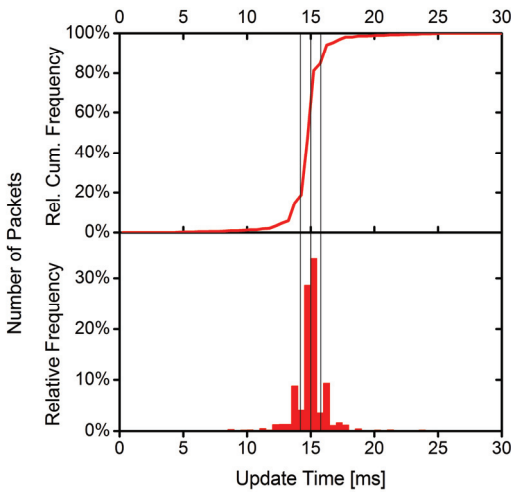


Figure 9: Probability Density Function and Probability Distribution Function (Cumulative Density Function) of packets' update time.

As an example typical graphs are depicted in Figure 9 for the probability function and the distribution function of packets with certain update time values. The update time is a Gaussian

distributed random value. The mean value indicates in the first place the usability of the wireless communication system for a certain cyclic control process. Another important parameter is the span in the automation area known as jitter. However, it has to be mentioned that the measured minimum and maximum values which are used to calculate the span are most likely not the absolute extreme values. Again an infinite measurement of the update time would be necessary or an inference to a larger population using methods of inferential statistics. Therefore the span can only be assumed with a certain probability.

Furthermore, measurements have shown that the standard deviation value is well suited in order to indicate influences on the wireless communication system. Therefore, this parameter together with the span can be used to assess the reliability of cyclic data transmissions. The mean value follows the application cycle if the system is correctly configured. This means it is equal to the application cycle which is the period e.g. a sensor value is updated in the communication buffer.

3.3 Assessment of Error Behaviour

Up to now it was assumed that none of the transferred data got lost. That means the configured number of re-transmissions were sufficient to transfer the user data successfully. This chapter discusses how the reliability is assessed using the Packet Loss Rate (PLR).

The packet loss rate (PLR) is ascertained according to the producer-consumer-model. It reveals how many of the packets, transferred from the application to the communication interface within the producer, are transmitted from the communication interface to the application within the consumer. The packet loss rate is determined as follows:

$$PLR = \frac{N_{tx} - N_{rx}}{N_{tx}} \quad (4)$$

Where N_{tx} means number of transmitted packets and N_{rx} means number of received packets.

In principle industrial wireless communication solutions are designed to cope with the special environmental conditions. They are considered robust against interferer and industrial propagation conditions. Therefore, in principle no packets disappear. However, tacking into account the maximum limits of the transmission delay the situation changes. A remarkable packet loss rate can be noticed in the case where a packet is considered

to be lost when a certain value of transmission delay is exceeded (or a deadline is missed). The calculation of the packet loss rate for a certain use case can be done as shown in formula (5). The number of packets is acquired, which have transmission delay values less or equal to the limit defined by the use case. The difference to the sample size is assumed to be lost packets. Thus, the packet loss rate is calculated with respect to use case specific limit of transmission delay. Especially interferers cause higher PLR values. Considering the packet losses and the transmission delay, the reliability can be assessed by comparing the PLR_{UC} with the required packet loss rate of the use case. Since also the PLR_{UC} is a random value, measures should be foreseen by the application for the case that data is not received within the expected time frame.

$$PLR_{UC} = \frac{N_{tx} - N_{rx}(t | t_{TD} \leq T_{TDmaxUC})}{N_{tx}} \quad (5)$$

In the following chapter some examples are given on how the reliability can be assessed.

4 EXAMPLES

4.1 Overview

The examples presented in this chapter shall illustrate how the characteristic parameters are used for certain purposes. The measurement scenarios, the results and their assessment are not the topic of this paper.

The results presented in this chapter come from measurements with an IEEE 802.11g based industrial communication solution. Different influences have been investigated. A test system generated packets with a length of 64 octets and transferred them with an application cycle of 15 ms to the interface of the test producer. At the test consumer, the packets are transferred to the test system. The test system measured the values of the characteristic parameters. In none of the presented cases could a packet loss concerning (4) be investigated.

The systems under investigation are not specified in detail in the current document since the project in which the measurements are made has not yet been completed. Moreover, in this paper the method of assessment is in the focus and not the absolute results of the tests.

4.2 Assessment of Event Driven Data Transmission

The following figures show the number of packets which are received after a certain time in line with the above given definition of transmission delay. Figure 10 shows the result of a measurement with an industrial wireless solution made in an absorber hall that means without any environmental influences.

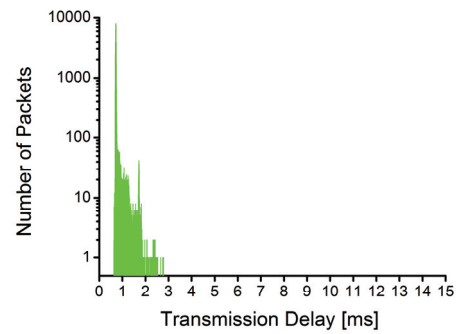


Figure 10: Histogram of transmission delay in absorber hall.

In Figure 11 the same system was placed in a factory hall. Influences due to the environment can be ascertained.

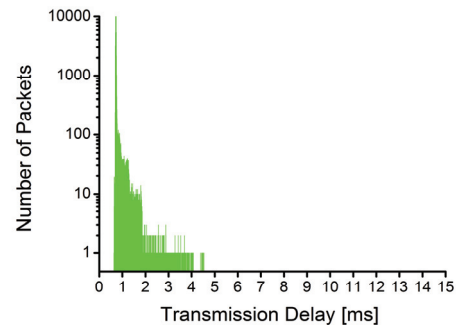


Figure 11: Histogram of transmission delay in factory hall.

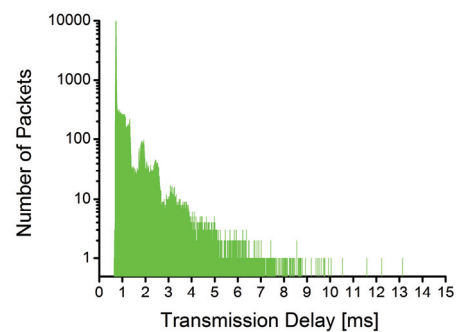


Figure 12: Histogram of transmission delay in factory hall with interferer.

In Figure 12 an interferer was finally activated and a considerable influence can be ascertained.

However, comparing the values in Table 3 with a transmission delay limit e.g. of a high speed I/O system which is 10 ms, it is obvious that the reliability of the communication is comparable for all investigated conditions. In particular there was no packet loss concerning the definition of formula (5). That means even when the wireless communication is noticeably influenced, this does not mean that the requirements of a certain application can not be fulfilled.

Table 3: Transmission Delay Values.

Transmission Delay [ms]	Min.	Max.	P95
Absorber Hall	0,6	2,8	0,8
Factory Hall	0,6	4,5	0,8
Interferer	0,7	13,1	2,0

4.3 Assessment of Cyclic Data Transmission

Figure 13 to Figure 15 show the update time for the same scenarios described in the previous section.

As shown in Table 4 the mean values of the update times are equal to the application cycle of 15 ms for all scenarios. By contrast the span differs. Taking a limit for the span from $\pm 1,5$ ms in the first case the requirement is fulfilled. In the second case 614 packets and in the third case 12.563 packets are out of range. This results in a reliability of 99,9% and 97,4% concerning the definition of this paper.

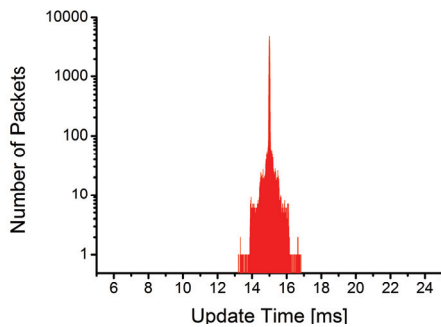


Figure 13: Histogram of update time in absorber hall.

Table 4: Update Time Values.

Update Time [ms]	Min.	Max.	Mean
Absorber Hall	13,2	16,8	15,0
Factory Hall	11,2	18,8	15,0
Interferer	7,0	23,0	15,0

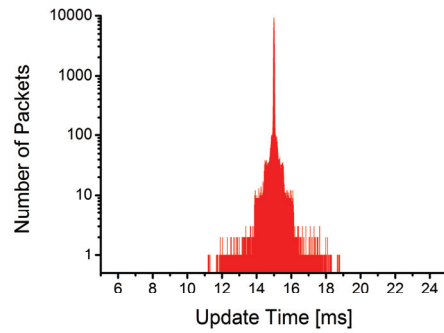


Figure 14: Histogram of update time in factory hall.

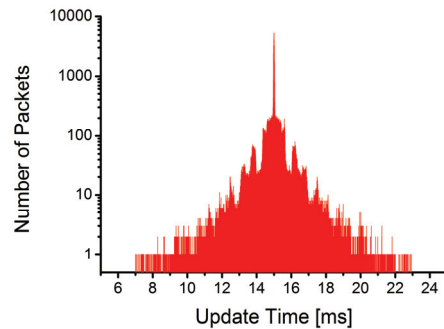


Figure 15: Histogram of update time in factory hall with interferer.

5 CONCLUSIONS

In the paper we presented a proposal on how to assess the reliability of industrial wireless solutions. A fundamental requirement for such a method is the focus on industrial automation applications. That is why characteristic values such as transmission delay, update time and packet loss are used in the way defined in this paper. It was pointed out that these parameters are random variables which mean the statistical parameters have to be considered.

The described method has been used to assess the coexistence of different industrial wireless communication solutions. Furthermore, these solutions are currently being used to assess the possibility of using wireless communication in automation applications with safety requirements. A test system is available which supports the measurements of the described characteristic parameters (Rauchhaupt, 2006).

The approach can be used for analytical studies, simulations and tests. The method is required for wireless communication since the dimension of influences can be remarkably greater than in wired systems.

The work described in this paper is accompanied by important manufacturers of automation and radio solutions, and users of such systems which work together in the German Society of Measurement and Automation. As a result the characteristic parameters presented in the paper are introduced in the VDI/VDE-Guideline 2185 "Radio based communication in industrial automation".

Siemens, 2007. "Industrial mobile communication."
Siemens AG, Order No. 6ZB5530-1AM02-0BA2

ACKNOWLEDGEMENTS

This paper presents results from the project "Funk-Transfer-Tester für industrielle Funklösungen" which deals with the development of a method and a tool used to investigate the time and error behaviour of radio solutions for industrial automation applications. The project is funded by the German Ministry of Commerce and Labour within the research program "Förderung von Forschung und Entwicklung bei Wachstumsträgern in benachteiligten Regionen" - INNOVATIVE-WACHSTUMS-TRÄGER (INNO-WATT).

REFERENCES

- DIN EN 61491, 1999. "Elektrische Ausrüstung von Industriemaschinen - Serielle Datenverbindung für Echtzeit-Kommunikation zwischen Steuerungen und Antrieben (IEC 61491:1995, modifiziert); Deutsche Fassung EN 61491:1998"
- DIN EN 61209, 2000. "Navigations- und Funkkommunikationsgeräte und -systeme für die Seeschifffahrt – Integrierte Brückensysteme (IBS) - Betriebs- und Leistungsanforderungen, Prüfverfahren und geforderte Prüfergebnisse (IEC 61209:1999); Deutsche Fassung EN 61209:1999"
- VDI/VDE 2185, 2007. "Funkgestützte Kommunikation in der Automatisierungstechnik (Radio based communication in industrial automation)"
- Rauchhaupt, L., Gnad, A., Hintze, E., 2006. "Wireless Automation Under Test", embedded world 2006 conference, Nurnberg, Germany, proceedings part I, ISBN 3 7723 0143 6, pp. 529
- Dzung, D., Apneseth, C., Endresen, J., Frey, J.-E., 2005. "Design and Implementation of a Real-Time Wireless Sensor/Actuator Communication System", 10.1109/ETFA.2005.1612710, Volume: 2, pp. 10 – 19, ISBN: 0-7803-9401-1
- Weczerek, J., 2005. "Erfahrungen beim industriellen Einsatz von Bluetooth als Wireless-IO-Lösung." VDI-KfIT-Fachtagung Funkgestützte Kommunikation in der industriellen Automatisierungstechnik, Düsseldorf, Germany, ISBN 3-18.091889-6, pp. 83-93

SHORT PAPERS

WALKING PLANNING AND CONTROL FOR A BIPED ROBOT UPSTAIRS

Chenbo Yin¹, Donghua Zheng¹ and Le Xiao²

¹*School of Mechanical and Power Engineering, Nanjing University of Technology, Nanjing, China
yinchenbo@njut.edu.cn, zhengdonghua1981@163.com*

²*School of Computer Science and Engineering, Changshu Institute of Technology, Suzhou, China
xiaole66@126.com*

Keywords: Humanoid robot, stability control, gait planning, stability margin, ZMP, FZMP.

Abstract: The focus of this paper is the problem of walking stability control in humanoid robot going upstairs. Walking stability is a very important problem in the field of robotics. Lots of researches have been done to get stable walking on plane. But it is very limited on going upstairs. We first plan the gait of ankle and hip when going upstairs as well as the calculation of stable region and stability margin. Then the emergency-coping strategy of enlarging the support polygon is provided. At last, a control system which is proved to be effective by simulation is presented. If the ZMP is in the support polygon, this control system makes fine setting to gait to get higher stability. If the ZMP is out of the support polygon, the control system adjusts the location of ZMP through the emergency coping strategy.

1 INTRODUCTION

Biped humanoid robots have better mobility than wheeled robots, especially for moving on rough terrain, steep stairs and obstacle environments (Huang, 1999). Research on humanoid robots has become one of the most exciting topics in the robotics field and there are many ongoing projects (Kaneko, 2002; Konno, 2002; Pfeiffer, 2002). Many researches are made on the walking stability of biped robots (Kajita, 2003; Stojic, 2000).

In order to realize stably walking, many different models are proposed. Such as the zero-moment point (ZMP), Vukobratovic (Vukobratovic, 1990; Vukobratovic, 2004) first proposed; the criterion of "Tumble Stability Criterion" for integrated locomotion and manipulation systems, proposed by Yoneda etc (Yoneda, 1996); the foot-rotation indicator (FRI), introduced by Goswami (Gowami, 1999) and so on. This paper uses the Fictitious Zero-Moment Point (FZMP) (Yin, 2005) criterion to calculate stability.

There are different walking patterns in different environment. Tatsuo Narikiyo etc (Narikiyo, 2006) researched walking control of robot when walking in space. Shuuji Kajita etc (Kajita, 2004) researched a biped which can jump. But as humanoid robot which

will be used widely in our daily life, it will be more welcomed if the robot can walk up and down stairs. So in this paper, we discuss the gait planning and present an effective control strategy of robot going upstairs.

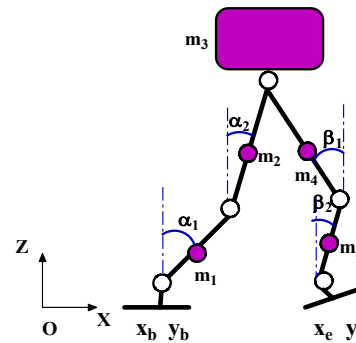


Figure 1: The link model of the humanoid robot.

2 STABILITY CALCULATION

In order to evaluate dynamic stability, we use the ZMP principle. The ZMP is the point where the influence of all forces acting on the mechanism can be replaced by one single force. If the ZMP is inside

the support polygon, the biped robot can be stable. If the ZMP is on the boundary of the support polygon, the robot will fall down or have a trend of falling down. If the computed ZMP is outside the support polygon, then the robot will fall down and in this case, the computed ZMP is called fictitious ZMP. The link model of the humanoid robot is shown in Figure 1.

The projection of position vector of computed ZMP can be computed by the following equations:

$$x_{zmp} = \frac{\sum_{i=1}^5 m_i (\ddot{z}_i + g) x_i - \sum_{i=1}^5 m_i \ddot{x}_i z_i + \sum_{i=1}^5 M_{iy}}{\sum_{i=1}^5 m_i (\ddot{z}_i + g)} \quad (1)$$

$$y_{zmp} = \frac{\sum_{i=1}^5 m_i (\ddot{z}_i + g) y_i - \sum_{i=1}^5 m_i \ddot{y}_i z_i + \sum_{i=1}^5 M_{ix}}{\sum_{i=1}^5 m_i (\ddot{z}_i + g)} \quad (2)$$

where m_i is mass of every links, (x_i, y_i, z_i) is the coordinate of the mass center of the links, $(M_{ix}, M_{iy})^T$ is the moment vector.

If the ZMP is inside the support polygon and the minimum distance between the ZMP and the boundaries of support polygon is large, then the biped will be in high stable, and this distance is called the stability margin. We can know the situation of walking stability from the stability margin.

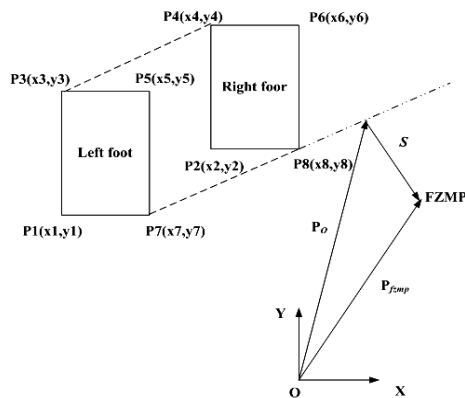


Figure 2: The relationship between FZMP and support polygon.

As shown in Figure2, if the ZMP is outside the support polygon, i.e. FZMP, the norm of vector s represents the shortest distance between FZMP and the edges of the support polygon. This

edge is called rotation edge. The direction of vector s is the rotation direction of the robot.

3 GAIT PLANNING OF ANKLE WHEN GOING UPSTAIRS

In order to simplify research process we first discuss how to get ankle trajectory and hip trajectory. Then the knee trajectory could be got by kinematics. Here we take the left foot for example and the right foot is similar only with a delay of half cycle. The link model we used is shown in Figure 3.

3.1 Gait Planning of Ankle

According to the walking procedure of human, we suppose that the walking cycle is T_c , $t = kT_c$ is the k th cycle begins with the moment when the left foot is just apart from the ground and ends with the left foot gets into contact with the ground; $kT_c < t \leq kT_c + T_d$ is double support phase, during which the sole is rotated about toes, and the center of gravity moving forwards; the swing foot reaches the highest point when $t = kT_c + T_n$.

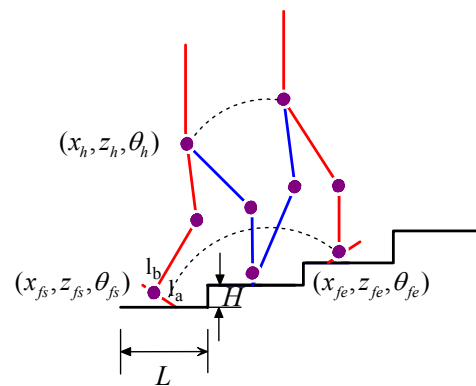


Figure 3: The model of the humanoid robot going upstairs.

We get the key point $x_f(t), z_f(t)$ of ankle in plane XOZ as follows:

$$x_f = \begin{cases} x_{fs} + 2Lk & t = kT_c \\ x_{fs} + 2Lk + l_a(1 - \cos \theta_f) + h_f |\sin \theta_{fs}| & t = kT_c + T_d \\ x_{fs} + 2Lk + L & t = kT_c + T_n \\ x_{fs} + 2L(k+1) - l_b(1 - \cos \theta_f) - h_f |\sin \theta_{fe}| & t = (k+1)T_c \\ x_{fs} + 2L(k+1) & t = (k+1)T_c + T_d \\ x_{fs} + 2L(k+2) & t = (k+2)T_c \end{cases} \quad (3)$$

$$z_f = \begin{cases} z_{fs} + 2Hk + h_f & t = kT_c \\ z_{fs} + 2Hk + h_f \cos \theta_{fs} + l_a |\sin \theta_{fs}| & t = kT_c + T_d \\ z_{fs} + 2Hk + h_f + H & t = kT_c + T_n \\ z_{fs} + 2H(k+1) + h_f \cos \theta_{fe} + l_b |\sin \theta_{fe}| & t = (k+1)T_c \\ z_{fs} + 2H(k+1) + h_f & t = (k+1)T_c + T_d \\ z_{fs} + 2H(k+2) + h_f & t = (k+2)T_c \end{cases} \quad (4)$$

where l_a is the distance between tiptoe and the centre of gravity of sole; l_b is the distance between heel and the centre of gravity of sole; h_f is the height of heel and T_n is the time when the robot just walks through a step.

The key point of the angle between sole and ground can be denoted as follows:

$$\theta_f = \begin{cases} 0 & t = kT_c \\ \theta_{fs} & t = kT_c + T_d \\ 0 & t = kT_c + T_n \\ \theta_{fe} & t = (k+1)T_c \\ 0 & t = (k+1)T_c + T_d \end{cases} \quad (5)$$

Since the whole sole of the right foot is in contact with the ground at $t = kT_c$ and $t = (k+1)T_c + T_d$, the following derivative constraints must be satisfied.

$$\begin{cases} \dot{x}_f(kT_c) = 0 \\ \dot{x}_f((k+1)T_c + T_d) = 0 \end{cases} \quad (6)$$

$$\begin{cases} \dot{z}_f(kT_c) = 0 \\ \dot{z}_f((k+1)T_c + T_d) = 0 \end{cases} \quad (7)$$

$$\begin{cases} \dot{\theta}_f(kT_c) = 0 \\ \dot{\theta}_f((k+1)T_c + T_d) = 0 \end{cases} \quad (8)$$

3.2 Gait Planning of Hip

We assume that the robot is decelerated in double support phase and accelerated in single support phase and the acceleration in direction of x -axis and z -axis are a_{xh} and a_{zh} respectively. The distance between the hip and the ankle of supporting leg is x_s at the beginning of the double support phase and x_e at the end of the double support phase.

The changes in the direction of z -axis are z_s and z_e at the beginning and end of the double support phase respectively. Then the trajectory of hip can be expressed like this:

It must satisfy the following constraints:

- The derivative constraints $\begin{cases} \dot{x}_h(kT_c) = \dot{x}_h(k+1)T_c \\ \dot{x}_h(kT_c) = \dot{x}_h(k+1)T_c \end{cases}$

and $\begin{cases} \dot{z}_h(kT_c) = \dot{z}_h(k+1)T_c \\ \dot{z}_h(kT_c) = \dot{z}_h(k+1)T_c \end{cases}$ must be satisfied.

- $z_h(t) \leq h_{\max}$, h_{\max} is the maximum height of hip; $h_{\max} = l_1 + l_2 + h_f$, l_1, l_2 are the length of thigh and shin respectively, h_f is the height of ankle.

- $z_h(t) \geq h_{\min}$, h_{\min} is the minimum height of hip and it's value can be set according to the process of human walking.

$$\{[x_h(t) - x_a(t)]^2 + [z_h(t) - z_a(t)]^2\}^{1/2} \leq l_1 + l_2$$

$$x_h(t) = \begin{cases} x_h + 2Lk + x_s & t = kT_c \\ x_h + 2Lk + x_s + a_{xh}t & t = kT_c + T_d \\ x_h + 2Lk + L & t = kT_c + T_n \\ x_h + 2L(k+1) - x_e - a_{xh}t & t = (k+1)T_c \\ x_h + 2L(k+1) - x_e & t = (k+1)T_c + T_d \\ x_h + 2L(k+1) + x_s & t = (k+2)T_c \end{cases} \quad (9)$$

$$z_h(t) = \begin{cases} z_h + 2Hk + z_s & t = kT_c \\ z_h + 2Hk + z_s + a_{zh}t & t = kT_c + T_d \\ z_h + 2Hk + H & t = kT_c + T_n \\ z_h + 2H(k+1) - z_e - a_{zh}t & t = (k+1)T_c \\ z_h + 2H(k+1) - z_e & t = (k+1)T_c + T_d \\ z_h + 2H(k+1) + z_s & t = (k+2)T_c \end{cases} \quad (10)$$

4 STABILITY CALCULATION

The maximum region enclosed by two soles' projection on the ground is called the stable region, as shown in Figure 4.

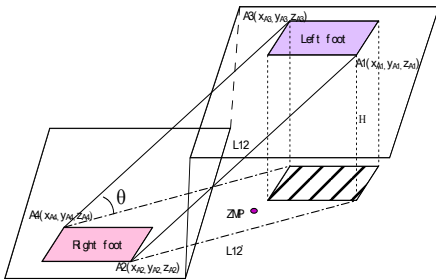


Figure 4: Stable region when going upstairs.

The equations of projection of line A_1A_2 and A_3A_4 can be got:

$$l_{12}' : \begin{cases} x = x_{A1} + (x_{A2} - x_{A1})t \\ y = y_{A1} + (y_{A2} - y_{A1})t \\ z = 0 \end{cases} \quad (11)$$

$$l_{34}' : \begin{cases} x = x_{A3} + (x_{A4} - x_{A3})t \\ y = y_{A3} + (y_{A4} - y_{A3})t \\ z = 0 \end{cases} \quad (12)$$

than the stable region can be expressed as follows:

$$\begin{cases} x_{\max} = \max(x_{Q1}, x_{Q2}) + l_a \\ x_{\min} = \min(x_{Q1}, x_{Q2}) - l_b \\ y_{\max} = \max(y_{Q1}, y_{Q2}) + l_m \\ y_{\min} = \min(y_{Q1}, y_{Q2}) - l_n \\ l_{12}' \\ l_{34}' \end{cases} \quad (13)$$

We can get the distances between ZMP and every boundary of stable region easily and the stability margin can be expressed as:

$$\gamma = \min(d_{x_{\max}}, d_{x_{\min}}, d_{y_{\max}}, d_{y_{\min}}, dl_{12}', dl_{34}') \quad (14)$$

5 MAINTAIN STABILITY BY ENLARGING SUPPORT POLYGON

As we said above, if the computed ZMP is outside the support polygon, the robot cannot be in dynamic stable and has the trend of falling down. In this case we can enlarge the support polygon to maintain stability.

As shown in Figure 5, the changed angle α_f^* of moving direction of the foot is determined by the following equation:

$$\alpha_f^* = \cos^{-1} \frac{e \cdot s}{\|e\| \cdot \|s\|} \quad (15)$$

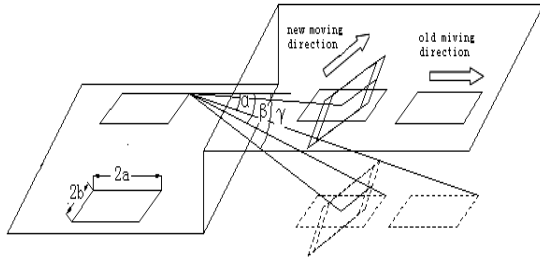


Figure 5: The determination of foot landing position.

where e is the normal vector of planned moving direction and s is the changed vector. If the gradient of step is θ and the distance between the feet is d , then the foot moving distance l_f^* relative to planned landing position is determined by the following formula:

$$l_f^* = \tan(\alpha^* - \beta) \left[\left(l^2 \cos^2 \theta + (d+3b)^2 \sin^2 \theta \right) \cos(\gamma - \alpha^*) - b \right] - \left[l^2 \cos^2 \theta + (d+3b)^2 \sin^2 \theta \right] \sin(\gamma - \alpha^*) - a \quad (16)$$

where β is the original angle of foot and γ is the changed angle of foot centre. a, b express half of the foot's long and width respectively.

6 CONTROL STRATEGY

If the computed ZMP is inside the support polygon then the robot can be in stable. But, the error between the computed ZMP and the designed ZMP is unavoidable. It means that the stability condition is not the best and we can make the robot more stable. Our control system (the left part) shown in Figure 7 can optimize the position of actual ZMP to enlarge the stable margin.

As shown in Figure 6, the inertial force F_R and ground reaction force R are not in the same line. In this case F_R, R and the error of ZMP form a moment which makes the robot roll. So it is necessary to reduce the error to diminish the moment. The rolling moment can be depicted like this

$$TM = (DZMP - AZMP) \times F_R \quad (17)$$

where $DZMP$ means *desired ZMP*; $AZMP$ means *actual ZMP*.

We can obtain the error of ZMP (ΔZMP) according to the desired ZMP and the actual ZMP. The gait adjustment parameter $\Delta\theta$ can be got using inverse kinematics.

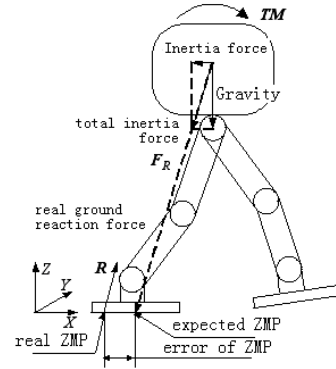


Figure 6: Error of ZMP.

$$\begin{aligned} \Delta\theta &= [\Delta\alpha_1, \Delta\alpha_2, \Delta\alpha_3, \Delta\beta_1, \Delta\beta_2]^T \\ &= K\Delta ZMP = [K_{\alpha_1}, K_{\alpha_2}, K_{\alpha_3}, K_{\beta_1}, K_{\beta_2}]^T \Delta ZMP \end{aligned} \quad (18)$$

where K is the adjustment coefficient matrix by experience.

If there is an external disturbance, the computed ZMP may be out of the support polygon and the robot may tip over, at this moment the control system (the right part) will take action.

Two important parameters can be got according to the definition of FZMP, i.e. the distance between FZMP and the rotation edge and the rotation direction. We can also get the change of the angle of link $\Delta\theta$ by the sensor fitted at the link. The emergency-coping strategies such as enlarging the support polygon, moving the upper body and contacting with surrounding by hands can be used to make the FZMP located in the support polygon and maintain the stability of the robot. Whereas, in some cases one method along may be unrealistic, two or three methods can be combined.

7 SIMULATION

We have constructed a simulator of a humanoid robot by using dynamic analysis software package ADMAS and the control system is built in Matlab. This allows us to analyze the joint torque, the change of ZMP, etc. The simulator built in ADMAS is shown in Figure 8.

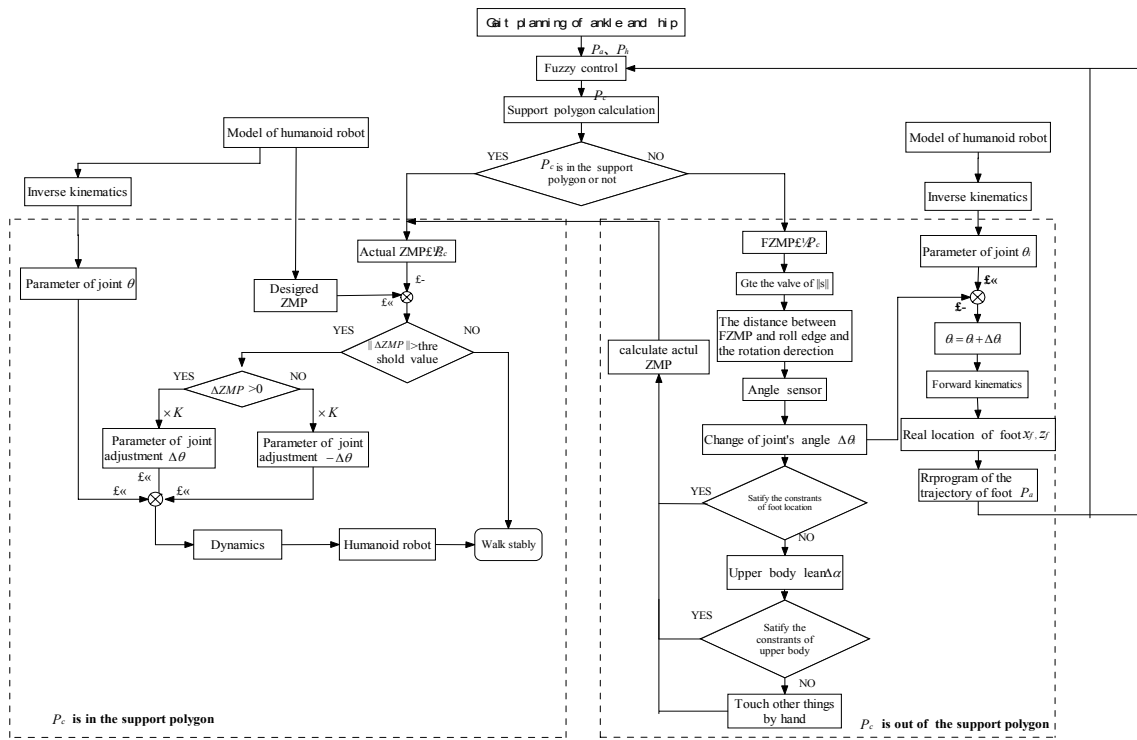


Figure 7: The control strategy.

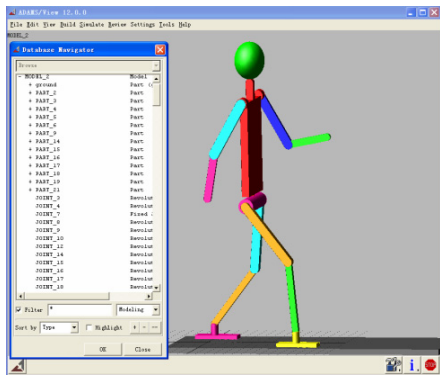


Figure 8: The ADMS model of humanoid robot.

The structure of control system based on Matlab/simulink is shown as follows:

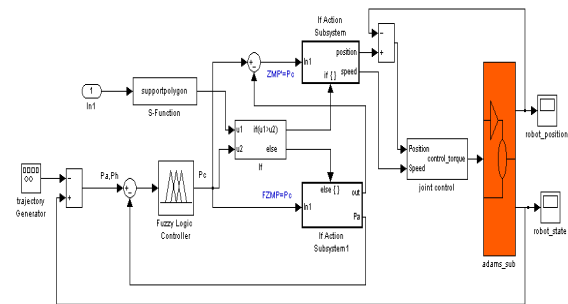


Figure 9: The structure of control system.

We simulate the procedure of walking up and down stairs with a height of 0.2m and a width of 0.6m. The pictures of series of walking upstairs are shown in Figure 10 at the time of 0.0s, 0.2s, 0.4s, 0.6s, 0.8s and 1.0s.

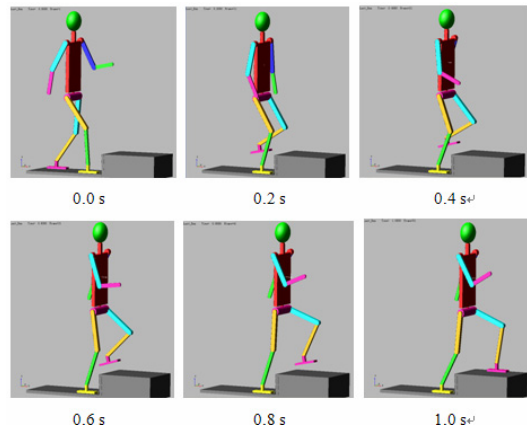


Figure 10: Series of walking upstairs.

The velocity and acceleration of ankle and hip are given in Figure 11 and 12 respectively. The x-axis represents time and the y-axis represents the velocity of ankle and hip. The red real line is velocity and the blue dashed is acceleration.

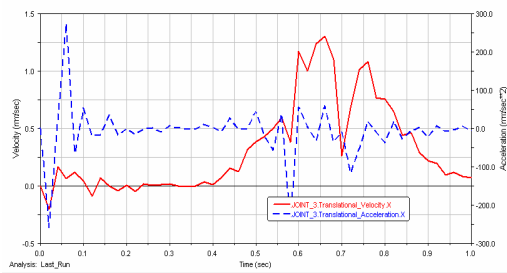


Figure 11: Velocity and acceleration of ankle.

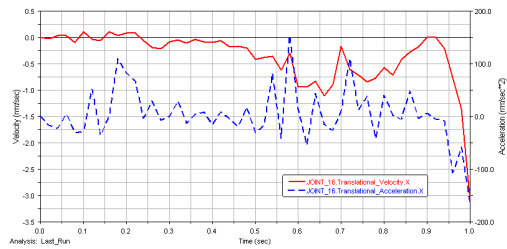


Figure 12: Velocity and acceleration of hip.

Figure 11 shows that the maximum velocity happens when the swing foot is descending and it is consistent with the real walking process of human. It also shows that the velocity and acceleration are close to zero at the time of 1.0s and the slope is getting smaller when the time is getting nearer to 1.0s. We all know that the landing acceleration of ankle is very important for the stability of robot. If the acceleration is too big, the impact force between robot and ground may be very large and the robot may become unstable. So in our simulation, the impact force is very small and the robot can walk stably. It shows that the strategy described above works.

8 CONCLUSIONS

Biped robots have better mobility than wheeled robots but tip over easily, so the walking stability is even more important. When going upstairs the stability problem is especially crucial and the research is far from enough. In order to make the robot go upstairs stably, it is necessary to have an efficient control strategy to scout and make adjustment on time. In this paper, we propose a method to plan a walking pattern and the way of calculating the stable region and stability margin is also presented. The stability maintenance method of enlarging support polygon is given out. The optimization control strategy which is proved to be useful by numerical simulation is proposed.

REFERENCES

- Huang, Q., Kajita, S., Koyachi, N., Kaneko, K., Yokoi, K., Arai, H., Komoriya, K., Tanie, K., 1999. A High Stability, Smooth Walking Pattern for a Biped Robot. Proceedings of IEEE International Conference Robotics and Automation.
- Kaneko, K., Kajita, S., Kanehiro, F., Yokoi, K., Fujiwara, K., Hirukawa, H., Kawasaki, T., Hirata, M., Isozumi, T., 2002. Design of Advanced Leg Module for Humanoid Robotics Project of METI. Proceedings of IEEE International Conference on Robotics & Automation.
- Konno, A., 2002. Design And Development of the Biped Prototype Robian. Proceedings of IEEE International Conference on Robotics & Automation.
- Pfeiffer, F., Loeffler, K., Gienger, M., 2002. The Concept of Jogging JOHNNIE. Proceedings of IEEE international conference on robotics & Automation.
- Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H., 2003. Biped Walking Pattern Generation By Using Preview Control of Zero-Moment Point. Proceedings of IEEE International Conference on Robotics & Automation.
- Stojic, R., Chevallereau, C., 2000. On the Stability of Biped with Point Foot-Ground Contact. Proceedings of IEEE International Conference on Robotics & Automation.
- Vukobratovic, M., 1990. Biped Locomotion: Dynamics, Stability, Control And Application. Springer Verlag, Berlin.
- Vukobratovic, M., Borovac, B., 2004. Zero-Moment Point-Thirty Five Years of Its Life. International Journal of Humanoid Robotics Vol. 1.
- Yoneda, K., Hirose, S., 1996. Tumble Stability Criterion of Integrated Locomotion and Manipulation. Proceedings of IEEE International Conference on Intelligent Robot and Systems.
- Gowami, A., 1999. Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point. The International Journal of Robotics Research, vol.18.
- Yin, C.B., Albert, A., 2005. Stability Maintenance of a Humanoid Robot under Disturbance with Fictitious Zero-Moment Point. IEEE/RSJ International Conference on Intelligent Robots and Systems.
- Narikiyo, T., Ohmiya, M., 2006. Control of a planar space robot: Theory and experiments. Control Engineering Practice, Vol. 14, Issue 8.
- Kajita, S., Nagasaki, T., Kaneko, K., Tanie, K., 2004. A Hop towards Running Humanoid Biped. Proceedings of IEEE International Conference Robotics and Automation

TEMPORAL MATCH OF MULTIPLE SOURCE DATA IN AN ETHERNET BASED INDUSTRIAL ENVIRONMENT

Daniela Hossu and Andrei Hossu

University Politehnica of Bucharest, Faculty of Control and Computers
313 Spl. Independentei, sector 6, RO-77206, Bucharest, Romania
dana@aii.pub.ro, hossu@aii.pub.ro

Keywords: Industrial Robotic Application, Ethernet-based communication, Data transfer consistency, Artificial Vision System, Control Management System, Routing Control System.

Abstract: The actual stream in automation control systems is to distribute the control tasks among different modular and easy to integrate processing cells. It is part of this trend the increase of the use of Ethernet technology for machine-machine data communication inside this distributed based architecture. The paper presents a robotic handling application of industrial parts transferred by a transport conveyor. Data representing a set of parameters of the parts to be handle from the conveyor is provided by a Routing Control System (RCS). The Control Management System (CMS), which controls a number of robotic cells is receiving this data from RCS and merge it with the information provided by an Inspection System (Artificial Vision System). The communication between these two Control Systems (RCS and CMS) is Ethernet-based. Ethernet technology is good, reliable and fast for large amount of data, but because of its non-deterministic character, it has a lack of tools for data synchronization. The paper includes an analysis of the experimental results of the measurements of the non-deterministic factor of the existing network. The "worst case scenario" of the largest communication delay caused by Ethernet traffic and the minimum time between two consecutive data commands, reveals that application requirements could not be achieved without recovering data transfer time-consistency. The paper is presenting a mechanism developed at protocol level, in order to guarantee the consistency in time, at CMS level (data consumer), of the merging process of the data provided by the two application partners, RCS and Vision System as the data producers.

1 THE DESCRIPTION OF THE ROBOTIC APPLICATION

The system described in the paper is dedicated for the robot-based automation of the unloading and packing stages in the flat glass industry. In Figure 1 it is presented the architecture of the proposed automation system. This architecture is often utilized in industrial applications (in palletizing of moving objects systems).

1.1 The Structural System Architecture

▪ *Active Elements:*

Control Management System (CMS), Routing Control System (RCS), Vision System and Robotic Cells.

▪ *Passive Elements:*

Conveyor, glass plates.

▪ *Infrastructure:*

Communicational Link between Vision System and CMS;

Communicational Link between CMS and Robots Controllers;

Communicational Link between CMS and RCS.

▪ *General assumptions:*

The plates are connected to the conveyor (the same speed and direction).

1.2 The Functional System Architecture

The Routing Control System has to provide for CMS the Routing Data - a description of the possible destinations (one or more of the robotic cells) of each plate in the moment the plate is passing the Decision Point of the Vision System. The role of the Vision System is to inspect the cutting accuracy and the shape parameters of every plate. The Vision

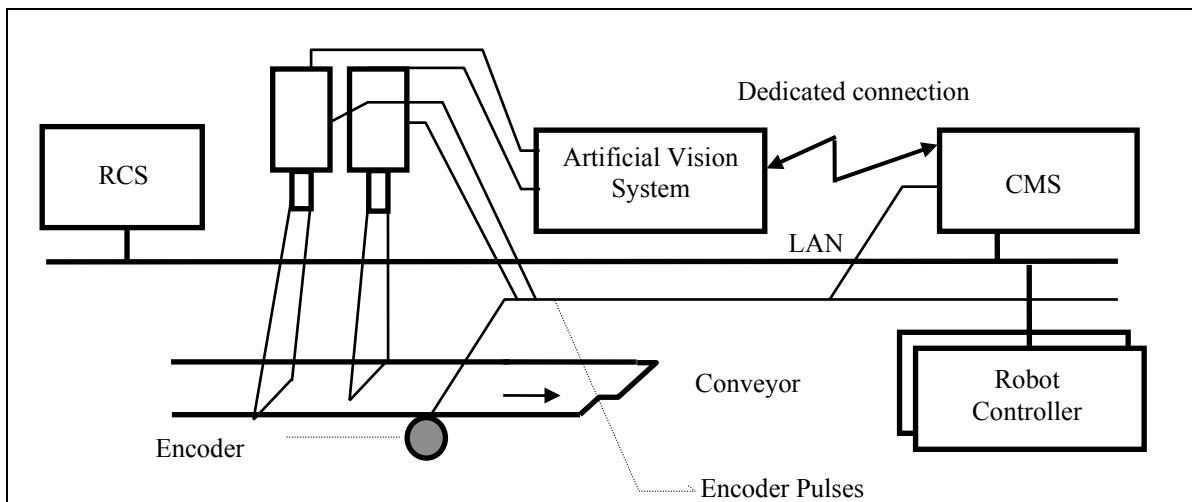


Figure 1: The robot-based automation system for handling glass plates from a moving conveyor.

System is analyzing the information provided by a Line Scan Acquisition System (a dual line scan camera system) in conjunction with the information provided by an encoder connected to the transport conveyor. The Vision Data, containing the data resulted from the inspection process, together with the data describing the location of the plate, are transmitted to CMS in the moment the Vision System processing time ended.

The moment (time-based) is called Vision Decision Point. Both sets of data (Routing Data and Vision Data) are merged by CMS. CMS will take the decision to send the pick the plate command to a certain robotic cell only if Vision Data describe the plate having cutting accuracy and shape parameters inside the accepted tolerances for a certain packing destination AND the plate is routed to that certain destination.

2 THE DESCRIPTION OF THE INFORMATIONAL SYSTEM ARCHITECTURE

The communication between Vision System and CMS is a dedicated connection. Some of the reasons for choosing this type of communication is the geographical neighborhood of these two systems and the fact the structure of the data transferred between Vision System and CMS could be very tight specified (no need for using a general type of protocol). Another reason is the concept that Vision System is an intelligent sensor of the Robotic Application, which means the Vision System will be

not “visible” on the higher automation level, but only on the Robotic Automation Level (the Vision System is “visible” only for CMS). This communication channel has a serial support. This type of connection is providing a deterministic character of the Vision System – CMS communication. For the communication between CMS and RCS it was adopted an Ethernet-based topology. The main reason of choosing this type of topology is the fact CMS is an automation entity visible on the high automation level (CMS includes all the automated stacking capabilities of the whole production line). Industrial Protocol on Ethernet is a very good and reliable support for modern configurations on industrial automations, but because it’s non-deterministic character, it is poor in data time-synchronization (Marshall, et al., 2004). The time-synchronization between the Routing Data (data coming from RCS) and the Vision Data (data coming from Vision) in the merging process in CMS it is a key factor of achieving the requirements of the automation application.

3 EXPERIMENTAL RESULTS

In Figure 2 are presented the experimental results of recording Ethernet Delays over around 10 minutes on the analyzed network. The Ethernet Delays are estimated as the differences between the CPU time of receiving Routing Data (sent over a non-deterministic communication channel) and the CPU time of receiving Vision Data (sent over a deterministic communication channel). A lost in

synchronization will occur only if the variation of the Ethernet Delay value from the average value of the Ethernet Delay is greater than the minimum time between two consecutive sets of Routing Data. Analyzing the manufacturing process we can identify what is the minimum time between two consecutive sets of Routing Data.

This is the minimum time between two glass plates coming on the conveyor (this is called “snapping period”). The actual glass manufacturing process has the minimum snapping period of 1.44 seconds. Analyzing the experimental results we could see the necessity of implementing a method for recovering data transfer time-consistency.

4 RECOVERING DATA TRANSFER TIME-CONSISTENCY

A few solutions were analyzed in order to solve this problem (Marshall, et al., 2004):

- Use a more powerful Ethernet board (instead of using 10MB/s type of board, to use a 100BT Ethernet module)
- Replace the communication software support (RSLinx) with another one with a better response time (a software module dedicated only for a specific protocol would provide a better response time related to a general software package like RSLinx, which is coming with a large CPU overhead).

The above two solutions could improve the Ethernet behavior, but the non-deterministic character of this type of communication is not eliminated.

- Install another dedicated Ethernet module in the RCS and an additional dedicated Ethernet module in the CMS PC. These modules would be connected to a separate isolated Ethernet switch. In this case, most of the delays experienced on the current Ethernet link would be eliminated since the only traffic on the link would be between the routing system and the CMS cell PC.
- Use an ASCII serial (RS-232, RS-485, etc.) connection rather than using Ethernet. This would make the communication time between the Routing and CMS systems deterministic.
- Use a dedicated digital signal from RCS to the CMS in addition of the Ethernet connection in order to be used to re-synchronize the Routing Data in CMS. This would be implemented by

energizing a digital output that would indicate to the CMS cell in the moment of sending current Routing Data. When the input was seen by the CMS system, the CMS system would capture its own internal time and it will use this time value in the moment the Routing Data is received over Ethernet.

These last three solutions could solve the time-synchronization problem, but any of these solutions wouldn't be accepted because of a dramatic aggression on the network topology previously agreed on the design time of the application (Mackay, et al., 2003), (Stenson, et al., 2002).

The solution proposed in the paper is based on inserting a “*timestamp parameter*” in any set of Routing Data transmitted from the Routing System to the Control Management System.

This *timestamp parameter* will be the Routing System CPU time in millisecond representation. This timestamp parameter value will be a wraparound counter representing the least significant two bytes (one Word) of the CPU time (in milliseconds).

This *timestamp parameter* will be used by CMS to estimate with a “good enough” approximation, the offset between the CPU time values of the producer of the message (RCS) and the consumer of the message (CMS). This estimated offset would be used for time synchronization of the current message. It means CMS will add the estimated CPU time values offset to the current timestamp parameter value contained in the current received message.

In order to provide a “good enough” approximation of the offset between the CPU time values of the producer (RCS) and the consumer (CMS) the algorithm has to estimate the minimum value of the statistical population containing all the offsets estimated for a large number of transmitted/received messages.

This minimum is a “moving minimum” (it will be estimated from a statistical population collected on a certain time window) because we expect a slippage between the clocks of the two CPU (this slippage is accumulative and will become significant in time).

For the support of building this statistical population of offset values is used an existing message from RCS to CMS, called “Request Status Message”.

This message is sent by RCS every half a second in order to check the communication with CMS and also to obtain from the CMS the status of the

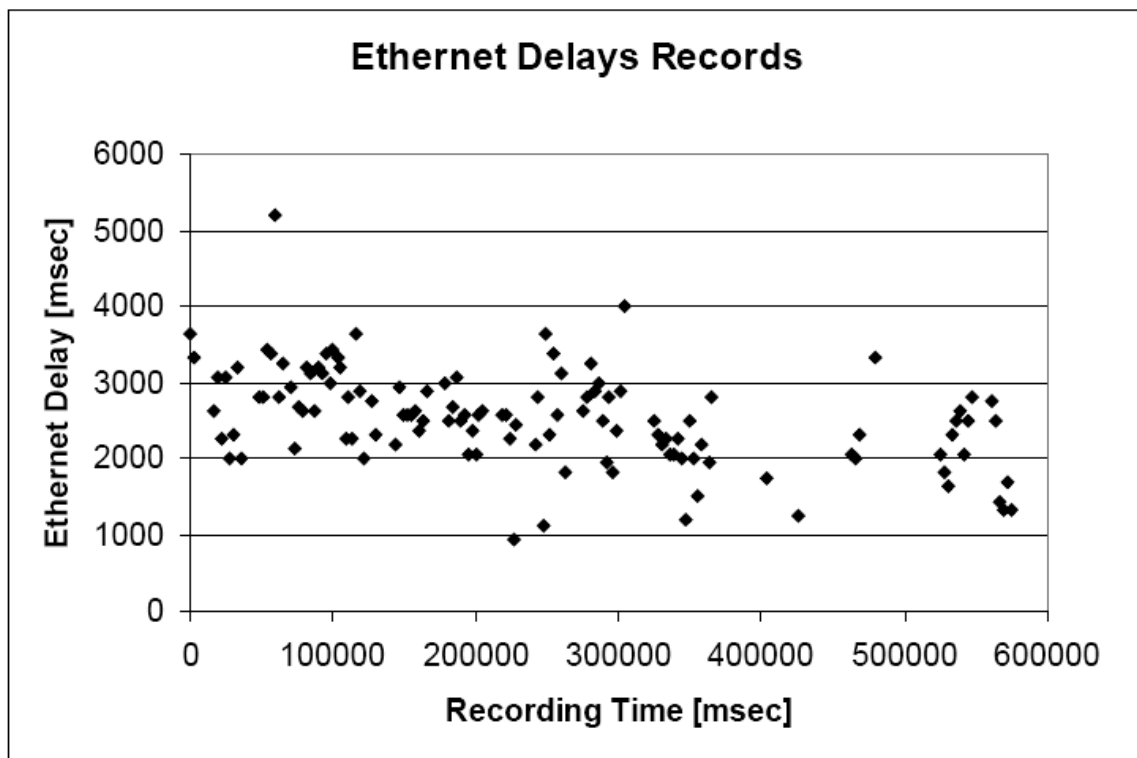


Figure 2: The experimental results of recording Ethernet Delays over around 10 minutes on the analyzed network.

availability of each possible plate destination stations (of each Robotic Cell).

The proposed solution requires the Routing System to move the CMS cell data hand-off point (the point where it is transmitting the Routing Data to CMS) more upstream the conveyor.

The CMS has to receive the Routing Data message of a plate before the Vision Systems ends the process of analyzing that plate (even on the highest Ethernet traffic). But the method is not anymore affected by receiving Routing Data in advance. CMS will build a buffer of all the Routing Data received from the RCS describing plates will come in time, and it will be able to recover the consistency in time of these data on the merging process with the Vision Data.

5 CONCLUSIONS

The paper presented an algorithm developed at protocol level, in order to guarantee the recovering of data transfer consistency in time, at CMS level, for merging of the data provided from RCS over an Ethernet communication channel with the data

provided by the Vision System over a deterministic communication channel.

This solution proposed in this paper is based on the following assumptions:

- *The actual slippage of the clocks on either the RCS or the CMS processors would be very minimal.* This assumption is not a restrictive one, being normal to accumulate a significant slippage value around one second in much more than days or even weeks.
- *In the time the slippage value of the clocks would become significant, the network connection would have a period of relative low traffic.* This assumption is also not restrictive one, because in those days or weeks that the slippage value of the clocks is becoming significant, it is more likely a relative calm moment to occur in the net traffic.
- *The number of collected messages transmitted by the producer/received by the consumer (till the slippage of the CPU clocks will accumulate a significant value) will be large enough to build a statistical population.* This assumption is also not a restrictive one because the statistical population main support is the "Request Status Message" which is set to

be transmitted every half second (most of the statistical population members are coming from collecting the estimated offsets for this type of message).

REFERENCES

- Mackay S., Wright E., Reynders D., Park J., 2003, *Practical Industrial Data Networks*, Elsevier.
- Marshall P.S., Rinaldi J.S., 2004, *Industrial Ethernet (2nd Edition)*, ISA – The Instrumentation, Systems, and Automation Society.
- Stenerson J., 2002, *Industrial Automation and Process Control*, Prentice Hal.

THE APPLICATION OF REFERENCE-PATH CONTROL TO VEHICLE PLATOONS

Drago Matko, Gregor Klančar, Sašo Blažič

*Faculty of electrical engineering University of Ljubljana, Slovenia
drago.matko@fe.uni-lj.si, gregor.klancar@fe.uni-lj.si, saso.blazic@fe.uni-lj.si*

Olivier Simonin

*lab: LORIA Maia project, University of Henri Poincare, Nancy, France
olivier.simonin@loria.fr*

Franck Gechter, Jean-Michel Contet, Pablo Gruer

*Systems and Transportation Laboratory (SET), University of Technology of Belfort-Montbéliard (UTBM), Belfort, France
franck.getcher@utbm.fr, jean-michel.contet@utbm.fr, pablo.gruer@utbm.fr*

Keywords: Platoon, reactive multiagent, longitudinal and lateral control, reference-path following control.

Abstract: A new algorithm for the control of vehicle platooning is proposed and tested on a robot-soccer test bed. We considered decentralized platooning, i.e., a virtual train of vehicles, where each vehicle is autonomous and decides on its motion based on its own perceptions. The platooning vehicles have non-holonomic constraints. The following vehicle only has information about its own orientation and about its distance and azimuth to the leading vehicle. Its position is determined using odometry and a compass. The reference position and the orientation of the following vehicle are determined by the estimated path of the leading vehicle in a parametric polynomial form. The parameters of the polynomials are determined using the least-squares method. This parametric reference path is also used to determine the feed-forward part of the applied control algorithm. The feed-back control consists of a state controller with three inputs: the longitudinal and lateral position errors and the orientation error. The results of the experiments demonstrate the applicability of the proposed algorithm for vehicle platoons.

1 INTRODUCTION

Vehicle platoon systems are a promising approach for new transportation systems because of their innovative capabilities. Their main goals, when applied to passenger cars are (i) an increase in the vehicle density on the highway (i.e., avoiding traffic jams), and (ii) security improvements thanks to automated or semi-automated driving assistance (adaptive cruise control, obstacle detection and avoidance, automatic car parking, etc.). Most of these platooning systems are based on a linear configuration (i.e., a virtual train of vehicles).

Among the several problems associated with the control of platooning systems, longitudinal and lateral control are the most important.

Longitudinal control involves controlling the braking and acceleration in order to stabilize the distance between the leading vehicle and the following vehicle. This control takes as a parameter the distance between the leading and the following ve-

hicles. Sheikholeslam and Desoer (Sheikholeslam and Desoer, 1993) proposed a form of longitudinal control based on linearization methods. Ioannou and Xu (Ioannou and Xu, 1994) controlled the brakes and the acceleration using a fixed-gain PID control with gain scheduling. In contrast, Hedrick, Tomizuka and Varaiya (Hedrick et al., 1994) used a control mode based on a non-linear method with PID. Lee, Tomizuka, Jung and Kim (Lee and Tomizuka, 2003; Lee et al., 2000) proposed a longitudinal control based on fuzzy logic.

Lateral control involves aligning the vehicle's direction relative to the vehicle in front. Daviet and Parent (Daviet and M.Parent, 1996) proposed a form of lateral control using a PID controller. This control consists of keeping the angle between the leading and the following vehicles close to zero. In the literature, papers can be found dealing with lateral and longitudinal control using physics-inspired models. For instance, Gehrig and Stein (Gehrig and Stein, 2001) designed a model based on particles'

submissive forces, whereas Yi and Chong (Yi and Chong, 2005) developed an impedance-control immaterial hook model. Halle and Chaib-draa (Halle and Chaib-draa, 2005) used a Multi-Agent System (MAS) in order to model immaterial vehicles using constant values from (Daviet and M.Parent, 1996). Contet, Gechter, Gruer and Koukam (Contet et al., 2007) proposed a solution for longitudinal and lateral control using Newtonian forces in an interactive model. In Bom et. all (Bom et al., 2005) a global platooning control strategy is proposed using nonlinear control law which decouples lateral and longitudinal control.

In this paper a novel approach to a platoon of non-holonomic vehicles using the well-known state-space control of nonholonomic systems is presented. The vehicle platooning control strategy relies on relative information to preceding vehicles only therefore no explicit inter-vehicle data exchange and global global information (such as GPS) is required. The important advantage here is that relative information can be measured with low cost sensor sets. Additionally the method to obtain on-line objectives for the follower vehicles control is presented, where the inter-vehicle distance is curvilinear one as also proposed in (Bom et al., 2005). The proposed control algorithm was tested in simulations and on a platoon of soccer robots.

Controlling nonholonomic systems as they follow a reference path is a well-known problem that has been studied by many authors (Kolmanovsky and McClamroch, 1995; Luca and Oriolo, 1995; Sarkar et al., 1994). The control of vehicles, especially mobile robots, by considering only the first-order kinematics is very common in the literature ((Canudas de Wit and Sordalen, 1992; Oriolo et al., 2002; Balluchi et al., 1996)) as well as in practice. The vehicle has to consider nonholonomic constraints, so its path cannot be arbitrary. Moreover in an environment with obstacles, limitations and other demands the vehicle should be controlled on a reference path, which should follow all the kinematic constraints and avoids obstacles.

The paper is organized as follows: In Section 2 a model of nonholonomic systems and the corresponding control law that can be applied to such systems are presented. The application of the proposed control law to platoon systems is derived in Section 3. The results of the tests on a robot-soccer set-up are presented in Section 4.

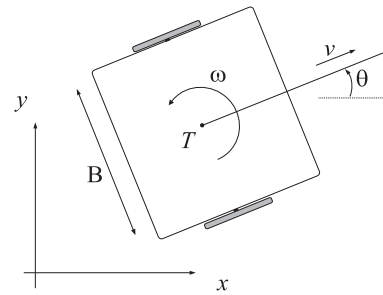


Figure 1: Vehicle architecture and symbols.

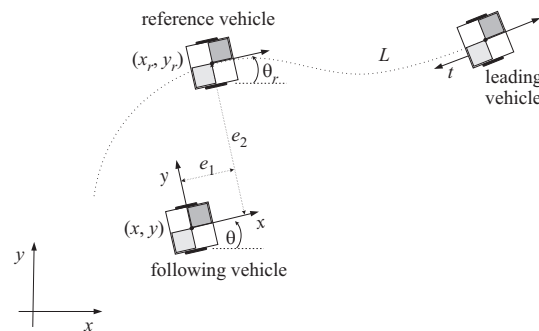


Figure 2: Illustration of the error transformation.

2 MODELING AND CONTROL OF NONHOLONOMIC SYSTEMS

In the following the direct and inverse kinematics for mobile vehicles with a differential drive are determined. The vehicle's architecture, together with its symbols, is shown in the Fig. 1, where it is supposed that the geometrical centre T and the centre of gravity coincide.

The equations of motion are as follows

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

where v and ω are the tangential and angular velocities of the platform shown in the Fig. 1. The right and left velocities of the vehicle's wheels are then expressed as $v_R = v + \frac{\omega B}{2}$ and $v_L = v - \frac{\omega B}{2}$, respectively, where B is the distance between the robot wheels.

For a given reference trajectory $(x_r(t), y_r(t))$ defined in the time interval $t \in [0, T]$ the feed-forward control law can be derived. From the obtained inverse kinematics the vehicle inputs are calculated, these drive the vehicle on the desired path only if there are no disturbances and no initial state errors. The required vehicle inputs, the tangential velocity v_{ff} and

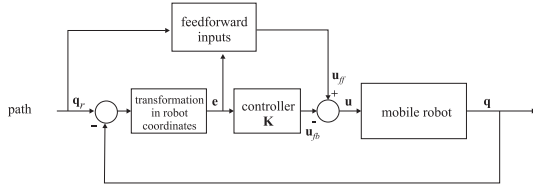


Figure 3: Mobile-vehicle control schematic.

the angular velocity ω_{ff} , are calculated from the reference path. The tangential velocity is given by

$$v_{ff}(t) = \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (2)$$

$$\omega_{ff}(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (3)$$

When a vehicle is controlled to drive on a reference path, it usually has some following error. This following error, expressed in terms of the real vehicle, as shown in the Fig. 2, is given by

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (4)$$

In the Fig. 2 the reference vehicle is an imaginary vehicle that ideally follows the reference path. In contrast, the real vehicle (when compared to the reference vehicle) has some error when following the reference path. Therefore, the control algorithm was designed to force the vehicle to follow the reference path precisely as proposed in (Luca and Oriolo, 1995; Oriolo et al., 2002). It is as follows

$$\begin{aligned} v_r &= v_{ff} \cos e_3 - v_{fb} \\ \omega_r &= \omega_{ff} - \omega_{fb} \end{aligned} \quad (5)$$

where v_r and ω_r are reference velocities (set-points) for the low level control controlling the wheels of the vehicle and v_{fb} , ω_{fb} are the outputs of the feedback controller given by

$$\begin{bmatrix} v_{fb} \\ \omega_{fb} \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -\text{sign}(u_{ff})k_2 & -k_3 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (6)$$

The schematic of the obtained control is explained in Fig. 3. The gains k_1 , k_2 and k_3 of the state feedback controller \mathbf{K} were determined by trial and error.

3 APPLICATION OF THE CONTROLLER TO A LINEAR PLATOON

It is supposed that there is no data communication between the leading and following vehicles. The following vehicle measures the distance and the azimuth (relative to its own orientation) of the leading vehicle. To ensure stable control also a measurement of the orientation of the following vehicle (e.g. with a compass) is also needed. No other sensors (e.g., GPS) are required. All the positions are treated in a coordinate system that is fixed to the ground. The following vehicle determines its own position using odometry. Having the current position $\mathbf{X}(k) = [x(k), y(k)]^T$, the position in the next sample is determined by a simple Euler integration

$$\mathbf{X}(k+1) = \mathbf{X}(k) + \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} v_{ref} \Delta t \quad (7)$$

where θ is the orientation of the following vehicle, v_{ref} is the reference speed of the vehicle and Δt is the sample time. As shown later, the method of integration and the associated errors in the accuracy of the absolute position are not significant, since only the relative position of both vehicles is important.

The path of the leading vehicle $\mathbf{X}_h(k) = [x_h(k), y_h(k)]^T$ is calculated by the following vehicle using its current position and the measurements of the distance D and the azimuth θ_a (e. g., by using a laser range finder) as follows:

$$\mathbf{X}_h(k) = \mathbf{X}(k) + \begin{bmatrix} \cos(\theta + \theta_a) \\ \sin(\theta + \theta_a) \end{bmatrix} D \quad (8)$$

This information is stored in the memory and represented in parametric form (with the parameter k - related in the time $t = k\Delta t$). The following vehicle is supposed to track the leading vehicle at a distance L - measured on the path of the leading vehicle. First, the time T needed by the leading vehicle to drive the distance L is calculated using

$$L = \int_0^T \sqrt{\dot{x}_h^2 + \dot{y}_h^2} dt \quad (9)$$

This time T is calculated by a linear interpolation of the two successive time instants ($k+1$ and k) defining the time interval where the numerically calculated distance L' becomes greater than the desired distance L .

$$L' = \sum_{k=0}^N \sqrt{[x_h(k+1) - x_h(k)]^2 + [y_h(k+1) - y_h(k)]^2} \quad (10)$$

According to relation (4) the interpolated value for time T is obtained by $T = k\Delta t + \frac{\Delta t}{L'(k+1) - L'(k)}(L - L'(k))$, where L is the desired tracking distance among the vehicles. Next, the path shape of the leading vehicle at the moment $-T$ (T seconds in the past) is expressed in the parametric polynomial form

$$x_h(t) = a_2^x t^2 + a_1^x t + a_0^x \quad (11)$$

$$y_h(t) = a_2^y t^2 + a_1^y t + a_0^y \quad (12)$$

The coefficients of the polynomials a_i^x and a_i^y are calculated using the least-squares method with more than three samples around the time T (seven were used in our experiments). The reference position and the orientation of the following vehicle are determined using

$$\mathbf{X}_r = \begin{bmatrix} x_h(T) \\ y_h(T) \end{bmatrix} = \begin{bmatrix} a_2^x T^2 + a_1^x T + a_0^x \\ a_2^y T^2 + a_1^y T + a_0^y \end{bmatrix} \quad (13)$$

$$\theta_r = \arctan \frac{2a_2^y T + a_1^y}{2a_2^x T + a_1^x} \quad (14)$$

respectively. In the Fig. 2 they are denoted as the reference vehicle. The tangential and angular velocities of the reference vehicle (needed for the feed-forward control) are

$$v_r(t) = \sqrt{(2a_2^x T + a_1^x)^2 + (2a_2^y T + a_1^y)^2} \quad (15)$$

and

$$\omega_r(t) = \frac{(2a_2^x T + a_1^x) \times 2a_2^y - (2a_2^y T + a_1^y) \times 2a_2^x}{(2a_2^x T + a_1^x)^2 + (2a_2^y T + a_1^y)^2} \quad (16)$$

respectively. For the feed-back control the error vector is given according to Eq. (4) by

$$e = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_r - \mathbf{X} \\ \theta_r - \theta \end{bmatrix} \quad (17)$$

4 RESULTS OF THE EXPERIMENTS

The proposed algorithm was tested on a robot-soccer set-up (see Fig. 4) consisting of three Middle League MiroSot category robots of size 7.5 cm cubed, a digital color camera and a personal computer. The color camera mounted above the pitch is a global sensor. The vision part of the programme ((Klančar et al., 2004)) processes the incoming image to identify the positions and orientations of the robots. The first, (leading) robot was driven on a prescribed path. The

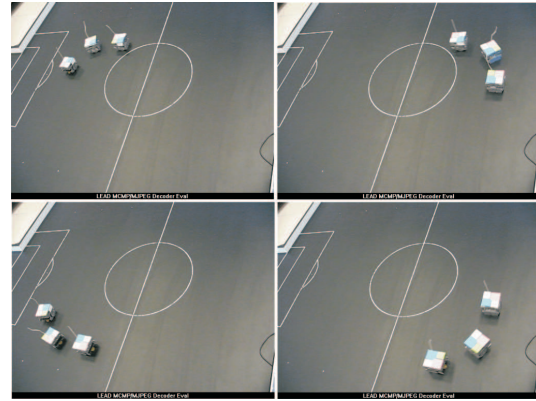


Figure 4: Real set-up experiment.

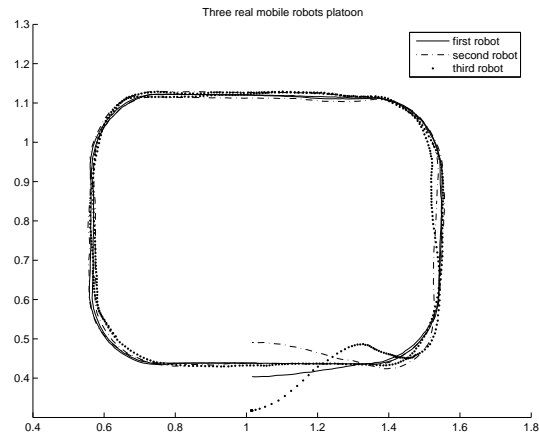


Figure 5: Results of real experiments.

second (the first following) robot receives only the information about its distance and azimuth to the first robot and its own orientation. The third (the second following) robot receives only the information about its distance and azimuth to the second robot and its own orientation. The noisy position estimates of the used camera sensor influences the calculated distance and azimuth information. The estimated noise deviation of measured robots positions was $\pm 5mm$. The distances and azimuth orientations are obtained by $D_i = \sqrt{(x_{i-1} - x_i)^2 + (y_{i-1} - y_i)^2}$ and $\theta_{ai} = \arctan \frac{y_{i-1} - y_i}{x_{i-1} - x_i}$, where $i = 2, 3$ is robot index.

The parameters values of the controller (5) were $k_1 = 2, k_2 = 20, k_3 = 2$, sampling time was $\Delta t = 33ms$ and the desired tracking distance was $L = 20cm$. The results of the tests are shown in the Fig. 5. The film of the real experiment can be seen at (Klančar, 2008).

In the Fig. 6 the time course of the distance between the robots is presented. The distance was calculated with assumption that the path between the

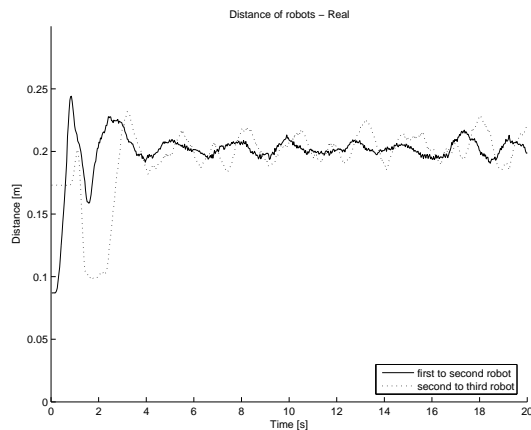


Figure 6: Distance between robots - experiment.

robots is an arc, which results in

$$L_{arch} = \frac{\Delta\theta}{2 \sin(\Delta\theta/2)} \times D \quad (18)$$

where D is the straight line between the robots and $\Delta\theta$ is the difference in their orientation angles. It is clear that after a transition phase (the merging and splitting of the platoons is currently under investigation) the second and third vehicle follow with acceptable accuracy. The results of the real experiments are slightly worse due to the noise in the position estimation and due to the time delay of the optical tracking and recognition. The accuracy of the integration method and the associated error, which is equivalent to the slipping of the vehicle's wheels, is analysed and illustrated in the Fig. 7, where the distance between the leading and the following platoon robots in a straight path is illustrated. It can be seen that the constant slipping of the wheels has no influence on the steady-state distance of the platoon vehicles. This conclusion makes sense since servoing accuracy should not be destroyed because relative information among vehicles (distances and azimuth orientations) are always obtained from accurate relative sensor.

5 CONCLUSIONS

A new algorithm for the control of vehicle platoons was proposed. The following vehicle only has information about its own orientation and about the distance and azimuth of the leading vehicle. Its own position is determined using odometry and a compass. It calculates the reference path in a parametric polynomial form, and the parameters of the polynomials are determined by the least-squares method. Having the reference path, the feed-forward and feed-back con-

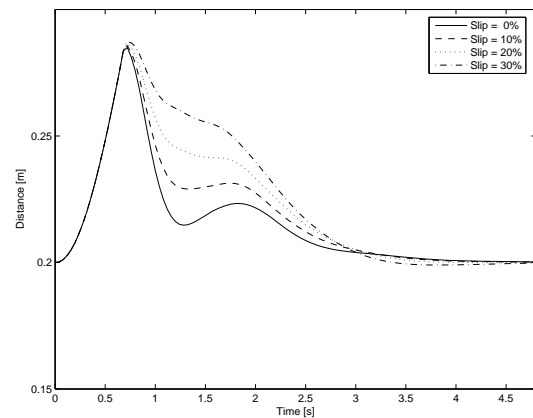


Figure 7: Distance between robots with slip in a straight path (simulation).

trol are applied to the following vehicle. The following vehicle calculates its own position by means of a simple Euler integration. It was established that the error in the integration procedure (equivalent to the errors due the wheel slipping) has a minor influence on the accuracy of the platoon distance. The proposed algorithm was tested on a robot-soccer test bed. The results confirm the applicability of the proposed method.

REFERENCES

- Balluchi, A., Bicchi, A., Balestrino, A., and Casalino, G. (1996). Tracking control for dubin's cars. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota*, pp. 3123-3128.
- Bom, J., Thuilot, B., Marmoiton, F., and Martinet, P. (2005). A global control strategy for urban vehicles platooning. relying on nonlinear decoupling laws. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta*, pp. 1995-2000.
- Canudas de Wit, C. and Sordalen, O. J. (1992). Exponential stabilization of mobile robots with nonholonomic constraints. In *IEEE Transactions on Automatic Control, Vol. 37, No. 11*, pp. 1791-1797.
- Contet, J., Gechter, F., Gruer, P., and Koukam, A. (2007). Application of reactive multiagent system to linear vehicle platoon. In *Annual IEEE International Conference on Tools with Artificial Intelligence(*ICTAI*)*, Greece, Patras.
- Daviet, P. and M.Parent (1996). Longitudinal and lateral servoing of vehicles in a platoon. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 41-46.
- Gehrig, S. K. and Stein, F. (2001). Elastic bands to enhance vehicle following. In *IEEE Conference on In-*

- telligent Transportation Systems, Proceedings, ITSC, pages 597-602.*
- Halle, S. and Chaib-draa, B. (2005). A collaborative driving system based on multiagent modeling and simulations. In *Transp. Res. C, Emerg. Technol. (UK)*, 13(4): 320 - 45.
- Hedrick, J., Tomizuka, M., and Varaiya, P. (1994). Control issues in automated highway systems. In *IEEE Control Systems Magazine*, 14(6):21-32.
- Ioannou, P. and Xu, Z. (1994). Throttle and brake control systems for automatic vehicle following. In *IVHS Journal*, 1(4):345.
- Klančar, G. (2008). <http://msc.fe.uni-lj.si/publicwww/klancar/robotsplatoon.html>.
- Klančar, G., Kristan, M., and S. Kovačič, O. O. (2004). Robust and efficient vision system for group of cooperating mobile robots with application to soccer robots. In *ISA Transactions*, vol. 43, pp. 329-342.
- Kolmanovsky, I. and McClamroch, N. H. (1995). Developments in nonholonomic control problems. In *IEEE Control Systems*, Vol. 15, No. 6, pp. 20-36.
- Lee, H. and Tomizuka, M. (2003). Adaptive vehicle traction force control for intelligent vehicle highway systems (ivhss). In *IEEE Transactions on Industrial Electronics*, 50(1):37-47.
- Lee, M., Jung, M., and Kim, J. (2000). Evolutionary programming-based fuzzy logic path planner and follower for mobile robots. In *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, (1) pp 139-144.
- Luca, A. and Oriolo, G. (1995). Modelling and control of nonholonomic mechanical systems. In *Kinematics and Dynamics of Multi-Body Systems*, Springer-Verlag, Wien.
- Oriolo, G., Luca, A., and Vandittelli, M. (2002). Wmr control via dynamic feed-back linearization: Design, implementation, and experimental validation. In *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 6, pp. 835-852.
- Sarkar, N., Yun, X., and Kumar, V. (1994). Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robot. In *The International Journal of Robotic Research*, Vol. 13, No. 1, pp. 55-69.
- Sheikholeslam, S. and Desoer, C. (1993). Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: A system level study. In *IEEE Transactions on Vehicular Technology*, 42(4):546-554.
- Yi, S.-Y. and Chong, K.-T. (2005). Impedance control for a vehicle platoon system. In *Mechatronics (UK)*, 15(5):627-38.

DESIGN AND BALANCING CONTROL OF AIT LEG EXOSKELETON-I (ALEX-I)

Narong Aphiratsakun, Kittipat Chirungsarpsook and Manukid Parnichkun
Asian Institute of Technology, P.O.Box 4, Klong Luang, Pathumthanee, 12120, Thailand
st103902@ait.ac.th, st104691@ait.ac.th, manukid@ait.ac.th

Keywords: ALEX-I, Exoskeleton, robot suit, balancing control, mechanical design.

Abstract: This paper is focused on the design of mechanical hardware, controller architectures, and analysis of balancing control at the Asian Institute of Technology Leg EXoskeleton-I (ALEX-I). ALEX-I has 12 DOF (6 DOF for each leg: 3 at the Hip, 1 at the knee and 2 at the ankle), controlled by 12 DC motors. The main objective of the research is to assist patients who suffer from the paraplegia and immobility due to the loss of lower limbs. ALEX-I's parts and assembly are designed on CAD software, SolidWorks, exported to MATLAB simulation environment, and observed using 3D VRML script interpreter to investigate balancing postures of the exoskeleton. The simulation model is proven to be accurate by comparing the resulting kinematics characteristics with the results from Corke's MATLAB Robotics Toolbox (Corke, 1996). PC104 is employed as the main (master) processing unit for calculation of the balanced gait motion corresponding to feedback signals from the force sensors mounted at the two feet plates, whereas ARM7's are used for the low-level (slave) control of the angular position of all joints. The balanced posture set-points (joint trajectories) under the Center of Mass (CM) Criterion are generated in the simulation before testing on the real mechanical parts is implemented to avoid damaging the system.

1 INTRODUCTION

Our society nowadays has many elders and patients that have difficulties in their locomotion. All of these patients need to sit, stand, walk, and perform other activities to fulfil their daily tasks. These people need assistance from either the nursing personnel or assistive devices such as walkers or wheelchairs. Our exoskeleton is intended to work as an intelligent assistive device that would help eliminating the difficulties and risks during the locomotion of the wearer. For this purpose, the exoskeleton has to be able to balance itself, carry the wearer, and walk even if the lower part of the patient is completely paralyzed. In addition to improving the quality of many lives, the developed exoskeleton can also serve as a tool used to imitate and integrate human natural blueprints.

Exoskeleton systems also find their applications in other various fields that draw a lot of interests from many robotics researchers who want to imitate the perfectly-designed and sophisticated biomechanics and human anthropometries. Some of the successful stories are HAL (Kawamoto, Kanbe, Lee and Sankai, 2002 and 2003), BLEEX (Chu, Kazerooni, Zoss, Racine, Huang and Steger, 2005),

and Sarcos (Guizzo and Goldstein, 2005) exoskeletons, which are designed for power enhancing and military missions respectively. HAL-3 was developed by the research team of Tsukuba in Japan. It was designed to help the elders in performing their daily activities such as walking, sitting, and standing. The latest model, HAL-5, is the whole-body suit unit, which is suitable for either the left or the right side paraplegic patient. BLEEX developed by the University of California, Berkeley, and Sarcos developed at Sarcos Research Corp. in Salt Lake City implemented the hydraulic-actuated exoskeletons as they are focusing on the power-enhanced legs for the application of carrying heavy loads in the difficult terrains.

Asian Institute of Technology Leg EXoskeleton-I or known as ALEX-I is developed with the aim to carry with it both the external loads and the pilot (or the wearer). The exoskeleton has to be able to walk on its own. Building up the robot and physically testing it by means of trial-and-error could result in damaging the robot links and fragile electronics devices. Hence, we have to model the exoskeleton robot to conduct the experiments in the both real world and simulated environments. The simulation model of ALEX-I has shown promising results

through the modelling with MATLAB's SimMechanics library. Consequently, precise gait pattern generation can be investigated based on the kinematics information of all moving bodies. This simulation model can serve as the framework for development of the whole-body exoskeleton and all types of biped robots, which will be developed in the future at AIT.

This paper describes the analysis of the architecture layout of the ALEX-I system in both software and mechanical hardware. The mechanical properties and controllers layout of the ALEX-I will be explained in the next 2 sections. The simulation model of the exoskeleton, its the interpretation of experimental result in 3D virtual reality (VR) environment, as well as the example of gait pattern generation of one-step gait motion will be discussed in section 4.

2 MECHANICAL DESIGN

Our previous work (Aphiratsakun and Parnichkun, 2007) reveals the required specification of the 12 actuators through the required torque calculations of all the joints. The range of motion of the joints determined in the previous work is refined to disregard the range that will never be employed in the real physical implementation, and the resulting range of motion of all DOF is shown in Table 1.

The ALEX-I has 12 DOF (6 DOF for each leg: 3 at the Hip, 1 at the knee and 2 at the ankle), controlled by 12 DC motors. Each motor is coupled with a 1:100 gearhead and equipped with a 1024-pulse incremental encoder as a feedback sensor. The Scooter DC motors and Bonfiglioli Gearhead model VF44P63B14 are selected in this work to conform to the required flexibility in the mounting structures, shapes, and weights. Table 2 gives specification of the motors and the gearheads mounted on each joint. Obviously, the torques offered by the gearheads in each joint are in comply with the torque requirements revealed in (Chu, Kazerooni and Zoss, 2005).

Table 1: Range of motion of each joint.

Joint	Axis	Range of rotation (degree)
Hip	X (pitch)	-90<0<90
	Y (yaw)	-35<0<35
	Z (roll)	-15<0<15
Knee	X (pitch)	0<0<90
Ankle	X (pitch)	-45<0<45
	Z (roll)	-20<0<20

Table 2: Specification of the motors coupled with 1:100 gearhead at each joint.

Motors	Joints the motors mounted on	RPM ; Rad/s	Torque [Nm]
250 W	Hip (yaw)	25 ; 2.62	95
350 W	Hip (roll), Ankle (roll)		134
	Hip (pitch), Knee (pitch), Ankle (pitch)		191

The lower limb exoskeleton mechanical parts are designed with a CAD Application, SolidWorks, as shown in Figure 1. The anthropometric considerations and other design parameters are discussed in (Aphiratsakun and Parnichkun, 2007). This CAD assembly can be imported to MATLAB development environment, which will be used to analyze for the balanced gait motion through the simulation model. The simulation model will be revisited in section 4.

With the CAD design, aluminum 5083 with the density of 2657.27 Kg/m³ is mainly used for the frame structure. The front and back views of the fabricated prototype is shown in Figure 2. The weight of the ALEX-I is measured to be 117.5 Kg excluding the weight of the bag pack.

Force sensors or load cells are used to measure the forces exerted by the body. Futek LLB400 load cell, which can measure up to 500 lb (2224 N) of force, is chosen in the implementation. Four of these sensors are placed between two plates of the ALEX-I's feet. INA126 micro power instrumentation amplifier is used as the amplifier for the load cell. The designed layout of the load cell and its amplifier circuit is shown in Figure 3. From the force reading from the load cells, the center of mass (CM) position could be calculated.

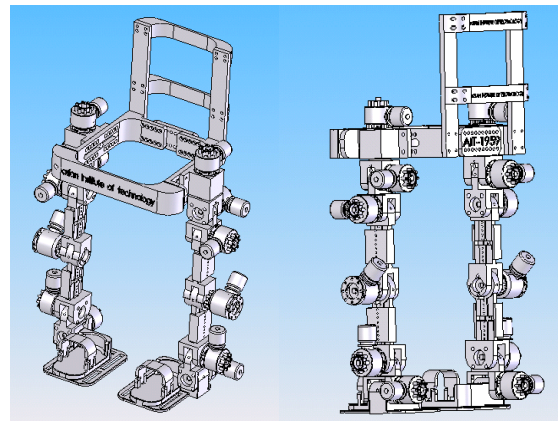


Figure 1: Prototype design of exoskeleton frame (lower part) (a) front view and (b) back view.

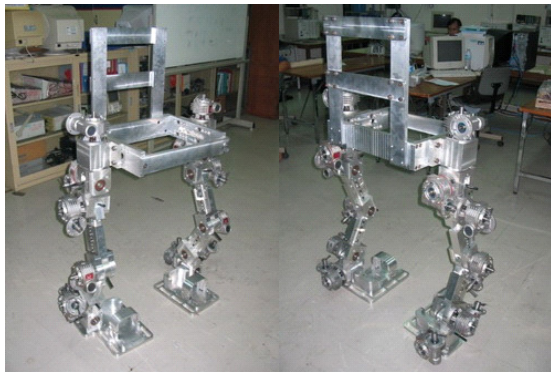


Figure 2: Front and back views of the ALEX-I mechanical frame.

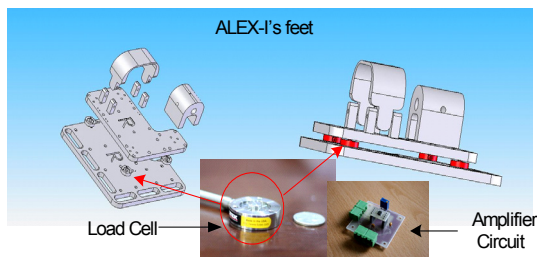


Figure 3: Load cells arrangement.

3 CONTROLLER ARCHITECTURE

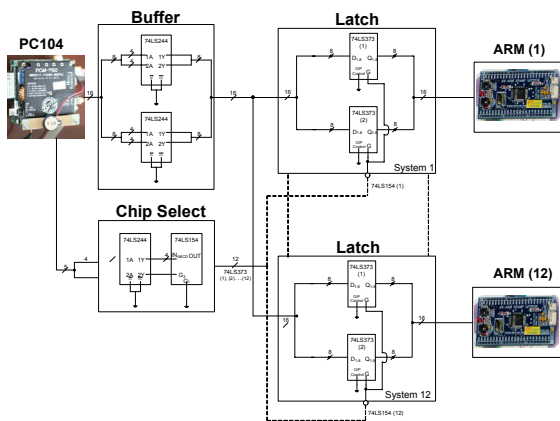


Figure 4: Overview of High and Low level controllers' architecture.

The data of gait analysis from the simulation is used as the input for positional control of the motors, which will eventually make ALEX-I walk in the desired motion. In this work PC104 and ARM7 LPC2138 are used as the high and low level controllers respectively. The overview of the

controllers' layout is shown in Figure 4. The twelve set-points data for the joints' trajectories, which are sent from PC104, are stored in the latching circuit to eliminate the lag time that might be incurred from serial communication. Chip selecting circuit is then used to address each slave-controller with its proper set point. Putting these set-points data in parallel manner allows low-level controller to acquire the data without delay.

3.1 Joint Controller: ARM7 LPC2138 Microcontroller

The joint controller block set is shown in Figure 5. The twelve sets of 16-bits set points command are sent from PC104, which configures the required motion balancing tasks for the whole system, as the input to the low-level close-loops that comprise 2 closed loops (P and PD) for each control block: speed and position loops. 10 Bits, 1024 pulses/rev Koyo TRD-S1024V series incremental encoder is used as a feedback sensor at each joint. LS7366 by LSI, is used to obtain the quadrature A/B of the incremental encoder signal. This IC communicates through SPI with ARM7 processor and increases the quadrature counting up to four times. It increases the resolution of the encoders to 4096 pulses/rev. Axor MicrospeedPlus is chosen as the servo driver and interfaced between ARM7 as shown in Figure 6.

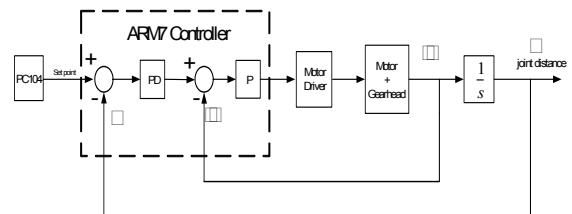


Figure 5: Joint controller.

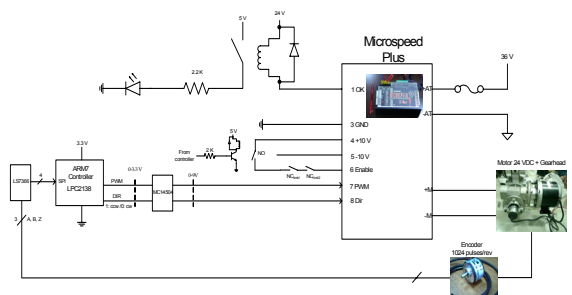


Figure 6: Servo interfacing circuit.

4 SIMULATION MODEL

This section shows the model of ALEX-I in its simulation environment and the gait pattern generation. All links and joints are modelled with their real inertia matrices, links' location of centers of gravity (CG), and location of joints as calculated automatically in Solidworks's mass properties command. Our simulation approach allows the researcher to keep track of all joints' and links' kinematics and dynamics properties very precisely through virtual sensors. Firstly, the ALEX-I SimMechanics model is verified with simple Denavit-Hartenberg (DH) matrix for analysis of manipulator's end-effector to verify the correctness of our simulation model. The balanced gait motion is also performed and shown in latter part of the section.

4.1 Model Verification with Denavit-Hartenberg Matrix and P. I. Corke's MATLAB Robotics Toolbox

To verify the correctness of the simulation modelling, the authors use the DH matrix in describing the 12-DOF ALEX-I assuming that the left ankle is fixed to the ground as if the whole robot is a 12-DOF manipulator with the right ankle being the end-effector. Obviously, the dynamics behavior of the robot with this assumption does not match the real situation. However, the position and velocity obtained from the DH and Jacobian matrix consideration proves our simulation model to be quite accurate in terms of kinematics characteristics. Figure 7 and Table 3 conclude the properties of links as defined by DH (Craig, 2005).

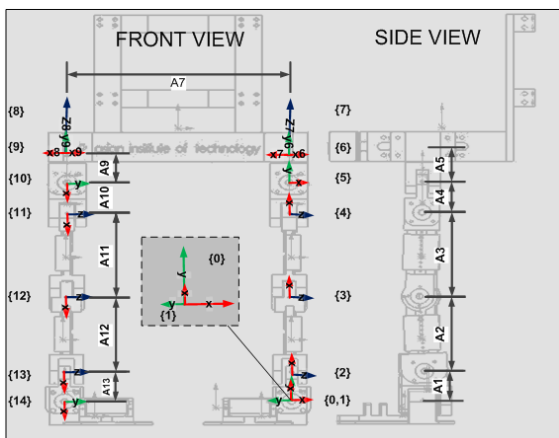


Figure 7: Assignment of coordinate systems in accordance to DH.

Table 3: Links' properties required for the calculation of DH transformation matrixes and Jacobian matrices.

Link	α [degree]	a [m]	θ [degree]	d (m)
1	90	0.1	90	0
2	0	0.26	0	0
3	0	0.28	0	0
4	90	0.11	0	0
5	0	0.12	0	0
6	-90	0	90	0
7	0	0.594	0	0
8	-90	0	0	0
9	0	0.12	-90	0
10	-90	0.11	0	0
11	0	0.28	0	0
12	0	0.26	0	0
13	90	0.1	0	0

Applying the links' properties in Table 3 with the Robotics Toolbox written (Corke, 1996), we obtain another version of stick diagram as shown in Figure 8. The toolbox calculates transformation matrix referred from the end-effector (right ankle) to the world coordinate (left ankle) as similar to that of our simulation model with accuracy of 1 millimeter as shown in the highlighted numerical data. The left circle highlights the transformation matrix resulted from the Corke's procedures whereas the right circle is the data from our developed simulation model.

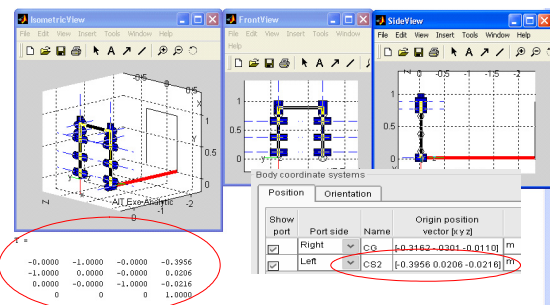


Figure 8: Simulation result from P. I. Corke Robotics Toolbox (Corke, 1996).

4.2 MATLAB Physical Model

The simulation modelling of the 12-DOF ALEX-I has to be started with the transformation of CAD data from the CAD application, SolidWorks, to the physical model format in MATLAB's

SimMechanics. The mating functions and mass properties are automatically translated into the joints and links with precise inertia matrices and joint coordinate systems location as referred from the grounded position. Figure 9 shows the flow of how the precise simulation model can be created from CAD assembly file format. Apparently, this simulation model is very accurate in resembling the real physical exoskeleton as it is created from the exact sizes, mass and inertia properties, and joint locations of the real fabricated links and assembled robot. In the figure, the block diagrams with the signs of the CGs and the signs of 5-DOF represent the robot links and joints as defined in the mating function respectively. Figure 10 shows the imported frontal and lateral views of the ALEX-I model.

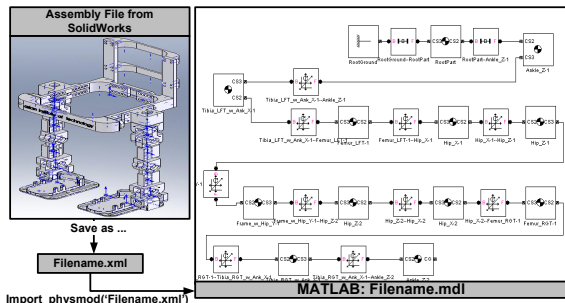


Figure 9: Importing the physical model from SolidWorks assembly file.

4.3 SimMechanics Virtual Sensors

Development of the simulation model on the MATLAB environment offers great advantages since the SimMechanics Library offers virtual sensors that allow monitoring of kinematics and dynamics properties of all moving bodies and joints, including the monitoring of position, velocity, acceleration, angular displacement, angular velocity, angular acceleration, reaction force, and reaction torque. More importantly, the SimMechanics also offers virtual actuators that allow the actuation of both the joints and the bodies by the Source toolbox in the Simulink Library. With the virtual tools offered by SimMechanics Library, the manipulation of all kinematics and dynamics parameters could be done and monitored so as to study the motion behaviour and gait generation of the ALEX-I in virtual environment.

Nevertheless, the numerical data observed from the virtual sensors does not give understandable interpretation unless applied with the graphical visualization. The authors create the graphical interpretation of results both in the forms of 2D

MATLAB graphics and in the 3D Virtual Reality (VR) environment. Figure 11 shows how the motion signals could be used as input to the 3D VR graphics.

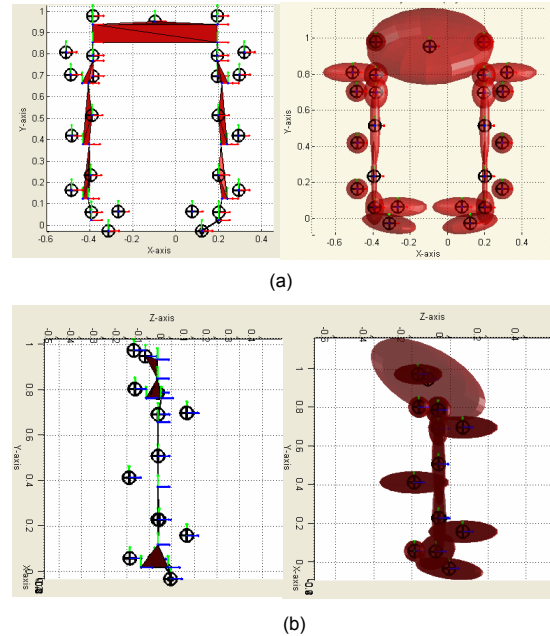


Figure 10: (a) Frontal and (b) Lateral views of transformed diagram.

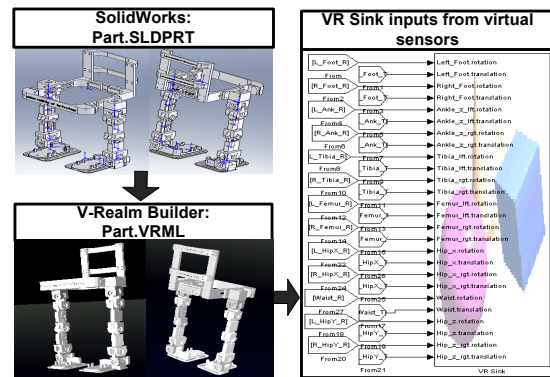


Figure 11: Procedures for the 3D Virtual Reality Graphical Interpretation.

Figure 13 shows the illustrated stick diagram, ellipsoidal mass-represented diagram, and 3D VR animation respectively. The input signals captured from the virtual sensors are fed to each joint for the angular position of all joints. The signals inputted to create the corresponding posture in Figure 13 are shown in Table 4 and Figure 12. θ_{1-12} in the Table 4 are left ankle [z,x], left knee [x], left hip [x,z,y], right hip [y,z,x], right knee [x], right ankle [x,z]

respectively. On the other hand, the input signals observed from virtual scope in Figure 13 are listed as the right ankle [z,x], right knee[x], right hip [x,z], left hip [z,x], left knee [x], and left ankle [x,z] in the order from the top to the bottom.

Table 4: 12 set-points angles.

Angles (degree)					
θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
0.294	36.88	58.95	-66.58	-14.98	0
Angles (degree)					
θ_7	θ_8	θ_9	θ_{10}	θ_{11}	θ_{12}
0	-7.61	-65.15	53.73	-31.47	15.48

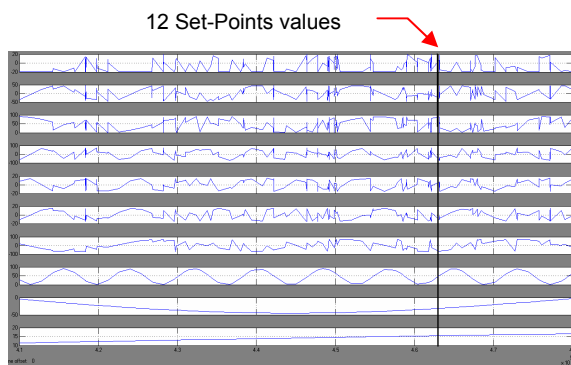


Figure 12: Sampled set-point signals.

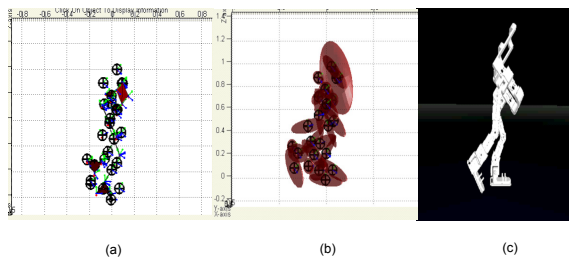


Figure 13: Example of the (a) stick diagram, (b) ellipsoidal mass-represented diagram and (c) 3D Virtual Reality Graphical Interpretation at following set-point.

4.4 One Step Gait Motion

The generation of all geometrically feasible postures of the exoskeleton is done using sine curves to characterize the changes in the joint trajectories by assuming that the left and right feet are supporting and swinging feet respectively. The CM equations as given in (1), (2) and (3) refer to well known zero moment point (ZMP) equations where all accelerations are equal to zero except $g \approx 9.81 \text{ m/s}^2$. With random sampling from all postures (6.1×10^9

postures) while arranging them from time 0 second to 6.1×10^9 seconds, the range of time (searching domain) that returns stable leg-swinging postures could be found from Figure 14 (a) between 4.18×10^9 - 5.91×10^9 seconds. From the sampled experiment, the authors could reduce size of the searching domain from 6.1×10^9 solutions to approximately 2×10^9 solutions. However, from the visual interpretation in VR environment, the postures that result from the solution numbered 4.7×10^9 to 5.91×10^9 show the waist orientation that would be difficult for the wearer of the exoskeleton to move along with the exoskeleton. Therefore, another detail simulation is performed to determine the CM-feasible postures (joint angles) within the searching domain 4.18×10^9 to 4.7×10^9 . The result is shown in Figure 14 (b).

$$z_{ZMP} = \frac{\sum_i m_i (\ddot{y} + g) z_i - \sum_i m_i \ddot{z} y_i - \sum_i I_{ix} \ddot{\theta}_{ix}}{\sum_i m_i (\ddot{y} + g)} \quad (1)$$

$$x_{ZMP} = \frac{\sum_i m_i (\ddot{y} + g) x_i - \sum_i m_i \ddot{x} y_i - \sum_i I_{iz} \ddot{\theta}_{iz}}{\sum_i m_i (\ddot{y} + g)} \quad (2)$$

$$y_{ZMP} = 0 \quad (3)$$

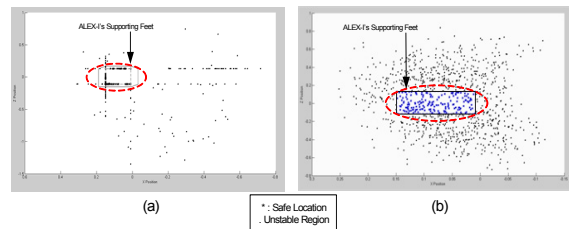


Figure 14: (a) Location of CM sampled over the entire searching domain, (b) Location of CM sampled over the reduced searching domain.

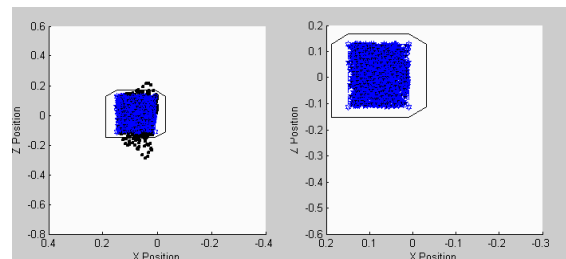


Figure 15: Filter and interpolated feasible posture CM-save joints.

Only the postures (joints angles) that return the balanced gait are saved into the database so that the

interpolation of all feasible joints angles could be interpolated. The filtered postures are again interpolated to obtain very detailed joint trajectories and filtered to get only the balanced CM joints angles. The resulting filtered CM locations are shown Figure 15.

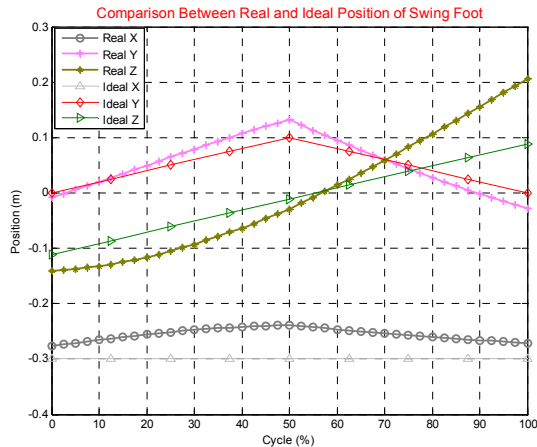


Figure 16: Comparison between ideal and CM-save gait pattern.

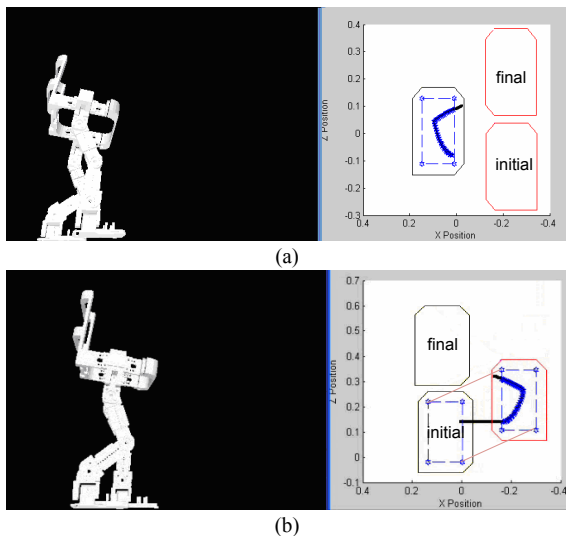


Figure 17: (a) Right-Swing step, (b) Left-Swing step.

With the feasible joint trajectories, the step parameters, which comprise the swinging height, step length, and step time, are obtained. After having all joints angles in the database together with the location of the swing foot and orientation of the ALEX-I from the virtual sensors in the simulation model, the one-step gait pattern is generated from the CM-feasible joint trajectories.

In Figure 16, the ideal walking pattern and balanced CM walking pattern are compared.

Apparently, the obtained step parameters could only be partially achieved since the ALEX-I has to balance itself and could not be in some particular postures. The successful right-swing and left-swing step are shown in the Figure 17 with the outlined initial and final locations of the swung feet.

5 CONCLUSIONS

This paper has revealed the balancing control analysis and design of the architecture layout of the ALEX-I. The ALEX-I was initially controlled to walk in open-loop manner. Position control for each joint is operated with 32-Bits processor ARM7 controller, which senses position feedback from 1024 pulses/rev encoder. PC104 is used as a main controller to control the entire joints controller and to calculate all the set-points for the gait motion of the ALEX-I. The ALEX-I simulation model has been verified with DH matrix Robotics Toolbox and the accurate results are observed. The model has been further integrated to perform gait motion analysis. The motion is captured in the form of 12 set-points observed with virtual sensors offered by SimMechanics library. The CM-feasible balance gait data are filtered and interpolated. One gait cycle has been shown in the simulation and in this study. With the obtained balanced gait motion, the data could be set and calculated by PC104. The future works would emphasize on the ZMP-feasible gait pattern generation, implementation with the real wearer, and disturbance-tolerating control system.

ACKNOWLEDGEMENTS

This research is financially supported by National Electronics and Computer Technology Center (NECTEC), Thailand.

REFERENCES

Aphiratsakun, N., Parnichkun, M., 2007. Preliminary Study and Design of an Exoskeleton. Proceeding of the CRIT 2007, Nakhon Pathom, Thailand.
 Chu, A., Kazerooni, H., Zoss, A., 2005. On the Biomimetic Design of the Berkeley Lower Extremity Exoskeleton (BLEEX). Proceeding of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, pp. 4345-4352.

- Corke, P. I., 1996. *A Robotics Toolbox for MATLAB*, IEEE Robotics and Automation Magazine, no.1, vol.3, pp.24-32.
- Craig, J. J., 2005. *Introduction to Robotics: Mechanics and Control*, Pearson Prentice Hall. 3rd edition.
- Guizzo, E., Goldstein, H., 2005. *The rise of Body Bots*. Spectrum, IEEE.
- Kawamoto, Sankai, Y., 2002. Comfortable Power Assist Control Method for Walking Aid by HAL-3. Proceeding of the 2002 IEEE/SMC International Conference.
- Kawamoto, H., Kanbe, S., Sankai, Y., 2003. Power Assist Method for HAL-3 Estimating Operator's Intention Based on Motion Information. Proceeding of the 2003 IEEE International Workshop on Robot and Human Interactive Communication, California, USA, pp. 67-72.
- Kazerooni, H., Racine, J. L., Huang, L., Steger R., 2005. On the Control of the Berkeley Lower Extremity Exoskeleton (BLEEX). Proceeding of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, pp. 4353-4360.
- Lee, S., Sankai, Y., 2002. Power Assist Control for Leg with HAL-3 Based on Virtual Torque and Impedance Adjustment. Proceedings of the 2002 IEEE/SMC International Conference.

SHOE GRINDING CELL USING VIRTUAL MECHANISM APPROACH

Bojan Nemec and Leon Zlajpah

Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

bojan.nemec@ijs.si, leon.zlajpah@ijs.si

Keywords: Robotics, Redundant robot control, Force control, Shoe grinding.

Abstract: The paper describes the automation of the shoe grinding process using an industrial robot. One of the major problems of flexible automation using industrial robots is how to avoid joint limitations, singular configuration and obstacles. This problem can be solved using kinematically redundant robots. Due to the circular shape of the grinding disc, the robot becomes kinematically redundant. This task redundancy was efficiently handled using virtual mechanism approach, where the tool is described as a serial mechanism.

1 INTRODUCTION

Shoe production is most likely labor intensive; the rate of automation is usually low. Therefore it is considered as the industry suitable for the counties with low labor cost (Taylor and Taylor, 1988; Nemec and et all, 2003). In last years new aspect in shoe production is arising - custom made shoes (Dulio and Boer, 2004). The customization in mass shoe production requires complex information system and full automation of planning, production and distribution processes. One of the major challenges is how to generate robot trajectories base solely on the CAD model of the shoe. Manual teaching and trajectory testing phases are not acceptable for customized shoes, where virtually each work-piece on the production line can differ from the previous one. Therefore, we have to design tools which enable to generate fault tolerant robot trajectories. This is usually accomplished using complex task planning algorithms, which are in most cases off-line batch procedures. In (Nemec and Zlajpah, 2008) we proposed a solution based on control algorithms for the kinematically redundant robot, where we sacrificed exact orientation of the tool in order to achieve additional degrees of redundancy. In this paper, we propose a new approach, called virtual mechanism approach. The proposed algorithm was applied to the shoe grinding cell, which uses an industrial robot with 7 D.O.F.

2 SOLVING TASK KINEMATIC REDUNDANCY USING VIRTUAL MECHANISM APPROACH

Robotic systems under study are n degrees of freedom (DOF) serial manipulators. We consider redundant systems, which have more DOF than needed to accomplish the task, i.e. the dimension of the joint space n exceeds the dimension of the task space m , $n > m$ and $r = n - m$ denote the degree of the redundancy. Let the configuration of the manipulator be represented by the a vector \mathbf{q}_r of n joint positions, and the end-effector position (and orientation) by m -dimensional vector \mathbf{x}_r of the robot tool center point positions (and orientations). The relation between the joints and the task velocities is given by the following well known expression

$$\dot{\mathbf{x}}_r = \mathbf{J}_r \dot{\mathbf{q}}_r \quad (1)$$

where \mathbf{J}_r is the $m \times n$ manipulator Jacobian matrix. The solution of the above equation for $\dot{\mathbf{q}}_r$ can be given as a sum of the particular and the homogeneous solution

$$\dot{\mathbf{q}}_r = \bar{\mathbf{J}}_r \dot{\mathbf{x}}_r + \mathbf{N}_r \xi \quad (2)$$

where

$$\bar{\mathbf{J}}_r = \mathbf{W}^{-1} \mathbf{J}_r^T (\mathbf{J}_r \mathbf{W}^{-1} \mathbf{J}_r^T)^{-1}. \quad (3)$$

Here, $\bar{\mathbf{J}}_r$ is the weighted generalized-inverse of \mathbf{J}_r , \mathbf{W} is the weighting matrix, $\mathbf{N}_r = (\mathbf{I} - \bar{\mathbf{J}}_r \mathbf{J}_r)$ is a $n \times n$ matrix representing the projection into the null space of \mathbf{J}_r , and ξ is an arbitrary n dimensional vector. We will denote this solution as the generalized inverse based

redundancy resolution at the velocity level (Nenchev, 1989). The homogenous part of the solution belongs to the Jacobian null-space. Therefore, we will denote it as $\hat{\mathbf{q}}_n$, $\hat{\mathbf{q}}_n = \mathbf{N}_r \xi$. Now consider the case where the robot Jacobian matrix \mathbf{J}_r is defined in Cartesian (world) coordinate system and the dimension of the Jacobian is $6 \times n$, but the task is described in another coordinate system. It can be shown that the transformation from the Cartesian to the task space can be very complex. As an alternative approach we propose to model the tool as a serial kinematic link. Let consider the general case where the robot holds the object to be machined and the work tool is fixed, as illustrated in Fig. 1. In such a case, we can define the direct kinematic transformation as

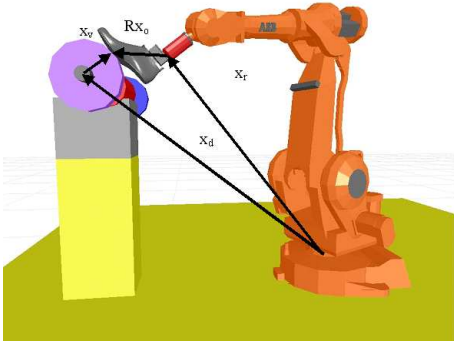


Figure 1: The robot holds the object and the work toll is fixed.

$$\mathbf{x}_r + [\mathbf{R} \ \mathbf{I}] \mathbf{x}_o = \mathbf{x}_d + \mathbf{x}_v \quad (4)$$

where \mathbf{x}_r is the robot Cartesian position and orientations, \mathbf{R} is the robot tool rotation 3×3 dimensional matrix, \mathbf{x}_o is the 6×1 vector of the object position and orientation, \mathbf{x}_v is the 6×1 vector of position and orientation of the top of the virtual mechanism and 6×1 vector \mathbf{x}_d describes the distance and orientation between the base coordinates system and the work tool coordinate system. Let consider robot and virtual mechanism as one mechanism with $n + n_v$ degrees of freedom. The configuration of the virtual mechanism can be described with the $n_v \times 1$ dimensional vector \mathbf{q}_v . The new task position is

$$\mathbf{x} = \mathbf{x}_r - \mathbf{x}_v \quad (5)$$

and the Jacobian of this new mechanism can be expressed as

$$\mathbf{J} = [\mathbf{J}_r \ | \ -\mathbf{J}_v] \quad (6)$$

where the Jacobian of the virtual mechanism \mathbf{J}_v is defined as

$$\mathbf{J}_v = \frac{\partial \mathbf{x}_v}{\partial \mathbf{q}_v} \quad (7)$$

As we can see, the task space preserves its dimension, while the joint space is increased with the dimension of the virtual mechanism. This approach has two major advantages. First, we can use the existing robot Jacobian, which is assumed to be known. Second, the augmented part of the Jacobian has very simple structure in most cases. Note that Eq. 4 does not handle orientations correctly, since orientation vectors can not be simply added in general case. Therefore, using Eq. 4 and 5 we obtain an approximate solution regarding the orientation vector. In most cases, the difference between the desired and the real orientation vector is small and is acceptable for operations like brushing and polishing. If orientations are important, we can use equation 4 and 5 for the calculation of positions only, while the orientations have to be calculated using rotation matrices as follows.

$$\mathbf{R}_o = \mathbf{R}_v^T \mathbf{R} \quad (8)$$

Here, \mathbf{R}_o and \mathbf{R}_v are 3×3 rotation matrices describing object rotation against virtual mechanism and virtual mechanism rotation expressed in the robot base coordinate system. The corresponding orientation vector can be then obtained using the transformation of the rotation matrix to the orientation vector described with euler or roll pitch yaw notation. Note that using this 'correct' transformation also rotation part of the Jacobian described by Eq. 6 becomes more complex. On the other hand, when the Jacobian is used in the control loop, only approximate values of Jacobian are needed. Therefore, control algorithm based on Jacobian defined by Eq. 6 gives equal results as control algorithm using the Jacobian calculated from rotation matrices, as described with Eq. 8.

3 CONTROL

Using the virtual mechanism approach, we can directly apply any control algorithm for the kinematically redundant robot. A suitable choice is the control law using generalized inverse-based redundancy resolution at velocity level in the extended operational space. Redundancy resolution at the velocity level is favorable because it enables direct implementation of the gradient optimization scheme for the secondary task. The gradient projection technique has been widely used for the calculation of the null space velocity that optimizes the given criteria. The reason for this is that a variety of performance criteria can be easily expressed as gradient function of joint coordinates. Although the control law using generalized inverse-based redundancy resolution at velocity level can not completely decouple the task and the

null space (Oh et al., 1998; Park et al., 2002; Nemeć and Zlajpah, 2000; Nemeć and Zlajpah, 2001), it enables good performance in real implementation (Nemeć et al., 2007). The joint space control law is

$$\tau_c = \mathbf{H}\mathbf{J}^T(\ddot{\mathbf{x}}_d + \mathbf{K}_v\dot{\mathbf{e}}_x + \mathbf{K}_p\mathbf{e}_x + \mathbf{K}_f\mathbf{e}_f - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{H}\mathbf{N}(\dot{\mathbf{q}}_{nd} + \mathbf{K}_n\dot{\mathbf{e}}_n - \dot{\mathbf{N}}\dot{\mathbf{q}}) + \mathbf{h} + \mathbf{J}^T\mathbf{f} \quad (9)$$

where \mathbf{J} is the Jacobian matrix, \mathbf{H} is $n \times n$ inertia matrix, \mathbf{h} is n -dimensional vector of the centrifugal, coriolis and gravity forces, \mathbf{f} is n -dimensional vector of the external forces acting on the manipulator's end effector and \mathbf{K}_p , \mathbf{K}_v , \mathbf{K}_f and \mathbf{K}_n are diagonal matrices representing positional, velocity, force and null-space feedback gains. The first term of the control law corresponds to the task-space control torque τ_x , the second to the null-space control torque τ_n and the third and the fourth is used to compensate the non-linear system dynamics and the external force, respectively. Here, $\mathbf{e}_x = \mathbf{x}_d - \mathbf{x}$ are the task-space tracking error, $\mathbf{e}_f = \mathbf{f}_d - \mathbf{f}$ and $\dot{\mathbf{e}}_n = \dot{\mathbf{q}}_{nd} - \dot{\mathbf{q}}_n$ is the null-space tracking error. \mathbf{x}_d , \mathbf{f}_d and $\dot{\mathbf{q}}_{nd}$ are the desired task coordinates, desired force and the desired null space velocity, respectively. The details of the control law derivation can be found in (Nemeć and Zlajpah, 2000; Nemeć et al., 2007).

An attention should be paid on the selection of the inertia of the virtual link. The inertia matrix \mathbf{H} has the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_v \end{bmatrix} \quad (10)$$

where \mathbf{H}_r is the robot inertia matrix and \mathbf{H}_v is the diagonal matrix describing the virtual mechanism inertia. Clearly, \mathbf{H}_v can not be zero, but arbitrary small values can be chosen describing the lightweight virtual mechanism. Selection of the inertia matrix of the virtual mechanism affects only the null space behavior of the whole system. Heavy virtual links with high inertia would slow down the movements of the virtual links. Therefore, low inertia of virtual links is an appropriate choice. On the contrary, we can assume that no gravity, coriolis and centrifugal forces act on the virtual links and the corresponding terms in the vector \mathbf{h} can be set to zero. Control law 9 assumes feedback from all joints, including non-existing virtual joints. There are multiple choices how to provide the joint coordinates and the joint velocities of the virtual link. A suitable method is to build a simple model composed of a double integrator

$$\begin{aligned} \dot{\mathbf{q}}_v &= \int \mathbf{H}_v^{-1}\tau_{cv} \\ \mathbf{q}_v &= \int \dot{\mathbf{q}}_v \end{aligned} \quad (11)$$

where τ_{cv} is the part of the control signals corresponding to the virtual link.

4 MOTION OPTIMIZATION

As we mentioned previously, one of the main problems in automatic trajectory generation is the inability to assure that the generated trajectory is feasible using a particular robot, either because of possible collisions with the environment or because of the limited workspace of the particular robot. Limitations in the workspace are usually not subjected to the tool position, but rather to the tool orientation. Another severe problem are wrist singularities, which can not be predicted in the trajectory design phase on a CAD system. A widely used solution in such cases is off-line programming with graphical simulation, where such situation can be detected in the design phase of the trajectory. Unfortunately this is a tedious and time consuming process and therefore not applicable in customized production, where almost each work piece can vary from the previous one (Dulio and Boer, 2004; Nemeć and Zlajpah, 2008). The problem can be efficiently solved using the null space motion, which changes the robot configuration, but does not affect the task space motion. The force and the position tracking are of the highest priority for a force controlled robot and are therefore considered as the primary task. The secondary task is defined by the optimization of a given cost function.

Let p be the desired cost function, which has to be maximized or minimized. Then the velocities

$$\dot{\mathbf{q}}_n = k\mathbf{N}\mathbf{H}^{-1}\left(\frac{\delta p}{\delta q_1}, \frac{\delta p}{\delta q_2}, \dots, \frac{\delta p}{\delta q_n}\right) \quad (12)$$

in Eq 12 maximize cost function for any $k > 0$ and minimize cost function for any $k < 0$ (Asada and Slotine, 1986), where k is an arbitrary scalar which defines the optimization step. In our case we have selected a compound p which maximizes the distances between obstacle and the robot links or robot work object, maximizes the distance to the singular configuration of the robot and maximizes the distance in joint coordinates between current joint angle and joint angle limit. We define the cost function as a sum of three cost functions $p = p_a + p_l + p_s$, where p_a denotes cost function for obstacle avoidance, p_l cost function for avoiding joint limits and p_s cost function for singularity avoidance. We select the cost function for obstacle avoidance as (Khatib, 1986; Khatib, 1987) $p_a = \frac{1}{2}\mathbf{E}d_0^2$, where \mathbf{E} is an $l \times l$ rotation matrix describing the direction of an artificial potential field pointing from the obstacle, l is the dimension of the position sub-space and d_0 is the shortest distance between the obstacle and the robot body. In our case the desired objective is fulfilled if the imaginary force is applied only on the robot joints. The cost function for the joint limits avoidance is defined as (Nemeć and

Zlajpah, 2008; Chaumette and Marchand, 2001)

$$p_l = \begin{bmatrix} (q_{max} - q)^2, |q_{max} - q|^2 < \varepsilon \\ 0 \\ (q_{min} - q)^2, |q_{min} - q|^2 < \varepsilon \end{bmatrix} \quad (13)$$

where ε is a positive constant defining the neighborhood of joint limits. For the singularity avoidance we use the manipulability index defined as (Asada and Slotine, 1986)

$$p_s = \sqrt{|\mathbf{J}\mathbf{J}^T|} \quad (14)$$

Then, the desired null space velocity for our task are calculated using

$$\dot{\mathbf{q}}_n = \mathbf{N}\mathbf{H}^{-1}(k_a\mathbf{J}^{03}Vd - 2k_l(\mathbf{q}_l - \mathbf{q}) - 2k_s\frac{\delta\mathbf{J}}{\delta\mathbf{q}}\mathbf{J}^T) \quad (15)$$

Matrix \mathbf{J}^{03} is the Jacobian matrix calculated from the robot base to the robot wrist. Scalars k_a, k_l and k_s are appropriate positive constants defining the optimization step. In real implementation, k_a and k_l are set to zero if the observed point is away enough from the possible collision points and joints are far away from their limits, respectively. Similar, the last term of $\dot{\mathbf{q}}_n$ is not computed if the manipulability index is large enough. Unfortunately, the partial derivative $\frac{\delta\mathbf{J}}{\delta\mathbf{q}}$ is not easy to calculate. However, we can use the numerical derivative of the manipulability measure p_s instead. Vector \mathbf{q}_l denotes the physical joint limit in the range $[q_{min}, q_{max}]$.

5 SHOE GRINDING

In the shoe assembly process, in order to attach the upper with the corresponding sole, it is necessary to remove a thin layer of the material off the upper surface so that the glue can penetrate the leather. To do this, the robot has to press the shoe against the grinding disc with the desired force while executing the desired trajectory. In the past, there were several approaches how to automate this operation. For mass production, there are special NC machines available. Their main drawback is relatively complicated setup and are therefore not suitable for the custom made shoes. Required flexibility is offered by the robot based grinding cell. In the EUROShoeE project (Dulio and Boer, 2004), a special force controlled grinding head has been designed. The robot manipulated with the grinding head while the shoe remained fixed on the conveyor belt (Jatta et al., 2004). The main drawback of this approach is relatively heavy and expensive grinding head. Additionally, force control can be applied only in one direction. In our approach, the robot holds the shoe and presses it against

the grinding disc of a standard grinding machine as used in the shoe production industry. The impedance force control was accomplished by the robot using universal force-torque sensor mounted between the robot wrist and the gripper which holds the shoe last. It is well known that the kinematic redundancy enables greater flexibility in execution of complex trajectories. For example, also humanoid hand dexterity is subjected by its kinematical redundancy. We used Mitsubishi Pa10 robot with 7 D.O.F in our roughing cell, which has one degree of redundancy. Additional two degrees of redundancy were obtained by treating the grinding disc as a virtual mechanism. The surface of the grinding disc can be naturally described with the outer surface of the torus, where R and r are the corresponding radius of the grinding disc, as shown in the Fig. 2. Let x be the task (Cartesian) coordinate of the whole system. Assuming that the robot tool position and robot Jacobian is known, the forward kinematics can be easily expressed as

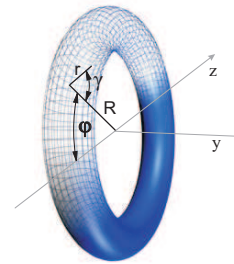


Figure 2: Rotary brush presented as torus.

$$\mathbf{x} = \mathbf{x}_r + \begin{bmatrix} s_\varphi(R + r c_\gamma) \\ r s_\psi \\ c_\varphi(R + r c_\gamma) \\ 0 \\ -\varphi \\ \psi \end{bmatrix}, \quad (16)$$

while the corresponding Jacobian is

$$\mathbf{J} = \begin{bmatrix} c_\varphi(R + r c_\psi) & -s_\varphi r s_\gamma \\ 0 & r c_\psi \\ -s_\varphi(R + r c_\psi) & -c_\varphi r s_\gamma \\ 0 & 0 \\ -1 & \\ 0 & 1 \end{bmatrix}. \quad (17)$$

Here, we used the abbreviation $c_\varphi = \cos(\varphi)$, $c_\gamma = \cos(\gamma)$, $s_\varphi = \sin(\varphi)$ and $s_\gamma = \sin(\gamma)$. Thus we have 9 degrees of freedom, 6 of them are required to describe the grinding task, while the remaining three degrees of freedom are used for the obstacle avoidance, joint limits avoidance and singularity avoidance.



Figure 3: Experimental cell for shoe bottom roughing.

The prototype of the cell is shown in figure 3. It consists of the Mitsubishi Pa10 robot with a force/torque sensor Jr3 mounted in the robot wrist, a grinding machine, a Pa10 robot controller and a cell control computer, which coordinates the task and calculates the required robot torques. The control computer is connected to the robot controller using Arc-Net. The frequency rate of the control algorithm (Eq. 9) and the motion optimization algorithm (Eq. 15) is 700 Hz. The grinding path is obtained from CAD model of the shoe. For this purpose, the control computer is connected to the shoe database computer using Ethernet. Unfortunately, CAD model itself can not supply all necessary data for the grinding process. CAD models are usually available for the reference shoe size, therefore, non-linear grading of the shoe shape is necessary for the given size. Additionally, some technological parameters such as material characteristics and shoe sole gluing technology have to be taken into account during the grinding trajectory preparation. For this purpose, we have developed a special CAD expert program, which enables the operator to define additional technological and material parameters. The program then automatically generates the grinding trajectory.

In order to show the efficiency of the proposed algorithm, we defined the shoe grinding trajectory as seen in the Fig 4. Note that without using trajectory optimization is is very hard to execute the given task without splitting the desired trajectory in two or more fragments. Fig 5 shows how the system rotated joints of virtual mechanism in order to avoid the joint limits and to minimize joint velocities of the robot and virtual mechanism.

Similar behavior could be obtained also by using the well known hybrid force-position control, where restricted coordinates are perpendicular to the grinding disc. However, the approach with hybrid force-position control has several disadvantages. First of

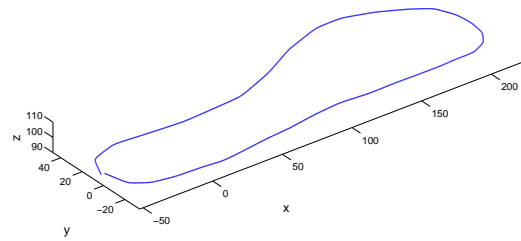


Figure 4: Shoe grinding trajectory.

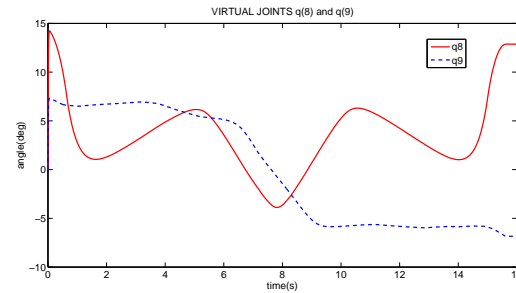


Figure 5: Virtual mechanism angles q_8 and q_9 .

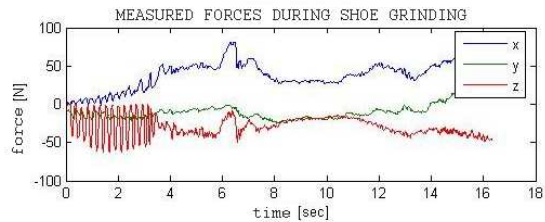


Figure 6: Resulting forces during the shoe grinding.

all, it requires perfect force tracking in order to maintain the tool always perpendicular to the grinding disc. Due to the irregularities of the shoe bottom and the grinding disc rotation, this is very hard to obtain. Results of the force tracking shown in Fig. 6 clearly demonstrates, that the resulting force tracking is still imperfect despite of the high sampling frequency of the control algorithm. Note that only the force in z direction was controlled in this case. On the contrary, our approach does not require force control to follow the shape of the grinding disc at all. Therefore, we were able to apply the impedance control law, which is more appropriate for the applications such as grinding and polishing.

6 CONCLUSIONS

In the paper we presented a cell for shoe grinding operation. We proposed a new method of solving the kinematic redundancy which arises from the shape

of the work tool. The main benefit of the proposed method is the simplicity and efficiency. It can be used on the existing robot controllers with very moderate changes of the control algorithm. The proposed method can be applied even in the case of moving obstacles during the task execution, assuming that the shape and position of the obstacle is known. Another, perhaps for the practical implementation even more attractive possibility is to use the proposed approach in the trajectory generation and not in the control algorithm. In this case we can use existing standard industrial robots and we benefit from the advantages of the kinematic redundancy due to circular shape of the tool. This latter approach was successfully implemented in the cell for custom finishing operations in shoe assembly (Nemec and Zlajpah, 2008). It was also demonstrated that the proposed control has many advantages when compared with the hybrid force-position control, which tracks an unknown shape using only the force sensor data.

REFERENCES

- Asada, H. and Slotine, J.-J. (1986). *Robot Analysis and Control*. John Wiley & Sons.
- Chaumette, F. and Marchand, . (2001). A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing. In *IEEE Transactions on Robotics and Automation*, 17(5).
- Dulio, S. and Boer, S. (2004). Integrated production plant (ipp): an innovative laboratory for research projects in the footwear field. In *Int. Journal of Computer Integrated Manufacturing*, 17(7) : 601-611.
- Jatta, F., Zanon, L., Fassi, I., and Negri, S. (2004). A roughing/cementing robotic cell for custom made shoe manufacture. In *Int. J. Computer Intergrated Manufacturing*, 17(7) : 645-652.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. In *Int. J. of Robotic Research*, 5 : 90 – 98.
- Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: the operational space formulation. In *IEEE Trans. on Robotics and Automation*, 3(1) : 43 – 53.
- Nemec, B. and et al (2003). *Technology fostering individual, organisational, and regional development: an international perspective*.
- Nemec, B. and Zlajpah, L. (2000). Null velocity control with dynamically consistent pseudo-inverse. In *Robotica*, 18 : 513 – 518.
- Nemec, B. and Zlajpah, L. (2001). Experiments with force control of redundant robots in unstructured environment using minimal null-space formulation. In *Journal of Advanced Computational Intelligence*, 5(5) : 263 – 268.
- Nemec, B. and Zlajpah, L. (2008). Robotic cell for custom finishing operations. In *Int. J. Computer Intergrated Manufacturing*, 21(1) : 33-42.
- Nemec, B., Zlajpah, L., and Omrcen, D. (2007). Comparison of null-space and minimal null-space control algorithms. In *Robotica*, 2007, 25(5):511-520.
- Nenchev, D. N. (1989). Redundancy resolution through local optimization: A review. In *J. of Robotic Systems*, 6(6) : 769 – 798.
- Oh, Y., Chung, W., Youm, Y., and Suh, I. (1998). Experiments on extended impedance control of redundant manipulator. In *Proc. IEEE/RJS Int. Conf. on Intelligent Robots and Systems*, : 1320 – 1325, Victoria.
- Park, J., Chung, W., and Youm, Y. (2002). Characterization of instability of dynamic control for kinematically redundant manipulators. In *Proc. IEEE Conf. Robotics and Automation*, : 2400 – 2405, Washington DC.
- Taylor, P. and Taylor, G. (1988). *Garments and Shoe Industry Robots*.

REAL TIME GRASPING OF FREELY PLACED CYLINDRICAL OBJECTS

Mario Richtsfeld, Wolfgang Ponweiser and Markus Vincze
Institute of Automation and Control, Vienna University of Technology
Gusshausstr. 27-29, Vienna, Austria
{rm, wp, vm}@acin.tuwien.ac.at

Keywords: Service robotics, laser-range scanning, object detection, task planning.

Abstract: In the near future, service robots will support people with different handicaps to improve the quality of their life. One of the required key technologies is to setup the grasping ability of the robot. This includes an autonomous object detection and grasp motion planning to fulfil the task of providing objects from any position on a table to the user. This paper presents a complete system, which consists of a fixed working station equipped with a laser-range scanner, a seven degrees of freedom arm manipulator and an arm prothesis as gripper. The contribution of this work is to use only one sensor system based on a laser-range scanning head to solve this challenge. The goal is that the user can select any defined object on the table and the robot arm delivers it to a target position or to the disabled person.

1 INTRODUCTION

At the beginning of the 1970's the development of service and rehabilitation robots started to support disabled people in their daily life. The goal is to make them more independent. Today we differ between fixed systems, in which an industrial robot is mounted on a working station and mobile systems, e.g. wheelchair mounted manipulators, like MANUS (Mokhtari, 2001) or FRIEND-I (Martens, 2001) and FRIEND-II (Ivlev, 2005). Popular fixed systems are e.g. DeVar (Van der Loos, 1995), ProVar (Van der Loos, 1999), RAID (Eftring, 1994), MASTER-RAID (Dallaway, 1995) or CAPDI (Casals, 1999).

Our vision is a fully autonomous mobile robot, which is able to detect, grasp and manipulate any kind of object. One of the key challenges of this work is the robust perception of objects. This challenge is analyzed by a fixed setup consisting of a laser-range scanner and a robot arm. We use an AMTEC¹ robot arm with seven degrees of freedom, which is used for object grasping and manipulation. The joint setup is assembled similar to a human arm. The robot arm is equipped with a hand prothesis from the company Otto Bock², which we are using as gripper. It is

thought that elderly persons will accept this type of gripper more easily than an industrial gripper, due to the form and the optical characteristics.

The outline of the paper is as follows: In the next section the state of the art of grasp robot systems, grasping technology and object perception based on structure in 2-d and 3-d is presented. Section 3 introduces our robotic system and its components. Section 4 describes the object identification to calculate the object position and Section 5 details the grasping and manipulation. Section 6 gives some experimental results during a live demo presentation and Section 7 finally concludes the paper.

2 STATE OF THE ART

In the early 1970's one of the first wheelchair mounted manipulator was developed at the V.A. Rehabilitation Engineering (formerly Prosthetics) center (Prior, 1993). From 1983 to 1988 the mobile manipulator MoVAR (Van der Loos, 1995) was developed. This PUMA-250 robot was instrumented with a camera for remote sensing, a six-axis force sensor and a gripper with finger pad-mounted proximity sensors. A nice overview of different systems, such as the Wolfson-Robot and the Wessex-Robot is given by Hagen and Hillmann (Hagan, 1997). Up

¹<http://www.amtec-robotics.com>

²<http://www.ottobock.de/>

to now a number of scientists have been working on the same idea to develop a wheelchair mounted robot or a mobile robot system with arms to handle objects and assist elderly and handicapped persons, e.g. (Martens, 2001), (Volosyak, 2005). In the FRIEND systems (Martens, 2001), (Ivlev, 2005) the robot arm is controlled by a PC, which is fixed on the backside of the wheelchair. Both systems use a stereo camera system for object detection. The user interaction is based on a LC-display. The object must be placed on a predefined position on a tray, mounted at the front side of the wheelchair. A successful execution of the grasping task in this system is only possible for similar types of objects. Additionally they developed a "smart tray" that is used in combination with the vision sensors. This "smart tray" measures the weight and the position of objects with a matrix foil position sensor.

In comparison to the FRIEND systems, Saxena et al. (Saxena, 2006) developed a learning algorithm that predicts the grasp position of novel objects as a function of 2-d images, without building an explicit 3-d model of the object. This algorithm is trained via supervised learning using synthetic images for the training set. The work focuses on the task of identifying grasping positions without taking any complex manipulation tasks into account. A similar system describes Miller et al. (Miller, 2003). Their work specifies an automatic grasp planning system for hand configurations using shape primitives. By modeling an object as a sphere, cylinder, cone or box. They also use a set of rules to generate grasp positions.

In our case the vision task is to detect edges of objects that indicate grasp points. Accurate 3-d data is achieved by direct depth measurements, like laser-range scanning. In the range images, grasp points are indicated by object edges and grasp surface patches. Wang et al. (Wang, 2005) developed a general framework of automatic grasping of unknown objects by incorporating a laser scanner and a simulation environment. Their algorithms need a lot of time to detect grasp points. To aid industrial bin picking tasks Boughorbel et al. (Boughorbel, 2007) developed a system that provides accurate 3-d models of parts and objects in the bin to realize precise grasping operations. Due to their superquadrics based object modelling approach only rotation-symmetric objects can be used. To that effect Biegelbauer et al. describes a new approach of a hierarchical RANSAC search to obtain fast detection results of objects, which are modeled using approximated Superquadrics (Biegelbauer, 2007).

One of the most fundamental techniques for edge detection in range images is the scan line approxima-

tion (Jiang, 1999). It is well known and more efficient than the standard Canny (Canny, 1986) algorithm. The raw data points are approximated by a set of bivariate polynomial functions, in which the discontinuity of the fitted functions indicate the edge position. Katsoulas (Katsoulas, 2004) proposed an improved scan line approach by using an additional statistical merging step for a better handling of outliers. Based on these techniques we developed a 3-d edge detection method that enables a faster cylinder fit in 3-d range data.

3 SYSTEM APPROACH

The goal is, that the user can select any object on a table and the robot arm delivers it to a defined position or to the disabled person. The main challenges to solve are the robust detection of edges and their interpretation as grasping points. Our approach is based on scanning the objects by a rotating laser-range scanner and execution of subsequent path planning and grasping motion. Hence the system consists of a pan/tilt-mounted red-light laser and scanning camera and a seven degrees of freedom robot arm, which is equipped with a human like prosthesis hand (see Fig. 1).



Figure 1: Overview of the system components and their interrelations.

3.1 Laser-Range Scanner

The laser-range scanner records a snap-shot of the object scene with the help of a pan/tilt-unit. At present, it is mounted on a table. We are working to miniaturize the laser-range scanner to mount it on the shoulder of the robot later. A high resolution sensor is needed in order to detect a reasonable number of

edge-points of the objects with the required accuracy. The laser-range scanner used for this work consists of a red-light LASIRIS laser from StockerYale³ with 635nm and a MAPP2500 CCD-camera from SICK-IVP⁴ mounted on a pan/tilt-unit (PowerCube Wrist from AMTEC robotics). With the help of a cylinder lens, the laser-light is expanded and moves horizontally over the scene of interest. The camera grabs the laser-light profiles and extracts the laser-lines with the integrated microprocessor. The 3-d data is transformed to the world coordinate system. Finally the result can be displayed as a point cloud (see Fig. 2).

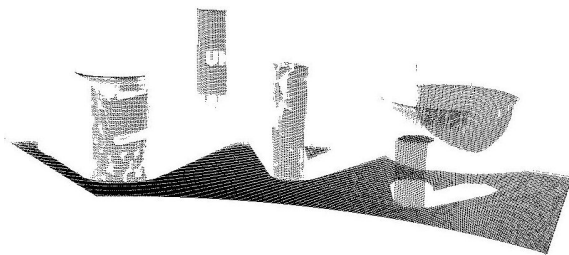


Figure 2: Exposure of the raw point cloud with 75.863 voxel. The two shadows from laser and camera are clearly visible.

3.2 Robot Arm and Gripper

For this work we use the "Light Weight Arm 7 DOF" from AMTEC robotics and a hand prothesis from Otto Bock as gripper. The robot arm exhibits seven degrees of freedom with a joint configuration similarly to the human arm (shoulder, elbow and wrist). The seventh degree of freedom is required to enable complex object grasping and manipulation and allow for some flexibility to avoid obstacles. The prothesis as end effector is selected due to the integrated force sensors as well as its increased acceptance of elderly and handicapped persons. It has three active fingers, the thumb, the index finger and the middle finger. The last two fingers are just for cosmetic reasons. Since, they have no active function in the grasping process their uncontrollable behavior must be considered, which reduces the grasping radius (see Fig. 1). As a huge advantage the integrated tactile sensors are used to detect a potential sliding of objects, which initializes a readjustment of the fingers.

3.3 Operation Sequence

The first step is to scan the scene on the table by the laser-range scanner. The camera converts the laser-

profiles to a 3-d point cloud, which can be visualized. Now the user can select the desired object. The developed algorithm analyzes the point cloud and calculates the position of the searched object. A commercial path planning tool from AMROSE⁵ calculates the trajectory to grasp the object. Before the robot arm delivers the object, the user can check the calculated trajectory in a simulation sequence. Then the robot arm executes the off-line programmed trajectory. The algorithm is implemented in C++. For displaying the results the Visualization Tool Kit (VTK)⁶ is used.

4 OBJECT IDENTIFICATION

The main goal of our work is to robustly detect cylindrical objects in the recorded point cloud in real time. Robustness includes the positive detection of defined objects despite any noise and outliers in a point cloud, which can be caused by specular surfaces (see Fig. 2, edges of the objects). To reduce complexity we only consider cylindrical objects for object detection for this work. An additional challenge is the complex interaction between the different operation parts. Finally to keep the standby time acceptable for the user the complete operating cycle should be finished within 20sec.. This time limit is challenging since usual object detection starts with an exhaustive segmentation step. As an example, object segmentation alone by recursive flood-filling with region-octree (Burger, 2007) of the desired table scene takes more than 30sec. (see Fig. 3). Thus a faster solution must be found. One alternative, which we exploited in our work is based on well investigated curvatures. Fig. 4 presents the steps of the fast object detection

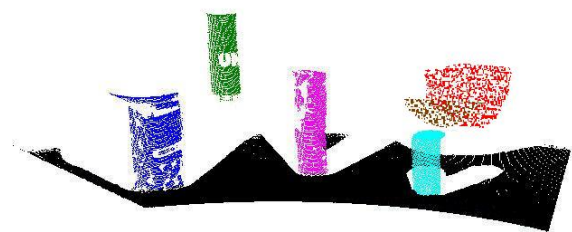


Figure 3: Segmentation of the different objects by recursive flood-filling. Images are best viewed in color.

method. In the first step, the "raw data preprocessing and vector estimation" the raw data points are pre-processed with a low pass filter to reduce any noise.

⁵<http://www.amrose.dk/>

⁶Freely available open source software, <http://public.kitware.com/vtk>.

³<http://www.stockeryale.com/index.htm>

⁴<http://www.sickivp.se/sickivp/de.html>

One of the most time consuming calculations is the normal vector estimation based on the orientation of the local neighborhood of 20mm, for what a region-occtree is used. These vectors are required to compute the axis of the cylinder objects. A lot of tests have shown that for a neighborhood of 20mm a reasonable accuracy can be achieved, while the calculation time stays acceptable. The "range image segmenta-

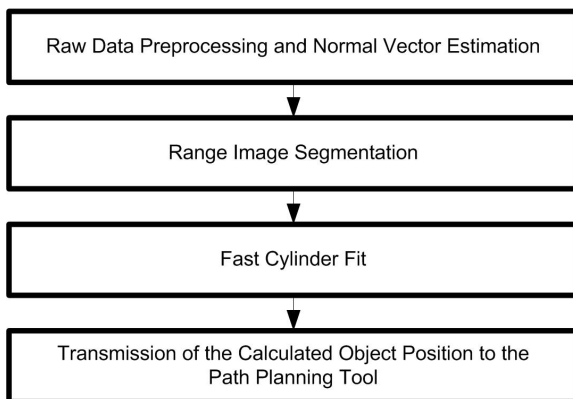


Figure 4: Flow chart of the object detection approach.

tion" starts by detecting the surface of the table with a RANSAC (Fischler, 1981) based plane fit. Then we analyze the curvature of the remaining points to filter neighbouring voxels with an angle difference between $\pm 78^\circ$ and $\pm 90^\circ$ (see Fig. 5).

The "fast cylinder fit" starts with a RANSAC based circle fit. Randomly three high curvature points are picked. The resulting circle is extended to a potential cylinder along its circumscribed axis down to the table. For every vicinity point, within a defined distance of 2mm of the calculated cylinder barrel, the normal distance to the cylinder barrel is calculated. The trial with the lowest mean of these distances is selected as cylinder (see Fig. 6). For comparison Jiang et al. (Jiang, 2005) published a method for 3-d circle

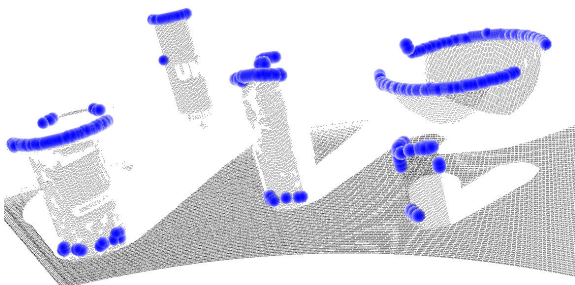


Figure 5: The acquired range image of the current table scene. The points with a high curvature are marked with blue dots. Images are best viewed in color.

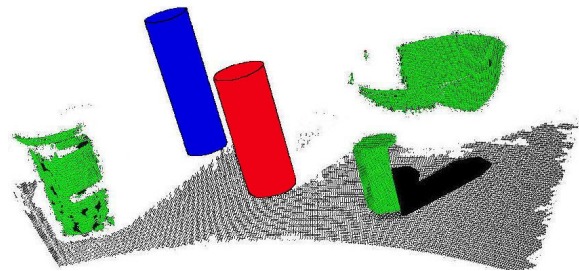


Figure 6: Detected objects in the table scene (blue cylinder - spray-on glue, red cylinder - beverage can, green points - rigid obstacles). Images are best viewed in color.

fitting. They reduce the number of local minima, but the error function is no more Euclidian. Here another simple proposal with an Euclidian function is used. For an explicit description, the raw data points of a profile scan are defined as (x_i, y_i, z_i) , n is the number of voxels and (x_a, y_a, z_a) is the circle's center. The resulting error function e is:

$$e = \sum_{i=1}^n \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2 + (z_i - z_a)^2} - r \quad (1)$$

The error must be smaller or equal than a defined threshold. In our case we use a distance of 2mm:

$$|e| \leq 2mm \quad (2)$$

In the last step of Fig. 4 "Transmission of the Calculated Object Position to the Path Planning Tool" the calculated object position in the actual environment model for collision avoidance has to be transmitted to the path planning tool. This 3-d mesh is generated by using all objects besides the target object, based on the triangles calculated by a DeLaunay triangulation (O'Rourke, 1998) This step is important to enable a collision free robot trajectory.

5 OBJECT GRASPING AND MANIPULATION

The task of this part of our work is to calculate a collision free robot path and to execute the grasping activity safely. The first step is performed by the path planning tool from AMROSE. The input to this tool is the detected object pose, the environment model and a transformation between the robot coordination system and the range scanner coordinate system. The output is a collision free trajectory to the desired object. Before the robot execution is approved, the user can check a simulation of the calculated trajectory and decide, if it is safe enough to handle the object (see Fig. 7 and Fig. 8). After the robot approaches the user

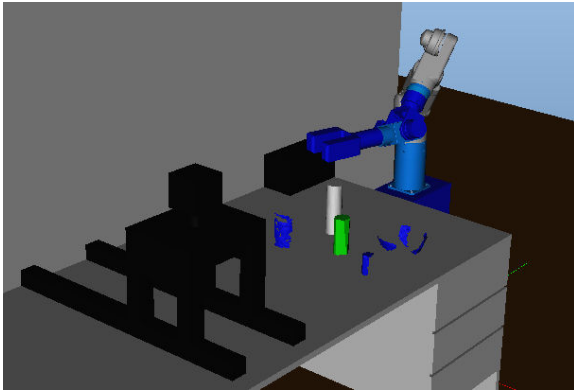


Figure 7: Visualization of the trajectory by a simulation tool. The white cylinder is the grasping object. The green cylinder (= 2.nd grasping object) and the blue objects are the obstacles. Images are best viewed in color.



Figure 8: Real position of the robot arm after the approach trajectory.

can initiate the closing of the gripper. As soon as the gripper encloses the object, the robot motion to the transfer point starts. Finally the desired object can be placed at a defined position or directly handed over to the user.

The calculation of the object detection and localization is performed by a PC with 1.8GHz Pentium IV processor and takes less than 12sec. depending on the range image size. The reliability depends on the ambient light, object surface properties, laser beam reflections and vibrations. Therefore, the laser-range scanner must be configured to the respective environment. By using an additional red-light filter the impact of light or reflections can be minimized.

6 EXPERIMENTS

The entire system exhibits its robust behavior and has been evaluated at a live demo presentation⁷ in front of more than 1000 college students. During the demonstration day about 50 runs were performed. The main problem that rarely appeared was a malfunction of the path planning tool, because no suitable trajectory could be found. Whereby the path planning had to be restarted. Sometimes the last two fingers, which reduce the grasping radius (see Fig. 1), shift the grasp object, but without a final effect on the success of the grasping process. Tab. 1 shows a short analysis of the arisen problems within 50 runs. The recognition of the cylindrical objects fails at strong environmental influences by the ambient light.

The autonomous grasping function should be able to find and grasp a cylindrical object in a defined area. When objects are positioned closer to each other, the autonomous grasping function show up difficulties to find the correct object. A minimum distance of 20mm (this distance is equal to the diameter of the thumb of the hand prosthesis) has to be observed between the objects.

Table 1: Evaluation of the arisen problems in percent [%] at 50 runs.

Arison Problems	Number of Events	Percent [%]
Path Planning	11	22%
Hand Prosthesis	4	8%
Object Recognition	2	4%
Sum	17	34%

7 CONCLUSIONS AND FUTURE WORK

This paper presents an approach of a robot system equipped with a laser-range scanner to get high accuracy table scene sensing. It shows that feature detection, in our case we only consider cylindrical objects, is a faster way (12sec.), than usual object segmentation (more than 30sec.) by a flood-filling recursive function. The presented method performs with very high reliability. Thus the approach for object detection and localization is well suited for use in related applications under difficult conditions.

A seven degrees of freedom arm manipulator and an arm prosthesis as gripper are used to grasp and deliver the desired object. The goal of this system is

⁷<http://www.yo-tech.at/>

to analyze the feasibility and reliability of object detection, which could be shown at a live demo. The cylinder detection approach can be extended to detect any type of object, since it is based on a grouping of high curvature points. This grasping approach can be applied for any kind of geometrical figures. This will expand the application to other tasks.

In the future, the robot arm will be installed on a mobile robot and for the object detection we calculate the grasping points of novel objects. This includes the revision of the path planning tool and a segmentation of sharp curvature points to speed up the method. Summarizing, this work illustrates that the concept of a 3-d vision guided robot arm can be adopted to many applications and has high potential to enable a more complex system. We will also deal with the development and the prototypes integration of a new laser range sensor with additional two cameras for stereovision to increase the robustness and predictability of the object detection system.

REFERENCES

- Biegelbauer, G.; Vincze, M. (2007). Efficient 3d object detection by fitting superquadrics to range image data for robot's object manipulation. In *International Conference on Robotics and Automation / ICRA*, pages 1086–1091. IEEE Press.
- Boughorbel, F.; Zhang, Y. (2007). Laser ranging and video imaging for bin picking. In *Assembly Automation*, volume 23, pages 53–59.
- Burger, W.; Burge, M. (2007). *Digital Image Processing - An Algorithmic Introduction Using Java*. Springer, UK, London, 1st edition.
- Canny, J. (1986). A computational approach to edge detection. In *Transactions on Pattern Analysis and Machine Intelligence*, volume 8, pages 679–698. IEEE Press.
- Casals, A.; Merchan, R. (1999). Capdi: A robotized kitchen for the disabled and elderly people. In *Proceedings of the 5th European Conference for the Advancement Assistive Technology / AAATE*, pages 346–351.
- Dallaway, J.L.; Jackson, R. (1995). Rehabilitation robotics in europe. In *Transactions on Rehabilitation Engineering*, pages 33–45. IEEE Press.
- Eftring, H. (1994). Robot control methods and results from user trials on the raid workstation. In *Proceedings of 4th Int. Conf. on Rehabilitation Robotics / ICORR*, pages 97–101.
- Fischler, M.A.; Boles, R. (1981). Random samples and consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, pages 381–395.
- Hagan, K.; Hillman, M. (1997). The design of a wheelchair mounted robot. In *Colloquium on Computers in the Service of Mankind: Helping the Disabled*, pages 1–6.
- Ivlev, O.; Martens, C. (2005). Rehabilitation robots friend-i and friend-ii with the dexterous lightweight manipulator. volume 17, pages 111–123. IOS Press.
- Jiang, X.; Cheng, D.-C. (2005). Fitting of 3d circles and ellipses using a parameter decomposition approach. In *5th International Conference on 3cartography Imaging and Modeling / 3DIM*, pages 103–109. IEEE Press.
- Jiang, X.; Bunke, H. (1999). Edge detection in range images based on scan line approximation. In *Computer Vision and Image Understanding*, volume 73, pages 183–199.
- Katsoulas, D.; Werber, A. (2004). Edge detection in range images of piled box-like objects. In *Conference on Pattern Recognition*, volume 2, pages 80–84.
- Martens, C.; Ruchel, N. (2001). A friend for assisting handicapped people. In *Robotics and Automation Magazine*, volume 8, pages 57–65. IEEE Press.
- Miller, A.T.; Knoop, S. (2003). Automatic grasp planning using shape primitives. In *International Conference on Robotics and Automation / ICRA*, volume 2, pages 1824–1829. IEEE Press.
- Mokhtari, M.; Abdurazak, B. (2001). Assistive technology for the disabled people: Should it work? the french approach. In *International Journal of Assistive Robotics and Mechatronics*, volume 2, pages 26–32.
- O'Rourke, J. (1998). *Computational geometry in C*. Univ. Press, Cambridge, 2nd edition.
- Prior, S.D.; Warner, P. (1993). Wheelchair-mounted robots for the home environment. In *Conference on Intelligent Robots and Systems / IROS*, volume 2, pages 1194–1200. IEEE Press.
- Saxena, A.; Driemeyer, J. (2006). Learning to grasp novel objects using vision. In *RSS Workshop on Manipulation for Human Environments*.
- Van der Loos, H. (1995). Va/stanford rehabilitation robotics research and development program: Lessons learned in the application of robotics technology to field of rehabilitation. In *Transactions on Rehabilitation Engineering*, volume 3, pages 46–55. IEEE Press.
- Van der Loos, H.; Wagner, J. (1999). Provar assistive robot system architecture. In *International Conference on Robotics and Automation / ICRA*, volume 1, pages 741–746. IEEE Press.
- Volosyak, I.; Ivlev, O. (2005). Rehabilitation robot friend-ii - the general concept and current implementation. In *9th International Conference on Rehabilitation Robotics / ICORR*, pages 540–544.
- Wang, B.; Jiang, L. (2005). Grasping unknown objects based on 3d model reconstruction. In *Proceedings of International Conference on Advanced Intelligent Mechatronics / ASME*, pages 461–466. IEEE Press.

MODIFIED LOCAL NAVIGATION STRATEGY FOR UNKNOWN ENVIRONMENT EXPLORATION

Safaa Amin, Andry Tanoto, Ulf Witkowski, Ulrich Rückert
System and Circuit Technology, Heinz Nixdorf Institute, Paderborn University
Fuerstenallee 11, D-33102 Paderborn, Germany
{amin, tanoto, witkowski, rueckert}@hni.upb.de

Saied Abdel-Wahab
Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt
mswahab@gmail.com

Keywords: Exploration, Mini-robot Khepera, Teleworkbench.

Abstract: This paper presents an algorithm for unknown environment exploration based on the local navigation algorithm (LNA). The original LNA doesn't take into account the case in which the robots are trapped and stop exploring the environment. In this paper, we propose some modifications to overcome this problem and demonstrate it by using real robots. For validation purpose, we ran several experiments using the mini-robot Khepera II running on the Teleworkbench. The complete environment is divided into small quadratic patches with some objects placed in it representing obstacles. With on-board infrared sensors and wheel encoder, the robot can successfully explore the unknown environment. Moreover, by calculating the distance to surrounding patches, the implemented algorithm will minimize the distance traveled, and in turn of the consumed energy and time. This paper also shows the advantage of using the Teleworkbench for performing experiments using real robots.

1 INTRODUCTION

Exploration of unknown environments is one of important problems in robotics. The goal of the exploration task is to cover the whole environment in a minimum amount of time or with minimum consumed energy depending on the application. Exploration approaches focus on guiding the robot efficiently through the environment in order to build a map. Exploration algorithms using either a single or a multi-robot system based on simulations have been studied extensively in the past (Stachniss, 2006, Simmons, 2000, Manolov, 2003, and Burgard, 2000). In this paper we present the result of the implementation of local navigation algorithm (LNA) for environment exploration as introduced in (Manolov, 2003, Amin, 2007). We also modify the LNA to solve the problem of trapped robot so that it can explore the whole environment successfully independent of its shape and the position of the obstacles. Moreover, we use the distance of the neighbouring patches relative to the current robot position to further improve the algorithm. We also

demonstrate the implementation of the modified algorithm using the mini-robot Khepera II. Furthermore, we demonstrate the advantage of using the Teleworkbench (Tanoto, 2005) as a test bed for performing and analysing experiments with real robots. For experiment analysis, we have developed a graphical analysis tool based on the MPEG-4 video standard (Tanoto, 2006). This tool allows us to record a video of the experiments together with experimental data and to visualize the internal and external behaviour of robots.

The paper is organized as follows: after presenting related work, Section 3 describes the LNA for unknown environment exploration and its limitation. Our modified algorithm (MLNA) is presented in Section 4. After that we present a comparison result between the two algorithms using the mini-robot Khepera II on the Teleworkbench. Finally, Section 6 concludes the paper.

2 RELATED WORK

Exploration is the task of guiding a vehicle in such a way that it covers the environment with its sensors. Efficient exploration strategies are also relevant for surface inspection, mine sweeping, or surveillance (Choset, 2001). In the past, several strategies for exploration have been developed. One group of approaches deals with the problem of simultaneous localization and mapping (Bourgault, 2002). A common technique for exploration strategies is to extract frontiers between known and unknown areas (Edlinger, 1994, Yamauchi, 1999) and to visit the nearest unexplored place. These approaches only distinguish between scanned and un-scanned areas and do not take into account the actual information gathered at each view-point. To overcome this limitation (Gonzales, 2001) determine the amount of unseen area that might be visible to the robot from possible view-points. To incorporate the uncertainty of the robot about the state of the environment (Moorehead, 2001) and (Bourgault, 2002) use occupancy grids (Hans, 1985) and compute the entropy of each cell in the grid to determine the utility of scanning from a certain location. (Whaite, 1997) present an approach that also uses the entropy to measure the uncertainty in the geometric structure of objects that are scanned with a laser range sensor. In contrast to the work described here they use a parametric representation of the objects to be scanned. (Edlinger, 1994) developed a hierarchical exploration strategy for office environments. Their approach first explores rooms and then traverses through doorways to explore other parts of the environment. (Tailor, 1993) describe a system for visiting all landmarks in the environment of the robot. Their robot maintains a list of unvisited landmarks that are approached and mapped by the robot. (Dudek, 1991) propose a strategy for exploring an unknown graph-like environment. Their algorithm does not consider distance metrics and is designed for robots with very limited perceptual capabilities. Recently, Koenig has shown, that a strategy, which guides the vehicle to the closest point that has not been covered yet, keeps the travelled distance reasonably small (Koenig, 2001). However, as experiments reported in this paper illustrate, such techniques can lead to a serious increase of measurements necessary to build an accurate map if the robot is not able to incorporate measurements on-the-fly while it is moving. This might be the case, for example, for robots extracting distance information from camera images.

3 LOCAL NAVIGATION STRATEGY FOR ENVIRONMENT EXPLORATION

The exploration strategy has to ensure that the complete area is explored. The LNA computes only the next step for moving. The computation is dependent on the area around the robot (Manolov, 2003).

The exploration algorithm works as follows. The complete environment is divided into small quadratic patches. The robot starts the exploration from any position in the environment. It can move between patches in all directions (east, west, north, south, and diagonal). When the robot visits a certain patch, it is considered to be analyzed. For the computation of the next movement, an algorithm is used to determine the costs of reaching each free patch around the robot. The cost function C for a free patch P is given as:

$$C(P) = N(P) \tag{1}$$

Where $N(P)$ is a function that computes the number of free neighbouring patches around patch P . A visualization of the evaluation is given in Figure 1. After evaluating the cost for all free patches around the robot, the robot moves to the patch with the lowest cost that has the lowest number of neighbours and which is therefore most unlikely to be reached again in the future (Manolov, 2003).

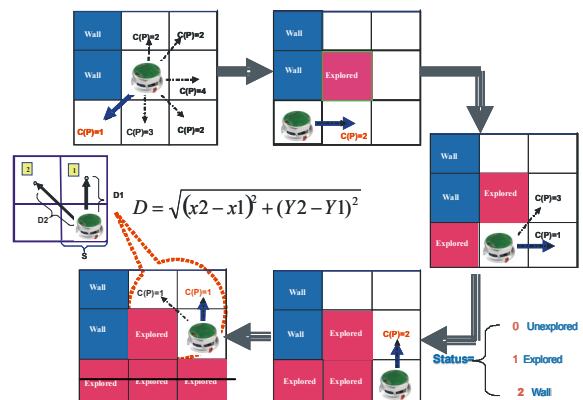


Figure 1: The algorithm determines for each free patch around the robot the costs C to reach it. The costs of all neighboring patches are different. Two patches have the same cost. The distance between the robot and the neighboring patches which have the same minimum costs (patch 1 and patch 2).

However, it is possible that there exist more than one patch with the same minimum cost. To solve this problem, we improve the algorithm by taking also into consideration the distance between the robot and each one of these patches, and then select the patch with minimum distance. Figure 1 illustrates the aforementioned situation.

4 THE MODIFIED LOCAL NAVIGATION ALGORITHM (MLNA)

LNA has a draw back that it can't ensure completeness in the case of a robot being trapped, e.g. when all of its neighbors are either obstacles or explored, and there is no free cell around to compute its cost. In this situation, it will simply stop and fails to complete exploring the whole environment as illustrated in Figure 2a.

We modified the algorithm to overcome this problem by calculating the shortest path to reach the unexplored area and continue exploring the environment. Our modified algorithm will determine the cost for reaching all the un-explored cells. We use the occupancy grid map in our algorithm to describe the environment. As illustrated in Figure 2b a cell can be in one of the following states which are represented by an integer number:

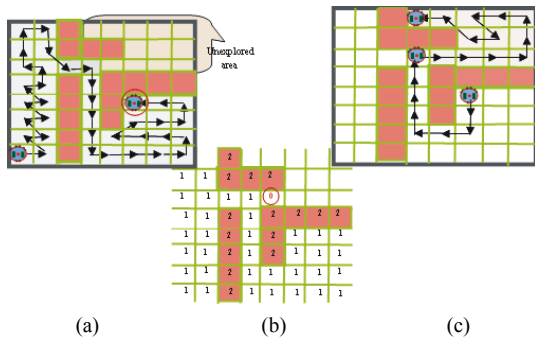


Figure 2: (a) the trapped robots Problem in LNA. (b) The map associated to the environment. (c) The shortest path that the robot follows to reach to the unexplored area and complete the exploration task.

Unexplored (0). No robot has been in the cell yet. As shown in Figure 2b, the cell with state 0 is detected by Khepera sensors as an unexplored free cell.

Explored (1). The cell has been traversed at least once by the robot, but it might need to go through it

again in order to reach unexplored regions. It also means that the cell is free.

Wall (2). The cell cannot be traversed by the robot because it is blocked by an obstacle or a wall.

We had to differentiate between the explored cells that contain an obstacle and the explored cells that are empty, in order to be able to identify the back path of the robot successfully if required.

4.1 Cost of Reaching a Target Location

To determine the cost of reaching the Frontier cells, which are the cells between known and unknown areas, we compute the optimal path from the current position of the robot to these cells based on a deterministic variant of the value iteration (Eq. (3)). In the following, $c_{x,y}$ corresponds to the x -th cell in the direction of the x -axis and the y -th cell in direction of the y -axis of the two-dimensional occupancy grid map. In our approach, the cost for traversing a grid cell $c_{x,y}$ is proportional to its occupancy value $p(c_{x,y})$. The minimum cost path is computed using the following two steps:

1. Initialization. The grid cell that contains the robot location is initialized with 0, all others with ∞ .

$$T_{x,y} = \begin{cases} 0, & \text{if } (x,y) \text{ is the position of the robot} \\ \infty, & \text{otherwise} \end{cases} \quad (2)$$

2. Update Loop. For all grid cells $C_{x,y}$ do:

$$T_{x,y} = \min \{ T_{x+\Delta x, y+\Delta y} + \sqrt{\Delta x^2 + \Delta y^2} \cdot P(C_{x+\Delta x, y+\Delta y}) \} \quad (3)$$

$$|\Delta x, \Delta y| \in \{-1, 0, 1\} \wedge P(C_{x+\Delta x, y+\Delta y}) \in [0, occ_{\max}]$$

Where occ_{\max} is the maximum occupancy probability value of a grid cell the robot is allowed to traverse. This technique updates the value of all grid cells by the value of their best neighbours, plus the cost of moving to this neighbour. Here, cost is equivalent to the probability $p(c_{x,y})$ that a grid cell $c_{x,y}$ is occupied times the distance to the cell. The update rule is repeated until convergence. Then each value $T_{x,y}$ corresponds to the *cumulative cost* of moving from the current position of the robot to $c_{x,y}$. The convergence of the algorithm is guaranteed as long as the cost for traversing a cell is not negative and the environment is bounded. Both criteria are fulfilled in our approach.

The resulting cost function T can also be used to efficiently derive the minimum cost path from the current location of the robot to arbitrary goal positions $c_{x,y}$. This can be done by steepest descent in T , starting at $c_{x,y}$. As shown in Figure 2c the algorithm will calculate the shortest path to the unexplored cell. As soon as the robot reaches this cell, it will complete exploring the environment using the cost equation (1).

5 EXPERIMENT USING THE MINI-ROBOT KHEPERA

To test the implemented exploration algorithm with real robot, we use the Teleworkbench. We built the environment on one small field (1 meter x 1 meter). We use Lego bricks to form the environment. As the robot platform we use mini-robot Khepera II <http://www.k-team.com>. Its dimension is 5 cm in diameter with one on-board microcontroller. The robot's base module is equipped with eight infrared sensors. The maximum detection range using the Khepera II basic setting. Up to 7cm distance. One of the advantages of this robot is that it is extensible, which means that diverse auxiliary modules can be added on top of it. To allow longer runtime and wireless communication, we extend the robot with our extension module consisting of an additional, battery and a Bluetooth chip. With this module, the robot can operate up to 3 hours continuously.

5.1 Teleworkbench

The Teleworkbench is a teleoperated platform and test bed for managing experiments using mini-robots (Tanoto, 2005). The system is accessible via the Internet. Through the web-based user interface, local or remote users can schedule experiments and set programs to be downloaded to each individual robot. Via a Bluetooth module, robots can exchange messages to each other or to the Teleworkbench Server wirelessly. During experiments, the video server tracks the robots on the field to provide position and orientation of the robots. In parallel, this data will be stored locally and streamed simultaneously as live-video via the Internet.

For experiment analysis purpose, we developed a graphical analysis tool based on the MPEG-4 video standard (Tanoto, 2006). This tool allows us to visualize the internal and external behaviour of robots. With this tool, the recorded video of the experiment can be displayed together with some

computer-generated objects representing important information, e.g. robot path, sensors' value, or exchanged messages. Moreover, users can interactively control the appearance of those objects during runtime. The Teleworkbench can use different types of mini-robots, such as Khepera II, as its robotic platform.

5.2 Algorithm Implementation

Based on the aforementioned algorithm, we develop the robot program in C language. The goal of the experiment is to explore an unknown bounded-environment of size 1 meter square. We divided the environment into 8 x 8 patches, each of which has a dimension of 0.125 x 0.125 m² as shown in Figure 3c. To detect obstacles or walls and their distance relative to the robot, we use robot's on-board infrared sensors. Thus, if an object is detected, the robot marks the patch with the object as occupied. Moreover, the robot gets its position by using odometry.

5.3 Experiment Setup and Execution

After code compilation, we download the program remotely via the Teleworkbench web-based user interface. During testing, we ran several experiments with different parameters, such as varying the threshold for the infrared sensors, etc. this is needed due to hardware unideality that is not taken into account during simulation. We did each experiment as follows: select a robot, download the program, turn on the webcam in record mode, free the robot after the experiment is over, and save the video data and the log files to be used for analysis. After each experiment, we ran the post experiment analysis tool which will generate an MPEG-4 video with the video of the experiment and the robot path as well as some colour tiles representing the patches.

5.4 Experimental Result

Experiments had been executed with the same environment setting and initial position. The result shows that by using the LNA the robot could explore all the free patches in the environment as shown in Figure 3a. But if the environment appears as shown in Figure 3b, the robot stops when it finds that all its neighbours are either explored or wall. But when applying the MLNA the robot could successfully explore all the free patches in the environment. Moreover, it could detect obstacles and walls

robustly by using only its on-board infrared sensors as shown in right image in Figure 4.

The MPEG-4 video played-back on OSMO4 video player shows the path of the robot during the experiment. If needed and available, other information, such as sensors, internal state, etc, can also be displayed.

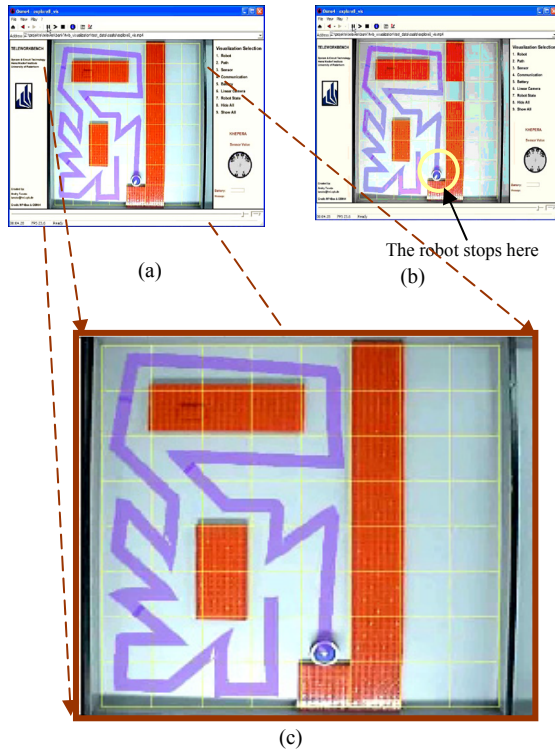


Figure 3: (a) Snapshots of the video visualizing the exploration experiment using LNA. (b) The trapped robot problem in LNA. (c) Zooming on the environment

However, we can see that the robot path deviates when it travelled from one patch to the other, as shown in Figure 3 and Figure 4. This is mainly due to the error generated by odometry. As a result, in some occasions the robot collided with obstacles or walls during its movement. Another interesting point from the experiment is that there is one occasion in which the robot had to select one of two patches with the same minimum cost (top-left image in Figure 4). By using the distance calculation (sec.4.1), the robot chose the patch exactly below it because of its shorter relative distance to the current robot position compared to the one of other cells.

6 CONCLUSIONS

In this work the modified local navigation strategy for static unknown environment exploration has been implemented and tested using the mini-robot Khepera running on the Teleworkbench. Experiments presented in this paper demonstrate that the modified exploration algorithm is able to cover successfully the whole unknown environment and overcome the draw back in LNA. Moreover, by taking into consideration the distance of neighbouring cells to the current robot position, the robot always select the cells with minimum distance, thus less energy and time. We notice also the weakness of odometry to provide the robot's position. To improve it, we plan to get more robust position information from the Teleworkbench. Moreover, varying environment setups and initial locations are necessary to prove the robustness of the algorithm.

ACKNOWLEDGEMENTS

This work is a result of common research between the *System and Circuit Technology, Heinz Nixdorf Institute, University of Paderborn* and *Scientific Computing Department, Faculty of Computer and Information Sciences, Ain Shams University, cultural department and study mission*. This work was partly supported by the Sixth Framework Programme of the European Union as part of the GUARDIANS project (no. 045269).

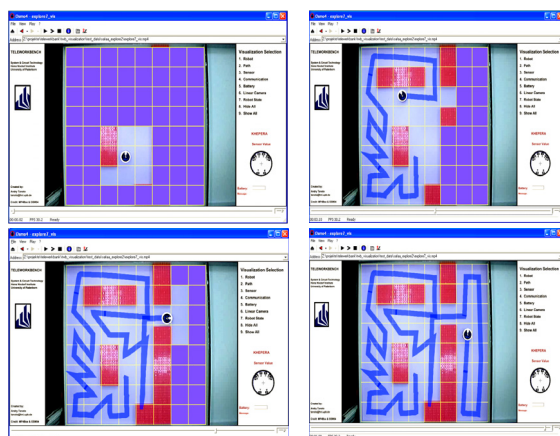


Figure 4: Snapshots of the video visualizing a step by step exploration experiment of an unknown environment using the MLNA.

REFERENCES

- Stachniss, C., 2006. Exploration and Mapping with Mobile Robots. *PhD Thesis, Albert-Ludwigs-University, Freiburg, Germany.*
- Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., and Sounes, H., 2000. Coordination for multi-robot exploration and mapping. *In AAAI National Conference on Artificial Intelligence, Austin, TX, USA.*
- Manolov, Ognyan; Iske, Burkhard; Noykov, Sv.; Klahold, Jürgen; Georgiev, G.; Witkowski, Ulf; Rückert, Ulrich . 2003. Gard - An Intelligent System for Distributed Exploration of Landmine Fields Simulated by a Team of Khepera Robots. *In Proceedings of the International Conference Automatics and Informatics, Sofia, Bulgaria.*
- S. Amin, A.Tanoto, U. Witkowski, U. Rückert, M. S. Abdel-Wahab, 2007. Environment Exploration Using Mini-Robot Khepera, *In Proceedings of the, 4th International Symposium on Autonomous Mini-robots for Research and Edutainment (AMiRE 2007), Buenos Aires, Argentina.*
- Burgard, W., Fox, D., Moors, M., R. Simmons and Thrun, S., 2000. Collaborative Multi-Robot Exploration, *In Proc. Intl. Conf.on Robotics and Automation, San Francisco.*
- Tanoto, A., Witkowski, U., Rueckert, U. (2005) Teleworkbench: A Teleoperated Platform for Multi-Robot Experiments, *In Proceedings of the 3rd International Symposium on Autonomous Mini-robots for Research and Edutainment (AMiRE 2005), Fukui, Japan.*
- Tanoto, A., Du, J., Kaulmann, L., Witkowski, U., 2006. MPEG-4-Based Interactive Visualization as an Analysis Tool for Experiments in Robotics, *In Proceeding of the 2006 International Conference on Modeling, Simulation and Visualization Methods, Las Vegas, USA.*
- Choset, H. Coverage for robotics - a survey of recent results. 2001. *Annals of Mathematics and Artificial Intelligence.*
- Bourgoult, F, Makarenko, A, Williams, B, Grocholsky, B., and DurrantWhyte, F., 2002. Information based adaptive robotic exploration. *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).*
- Eddinger, T, and Puttkamer, E., 1994. Exploration of an indoor-environment by an autonomous mobile robot. *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).*
- Yamauchi, B., Schultz, A. and Adams, W., 1999. Integrating exploration and localization for mobile robots. *Adaptive Systems, 7(2).*
- González, H. Baños, H, and Latombe, J, 2001. Navigation strategies for exploring indoor environments. *International Journal of Robotics Research.*
- Moorehead, S, Simmons, R, and Whittaker, W., 2001. Autonomous exploration using multiple sources of information. *In Proc. of the IEEE Int. Conf. on Robotic.*
- Hans P. Moravec and Elfes, A., 1985. High resolution maps from wide angle sonar. *In Proc. IEEE Int. Conf. Robotics and Automation.*
- Whaite, P and Ferri, F., 1997. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(3):193–205*
- Taylor, C and Kriegman, D., 1993. Exploration strategies for mobile robots. *In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), pages 248–253.*
- Dudek, G, Jenkin, M, Milios, E and Wilkes, D., 1991. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation, 7(6):859–865.*
- Koenig, S., Tovey, C., and Halliburton, W., 2001. Greedy mapping of terrain. *In Proc. Of the IEEE Int. Conf. on Robotics & Automation (ICRA).*
- Makarenko, A., Williams, B., Bourgoult, F., and Durrant-Whyte, F., 2002. An experiment in integrated exploration. *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Lausanne, Switzerland.*
- Stachniss, C., Hähnel, D., Burgard, W., and Grisetti, G., 2005. On actively closing loops in grid-based FastSLAM. *Advanced Robotics.*
- Stachniss, C., and Burgard, W., 2005. Mobile robot mapping and localization in nonstatic environments, *In Proc. of the National Conference on Artificial Intelligence (AAAI),Pittsburgh, PA, USA.*
- Dissanayake, G., Newman, P., Clark, S., Durrant-Whyte, F., and Csobor, M., 2001. A solution to the simultaneous localization and map building (SLAM) problem, *IEEE Trans. on Robotics and Automation..*
- Choset, H., 2001. Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation.*
- Moorehead, S., Simmons, R., and Whittaker, L., 2001. Autonomous exploration using multiple sources of information, *In IEEE International Conference on Robotics and Automation (ICRA '01).*
- Stachniss, C, Burgard, W., 2003. Exploring unknown environments with mobile robots using coverage maps. *In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI).*

LegOSC

Mindstorms NXT Robotics Programming for Artists

Jorge Cardoso

*Research Centre for Science and Technology in Art (CITAR), Portuguese Catholic University
Rua Diogo Botelho 1327, 4169-005 Porto, Portugal
jccardoso@porto.ucp.pt*

Manuel Ferreira, Cristina Santos

*Department of Industrial Electronics, University of Minho, Campus de Azurem – Guimarães, Portugal
mjf@dei.uminho.pt, cristina@dei.uminho.pt*

Keywords: Robots, Art, Lego Mindstorms, OSC.

Abstract: Robotics is an interesting but difficult area for digital artists who generally don't have much academic background on electronics or computer programming. Digital art students normally use high-level application to program their visual and sonorous installations. This paper presents LegOSC - a tool that allows the control of the Mindstorms NXT robots from any application that uses the Open Sound Control protocol which is implemented by most of those high-level applications. This allows artists to create works which incorporate robotic parts using the familiar programming environment.

1 INTRODUCTION

Robotics are becoming increasingly interesting for artists in many areas, e.g., painting (Moura and Pereira, 2004), theater (Ullanta, 2007), sculpture (Pisaturo, 2007), installation (da Costa, 2007), music (f18institute, 2007). More and more, art work incorporates some electro-mechanic parts which provide more ways for the artist to express himself, or to complement his ability to do so.

However, using robotic systems still requires some expertise that most artists don't possess. Even in digital art degrees, robotics is usually not a subject. Art students generally lack the necessary background in electronics.

Although there are now some simple tools to build and program robotic systems (of which, perhaps, the most widely known is the Lego Mindstorms (Lego Group, 2007)) and many uses in classrooms (Fagin, 2003; Klassner and Anderson, 2003; Bruder and Wedeward, 2003; Ceccarelli, 2003), these can still be difficult to integrate in an art work.

As an example, students in the author's school usually use platforms like Processing (Fry and Reas, 2007), Eyesweb (Camurri et al., 2000), Adobe Flash (Adobe, 2007b), Adobe Director (Adobe, 2007a), Max/MSP (Cycling74, 2007), Pure Data (Puckette, 1996), to implement their visual

and sonorous installations. These platforms can (and usually are) be interconnected using MIDI messages <http://www.midi.org>, or Open Sound Control (OSC) (Wright and Freed, 1997) messages.

In order to provide an easier setting for the use of a robotic system that can be controlled by a platform like the ones listed above, LegOSC has been implemented – an Open Sound Control gateway application to control the Lego Mindstorms NXT robotic system.

The rest of this paper is organized as follows: section 2 introduces the Lego Mindstorms NXT system; section 3 describes the Open Sound Control protocol; section 4 presents the architecture and usage of the LegOSC application; section 5 describes some usages of LegOSC; section 6 describes some of the limitations of LegOSC; finally, section 7 concludes.

2 MINDSTORMS NXT

The Lego Mindstorms NXT system consists of three main component types: the NXT brick; motors and sensors and assorted Lego bricks.

The NXT brick has a 32-bit ARM7 microcontroller, 256 Kbytes FLASH, 64 Kbytes RAM. It has Bluetooth wireless communication (Bluetooth Class II V2.0 compliant) a USB full speed port (12 Mbit/s),

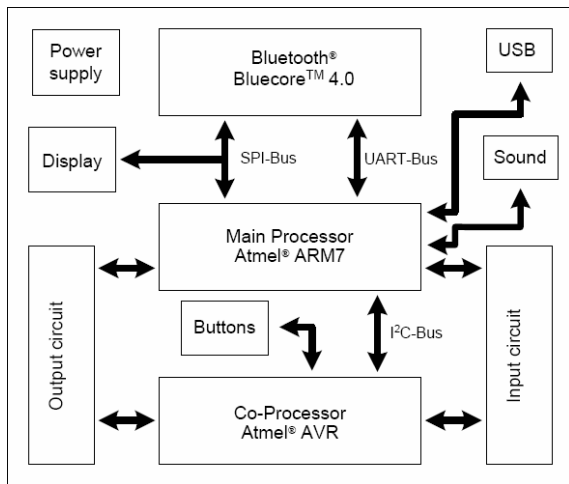


Figure 1: Hardware block diagram for the NXT brick, from (Lego Group, 2006b).

4 input ports, 3 output ports, 100 x 64 pixel LCD graphical display and a loudspeaker - 8 kHz sound quality. Figure 1 shows the block diagram of the NXT brick.

Three servo-motors can be connected to the three output ports and sensors (light, ultrasonic, pressure, sound, etc) to the four input ports.

The rest of the Lego bricks allow the construction of various shaped and sized robots.

The Mindstorms NXT robots can be controlled by uploading a program to the NXT brick and have it run in an autonomous fashion. These programs can be written using the Mindstorms NXT visual programming software. In alternative, one can use other languages with syntaxes close to C, such as “Not eXactly C” <http://briccc.sourceforge.net/nbc> or with Java syntaxes such as “LeJos” <http://lejos.sourceforge.net>, although some may require changing the NXT firmware.

The robot can also be controlled wirelessly by using the Bluetooth Direct Commands protocol (Lego Group, 2006a). This protocol allows sending instructions to actuate the motors or read sensors without the need to previously upload a program to the NXT brick. It also provides a way to interface the robot with other programs that understand this bluetooth protocol.

2.1 Bluetooth Protocol

Figure 2 shows the block diagram for the communication between a PC and Lego NXT.

Communication can be accomplished by using an USB cable connecting the PC and the NXT or wirelessly by using bluetooth.

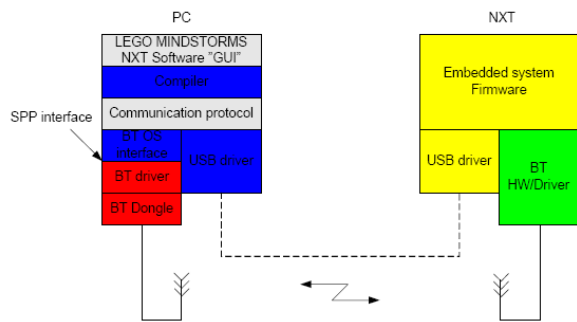


Figure 2: Communication block diagram, from (Lego Group, 2006a).

The protocol can be used to (based on (Lego Group, 2006a)):

1. Read, write and delete files.
2. Direct communication with the NXT system to:
 - Send direct commands to the virtual machine.
 - Send message commands to program mail-boxes.
 - Get file list within NXT.

The bluetooth protocol package is shown on Figure 3.

Bytes 0 and 1 are the LSB and MSB bytes, respectively, of the length of the command data.

Byte 2 is the command type. The 7 least significant bits identify the command type and the most significant bit (bit 7) determines if the command requires a reply from the NXT, or not. The command type can be one of the following:

- 0x00: Direct command, reply required.
- 0x01: System command, reply required.
- 0x02: Reply command.
- 0x80: Direct command, reply not required.
- 0x81: System command, reply not required.

Byte 3 identifies the command.

Byte 4-N are the command specific data.

LegOSC uses only the Direct Commands subprotocol to communicate with the NXT.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte N
Length, LSB	Length, MSB	Command Type	Command		

Figure 3: Bluetooth Protocol package.

3 OPEN SOUND CONTROL

The Open Sound Control is an application level communication protocol. It was meant to replace MIDI but, although it was not successful at that, it has become a widely used protocol in sound synthesis and video processing applications and many general purpose programming environments (see <http://www.cnmat.berkeley.edu/OpenSoundControl/> for a more comprehensive list).

OSC is a simple message based, transport-independent protocol, although most of its implementations use UDP or TCP as the transport layer. OSC messages have an address and a variable number of typed arguments. OSC standard types include 32-bit integers and floats, strings, blobs and 64-bit fixed point timetags.

An OSC Message consists of the following parts:

Address Pattern	Type Tag	Arg 0	...	Arg n
-----------------	----------	-------	-----	-------

The OSC Address Pattern is an OSC String¹ that starts with the '/' character. The OSC Address Pattern is pattern-matched by the receivers to decide if a message should be delivered.

The Type Tag is also an OSC String in which each character represents the type of an OSC Argument in the message.

Each OSC Message may have a variable number of binary represented arguments. Each argument representation is padded with zeroes to make it a multiple of 4.

4 LegOSC

LegOSC is a gateway application that translates a set of pre-defined OSC messages into Bluetooth Direct Commands for the NXT brick, and vice-versa, as shown in Figure 4.

To configure LegOSC we need to define the local UDP port on which it will listen for OSC messages, the IP address and port of the OSC Application that will be communicating with LegOSC (and listening for OSC messages) and the virtual COM port on which the NXT Brick was connected.

Figure 5 shows a screenshot of the LegOSC application.

LegOSC will listen for OSC messages that tell it to actuate the motors or to read sensor values. In case of the latter, it will respond with another OSC message with the sensor value. For some applications however,

¹An OSC String is null-terminated string of ASCII characters, padded with nulls to make the total number of characters a multiple of 4.

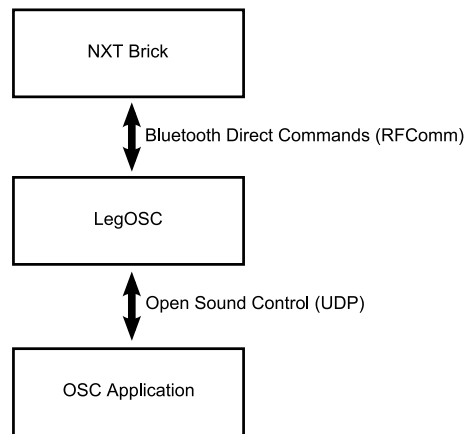


Figure 4: Communication between the NXT brick, LegOSC and the OSC Application.

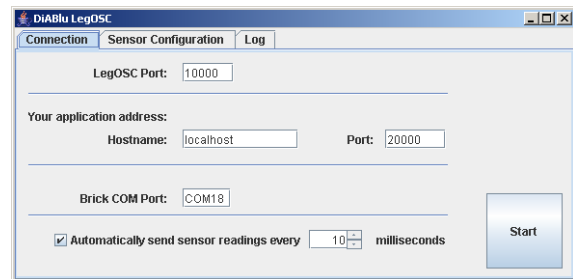


Figure 5: LegOSC Application.

reading sensors will be a continuous operation so, in order to save some OSC traffic, LegOSC can be programmed to continuously send sensor values without being asked for. The user can tell LegOSC that automatic readings are required and how often a reading should be made.

In order to be able to read a sensor (automatically), LegOSC must know the type of sensor connected to each port in the NXT brick. The user can tell LegOSC the type of sensor using the Sensor Configuration tab (Figure 6).

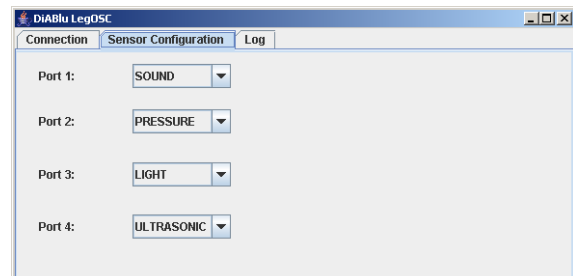


Figure 6: Configuring automatic sensor reading.

4.1 OSC Messages

The list of currently implemented OSC messages that LegOSC understands is:

- /motorForward ii – The first integer argument is the motor number and the second is the power to apply to the motor. This message will make the specified motor start to rotate at the specified power.
- /motorSlowStop i – The integer argument is the motor number. This message stops the specified motor without applying “brakes”.
- /motorBrake i – The integer argument is the motor number. This message stops the specified motor and applies “brakes”.
- /resetMotor i – The integer argument is the motor number. This message resets the tachometer of the specified motor.
- /getMotorTachoCount i – The integer argument is the motor number. This message asks for the current tacho count of the specified motor and originates a /motorTachoCount message as the reply.
- /getButtonState i – The integer argument is the port to which the pressure sensor is attached. This message asks for the current state of the pressure sensor and originates a /buttonState message as the reply.
- /getLightLevel i – The integer argument is the port to which the light sensor is attached. This message asks for the current value of the light sensor and originates a /lightLevel message as the reply.
- /getSoundLevel i – The integer argument is the port to which the sound sensor is attached. This message asks for the current value of the sound sensor and originates a /soundLevel message as the reply.
- /getProximityLevel i – The integer argument is the port to which the ultrasonic sensor is attached. This message asks for the current value of the ultrasonic sensor and originates a /proximityLevel message as the reply.
- /getBatteryLevel – This message asks for the current voltage of the battery of the NXT.

Some of the above messages generate a response:

- /motorTachoCount ii – Response to /getMotorTachoCount. The first integer argument is the motor number; the second integer argument is the current tacho count for that motor.
- /buttonState ii – Response to /getButtonState. The first integer argument is the port number to which

the pressure sensor is attached; the second integer argument if the current state of the pressure sensor (0 – not pressed; 1 – pressed).

- /lightLevel ii – Response to /getLightLevel. The first integer argument is the port number to which the light sensor is attached; the second integer argument if the current value of light sensor.
- /soundLevel ii – Response to /getSoundLevel. The first integer argument is the port number to which the sound sensor is attached; the second integer argument if the current value of sound sensor.
- /proximityLevel ii – Response to /getProximityLevel. The first integer argument is the port number to which the ultrasonic sensor is attached; the second integer argument if the current value of ultrasonic sensor.
- /batteryLevel i – Response to /getBatteryLevel. The integer argument is the current voltage in millivolts.

4.2 Java Libraries

LegOSC was written using the Java programming language and as a by-product of developing this gateway we developed a Java library that implements the NXT Bluetooth low-level commands as well as a higher level library that abstracts these low-level commands into higher level NXT-related objects. The class diagram is shown on Figure 7 (details of each class are hidden to save space).

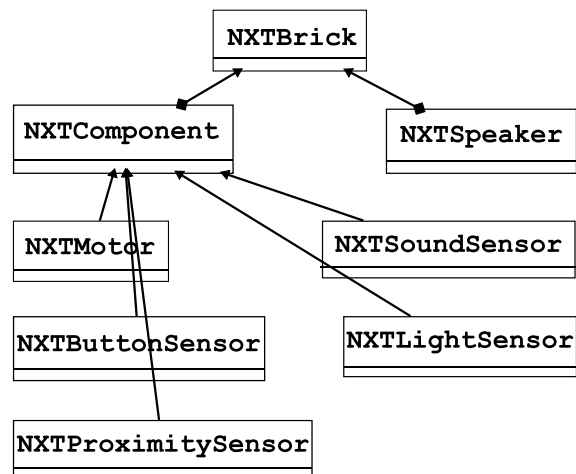


Figure 7: High-level Java library class diagram.

Since we had to develop this Java library and since the modifications were small, we decided to adapt the Java library to a Processing library. Processing

is a tool/programming language widely used in digital arts, so it made sense to allow direct control of the NXT without the need to use a different program. Processing libraries are usually composed of a single class to simplify its usage as much as possible. The (single) class diagram for this library is shown on Figure 8.

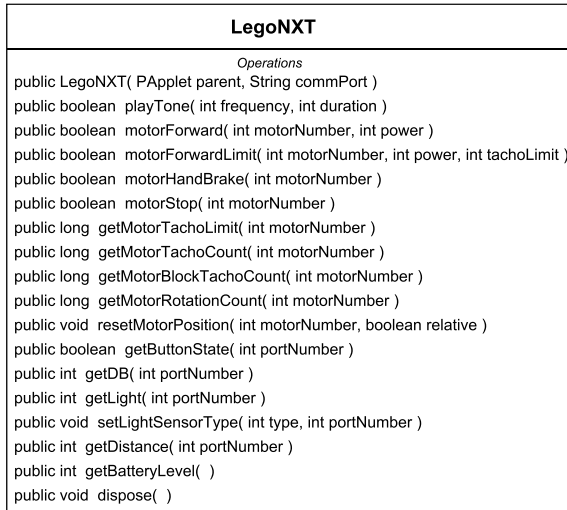


Figure 8: Processing library class diagram.

5 EXAMPLE USAGES

To give a better idea on how LegOSC can be used, a brief description of some typical exercises and implementations that students are asked to do is given next.

The first one is a non-classic way of controlling a robot. Instead of using a mouse/joystick/keyboard, students are asked to think of a way to control a robot using sound. Figure 9 shows an implementation using Pure Data – an audio synthesis platform. In this example, a stereo microphone is used to control two motors. The sound level at each channel drives its own motor, thus enabling the user to direct the robot by making sound louder at one microphone or the other.

The second example deals with mapping some physical parameter of the robot, e.g., distance to a wall, light level reading, into another type of signal. In this example (Figure 10) a simple Theremin was implemented. Readings from the proximity sensor and from the light sensor are used to drive two audio oscillators that together generate a frequency modulated audio signal. In this case the movement of the robot (which could be programmed in a different number of ways) generates an audio signal.

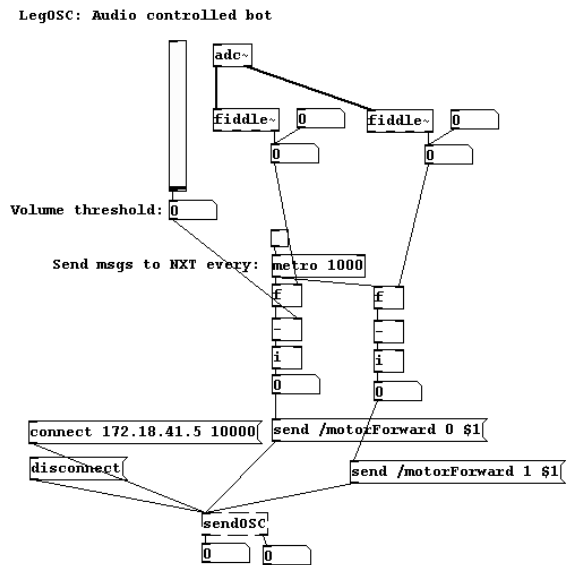


Figure 9: Pure Data implementation of a sound controlled robot.

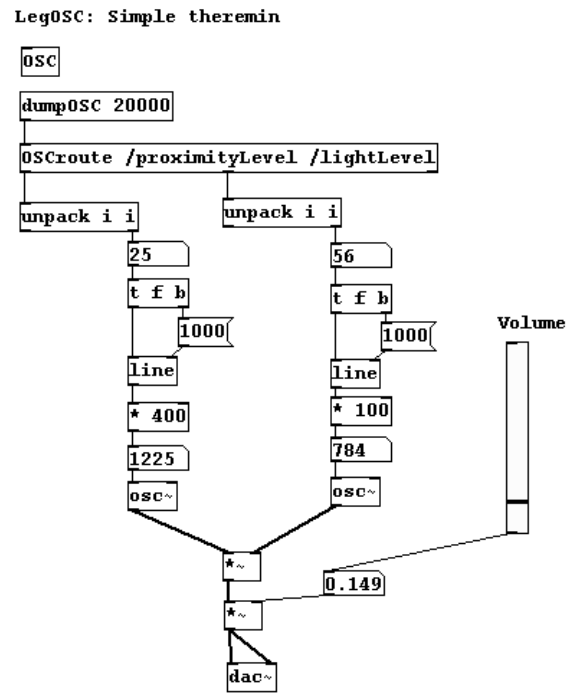


Figure 10: Pure Data implementation of a robot theremin.

Both examples were implemented in Pure Data and the figures show the complete program. Pure Data is an audio synthesis and manipulation platform so it simplifies the kind of programming needed to implement the examples.

6 LIMITATIONS

LegOSC was developed with an education and artistic purposes in mind. It is not intended for precision robotics.

The understood messages were kept as simple as possible in order to allow basic control of the robots but not to overwhelm the student/artist with detail.

The bluetooth communication latency may also make it unsuitable for some applications where a rapid response to an event is required.

7 CONCLUSIONS AND FUTURE WORK

We have developed an Open Sound Control gateway application for controlling the Lego Mindstorms NXT robotics system. This application is intended to be used by digital art students as a simple way to control and integrate robotic art work with other often used systems to develop visual and sonorous installations.

We hope it will allow an easier first approach to teaching and using robotics in the digital arts area.

LegOSC is currently being used by the author's students and we hope to enhance it with the experience gained with its usage.

One of the improvements that we have already gathered from experiences is the ability to control several NXT using the same LegOSC instance. This will facilitate the programming of applications that make use of several robots at the same time.

The tool is freely available for download at <http://diablu.googlecode.com/svn/trunk/LegOSC/>. The Processing library is also available from the Processing site at <http://processing.org>.

REFERENCES

- Adobe (2007a). Adobe director. <http://www.adobe.com/products/director/>.
- Adobe (2007b). Adobe flash. <http://www.adobe.com/products/flash/>.
- Bruder, S. and Wedeward, K. (2003). Robotics in the classroom. *IEEE Robotics & Automation Magazine*, 10(3):25–29.
- Camurri, A., Hashimoto, S., Ricchetti, M., and et al (2000). Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal*, 24(1):57–69.
- Ceccarelli, M. (2003). Robotic teachers' assistants. *IEEE Robotics & Automation Magazine*, 10(3):37–45.
- Cycling74 (2007). Max/msp. <http://www.cycling74.com>.

- da Costa, B. (2007). Beatriz da costa. <http://www.beatrizdacosta.net/index.php>.
- f18institute (2007). Cellobot. <http://www.f18institut.org/f18institut/f18robotics.html>.
- Fagin, B. (2003). Ada/mindstorms 3.0. *IEEE Robotics & Automation Magazine*, 10(2):19–24.
- Fry, B. and Reas, C. (2007). Processing.org. <http://www.processing.org>.
- Klassner, F. and Anderson, S. D. (2003). Lego mindstorms: not just for k-12 anymore. *IEEE Robotics Automation Magazine*, 10(2):12–18.
- Lego Group (2006a). Lego mindstorms nxt bluetooth developer kit. <http://mindstorms.lego.com/Overview/NXTreme.aspx>.
- Lego Group (2006b). Lego mindstorms nxt hardware developer kit. <http://mindstorms.lego.com/Overview/NXTreme.aspx>.
- Lego Group (2007). Lego.com mindstorms nxt home. <http://mindstorms.lego.com>.
- Moura, L. and Pereira, H. G. (2004). *Man + Robots : Symbiotic Art*.
- Pisaturo, C. (2007). Works of carl pisaturo. <http://www.carlpisaturo.com/index.html>.
- Puckette, M. (1996). Pure data: another integrated computer music environment. In *Proceedings of the Second Intercollege Computer Music Concerts*, pages 37–41.
- Ullanta (2007). Performance robotics. <http://www.ullanta.com/ullanta/>.
- Wright, M. and Freed, A. (1997). Opensound control: A new protocol for communicating with sound synthesizers. In *Proceedings of the 1997 International Computer Music Conference*.

FOOTSTEP PLANNING FOR BIPED ROBOT BASED ON FUZZY Q-LEARNING APPROACH

Christophe Sabourin, Kurosh Madani

*Laboratoire Images, Signaux, et Systèmes Intelligents (LISSI EA / 3956), Université Paris-XII
IUT de Sénart, Avenue Pierre Point, 77127 Lieusaint, France
sabourin@univ-paris12.fr; madani@univ-paris12.fr*

Weiwei Yu, Jie Yan

*Flight Control and Simulation Institute, Northwestern Polytechnical University, Xi'an 710072, China
yuweiwei_shirley@hotmail.com*

Keywords: Biped robots, Footstep planning, Fuzzy Q-Learning.

Abstract: The biped robots have more flexible mechanical systems and they can move in more complex environment than wheeled robots. Their abilities to step over both static and dynamic obstacles allow to the biped robots to cross an uneven terrain where ordinary wheeled robots can fail. In this paper we present a footstep planning for biped robots allowing them to step over dynamic obstacles. Our footstep planning strategy is based on a fuzzy Q-learning concept. In comparison with other previous works, one of the most appealing interest of our approach is its good robustness because the proposed footstep planning is operational for both constant and random velocity of the obstacle.

1 INTRODUCTION

In contrast with the wheeled robots, the biped robots have more flexible mechanical system and thus they can move in more complex environment. Actually, their abilities to step over both static and dynamic obstacles allow to the biped robots to cross an uneven terrain where regular wheeled robots can fail. Although there are a large number of papers dealing with the field of biped and humanoid robots (see for examples (Hackel, 2007) and (Carlos, 2007)), only a few of publication researches concern the path planning for biped robots (Ayza, 2007), (Chestnutt, 2004), (Sabe, 2004). In fact, the design of a path planning for biped robots into indoor and outdoor environment is more difficult than for wheeled robots because it must take into account their abilities to step over obstacles. Consequently, path planning with obstacle avoidance strategy like the wheeled robots is not sufficient.

Generally, the previous proposed approaches in the field of path planning for biped robots are based on a tree search algorithm. In (Kuffner, 2001), Kuffner et al. have proposed a footstep planning approach using a search tree from a discrete set of feasible footstep locations. This approach has been validated on the robot H6 (Kuffner, 2001) and H7 (Kuffner, 2003). Later, this strategy has been extended for the robot

Honda ASIMO (Chestnutt, 2005). Although the footstep planning proposed by Kuffner seems an interesting way to solve the problem of the path planning for biped robots, the main drawbacks are on the one hand the limitation at 15 foot placements (Kuffner, 2001) in order to limit the computational time, and on the other hand, this approach is operational only in the case of the predictable dynamic environments (Chestnutt, 2005). In this paper, we present a new concept of a footstep planning for biped robots in dynamic environments. Our approach is based on a Fuzzy Q-learning (FQL) algorithm. The FQL, proposed by Glorennec et al. (Glorennec, 1997) (Jouffe, 1998), is an extension of the traditional Q-learning concept (Watkins, 1992) (Sutton, 1998) (Glorennec, 2000) allowing to handle the continuous nature of the state-action. In this case, both actions and Q-function may be represented by Takagi-Sugeno Fuzzy Inference System (TS-FIS). After a training phase, our footstep planning strategy is able to adapt the step length of the biped robot only using a Fuzzy Inference System. However, our study is limited to the sagittal plane and does not take into account the feasibility of the joint trajectories of the leg. In fact, the footstep planning gives only the position of the landing point. But the first investigations show a real interest of this approach because:

- The computing time is very short. After the learning phase, the footstep planning is based only on a FIS,
- The footstep planning is operational for both predictable and unpredictable dynamic environment allowing to increase the robustness.

This paper is organized as follows. In Section 2, the Fuzzy Q-learning concept is presented. Section 3 describes the footstep planning based on the Fuzzy Q-learning. In section 4, the main results, obtained from simulations, are given. Conclusions and further developments are finally set out in section 5.

2 FUZZY Q-LEARNING CONCEPT

Reinforcement learning (Sutton, 1998) (Glorennec, 2000) involves problems in which an agent interacts with its environment and estimates consequences of its actions on the base of a scalar signal in terms of reward or punishment. The goal of the reinforcement learning algorithm is to find the action which maximize a reinforcement signal. The reinforcement signal provides an indication of the interest of last chosen actions. Q-Learning, proposed by Watkins (Watkins, 1992), is a very interesting way to use reinforcement learning strategy. However, the Q-Learning algorithm developed by Watkins deals with discrete cases and assumes that the whole state space can be enumerated and stored in a memory. Because the Q-matrix values are stored in a look-up table, the use of this method becomes impossible when the state-action spaces are continuous. For a continuous state space, Glorennec et al. (Glorennec, 1997) (Jouffe, 1998) proposed to use fuzzy logic where both actions and Q-function may be represented by Takagi-Sugeno Fuzzy Inference System (TS-FIS). Unlike the TS-FIS in which there is only one conclusion for each rule, the Fuzzy Q-Learning (FQL) approach admits several actions per rule. Therefore, the learning agent has to find the best issue for each rule.

The FQL algorithm uses a set of N_K fuzzy rules such as:

$$\text{IF } x_1 \text{ is } M_1^1 \text{ AND } x_i \text{ is } M_i^j \text{ THEN } \begin{cases} y_k = a_k^1 & \text{with } q = q_k^1 \\ \text{or } y_k = a_k^l & \text{with } q = q_k^l \\ \text{or } y_k = a_k^{N_l} & \text{with } q = q_k^{N_l} \end{cases} \quad (1)$$

x_i ($i = 1..N_i$) are the inputs of the FIS which represent the state space, N_i is the size of the input space. Each

fuzzy set j for the input i is modeled by a membership function M_i^j and its membership value μ_i^j . a_k^l and q_k^l are respectively the l^{th} possible action for the rule k and its corresponding Q-value ($k = 1..N_k; l = 1..N_l$). At each step time t , the agent observes the present state $X(t)$. For each rule k , the learning system has to choose one action among the total N_l actions using an Exploration/Exploitation Policy (EEP). In our approach, ϵ -greedy algorithm is used to select the local action for each activated rule. The action with the best evaluation value ($\max(q_k^l), l = 1..N_l$) has a probability P_ϵ to be chosen, otherwise, an action is chosen randomly among all possible actions. After, the execution of the next computed action, the agent may update the Q-value using of a reinforcement signal. The algorithm of the FQL may be decomposed into four stages:

- After the fuzzification of the perceived state $X(t)$, the rule values $\alpha_k(t)$ are computing using equation (2):

$$\alpha_k(t) = \mu_1^j \mu_2^j \dots \mu_{N_i}^j \quad (2)$$

- The final action $Y(t)$ is computed through two levels of computation: in the first level, local action l in each activated rule is determined by using EEP, and in the second level global action is calculated as a combination of all local actions. Equations (3) and (4) give respectively the computation of the global action $Y(t)$ and the corresponding $Q(t)$ value according to the truth value $\alpha_k(t)$:

$$Y(t) = \sum_{k=1}^{N_k} \alpha_k(t) a_k^l(t) \quad (3)$$

$$Q(t) = \sum_{k=1}^{N_k} \alpha_k(t) q_k^l(t) \quad (4)$$

- Matching up the new action, given by $Y(t)$ and taking into account the environment's reply, $Q(t)$ may be updated using equation (5):

$$\Delta Q(t) = \beta[r + \gamma V_{max}(t+1) - Q(t)] \quad (5)$$

Where $V_{max}(t+1)$ is the maximum Q-value for the activated rule at the next step time $t+1$:

$$V_{max}(t+1) = \sum_{k=1}^{N_k} \alpha_k(t+1) \max(Q_k^l(t+1)) \quad (6)$$

γ is a discount factor which can be chosen from 0 to 1. If it is close to 0, the reinforcement information tends to consider only the immediate reward, while if it is closer to 1, it considers the future

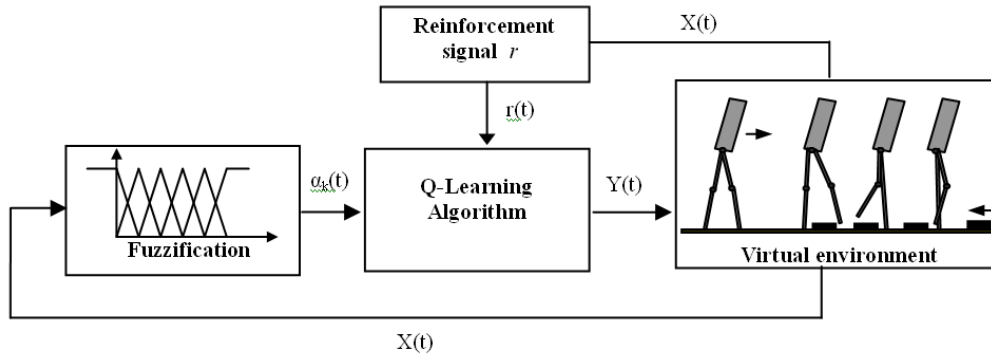


Figure 1: Footstep planning strategy.

rewards with greater weight. β is a learning rate parameter allowing to weight the part of the old and new rewards in a reinforcement signal r .

- Finally, for each activated rules, the corresponding elementary quality Δq_k^l of the Q-matrix is updated as:

$$\Delta q_k^l = Q(t)\alpha_k(t) \quad (7)$$

3 FOOTSTEP PLANNING

The proposed footstep planning is based on a FQL approach. Our aim is to design a control strategy allowing to adjust automatically the step length of a biped in order that the robot avoids dynamic obstacles by using step over strategy. As figure 1 shows it, our footstep planning may be divided into four parts:

- The first part involves a fuzzification of inputs of the state $X(t)$,
- The second concerns the FQL algorithm allowing to compute the length of the step,
- The third part allows simulating dynamic environment into which the robot moves,
- And the fourth part gives the reinforcement signal.

3.1 Virtual Dynamic Environment

The both robot and obstacle move in sagittal plane but in opposite directions. We consider that the walking of the biped robot may include as well strings of single support phases (only one leg is in contact with the ground) as instantaneous double support phases (the two legs are in contact with the ground). The biped robot may adjust the length of its step but we consider that the duration of each step is always equal

to $1s$. The size and velocity of the obstacle are included into $[0, 0.4m]$ and $[0, 0.4m/s]$ ranges respectively. Although the robot has the ability to adjust its step length, there are two possibilities in which the robot may crash with the obstacle. First one occurs when the length of the step is not correctly adapted according to the position of the dynamic obstacle. In this case, the swing leg touches directly the obstacle during a double support phase. The other case corresponds to the situation where the obstacle collides with the stance leg during the single or double support phase.

3.2 Fuzzification

The design of our footstep planning is based on both Takagi-Sugeno FIS and Q-learning strategies. Consequently, it is necessary to use a fuzzification for each input. In the proposed approach, we use two inputs in order to perform a correct footstep planning. These inputs are the distance between the robot and the obstacle d_{obs} and the velocity of the obstacle v_{obs} . d_{obs} and v_{obs} are updated at each double support phase. d_{obs} corresponds to the distance between the front foot and the first side of the obstacle. v_{obs} is computed from the distance covered during $1s$. The fuzzification of v_{obs} and d_{obs} is carried out by using respectively 6 and 11 triangular membership functions. Figure 2(a) and 2(b) gives the membership functions of the obstacle velocity and distance respectively.

3.3 FQL-based Step Length

The FQL algorithm uses a set of fuzzy rules such as equation (1). For the proposed problem, the number of the rules is 66 (6 and 11 membership functions for velocity and distance of the obstacle respectively). For each rules, we define 5 possible outputs which are $[0.1, 0.2, 0.3, 0.4, 0.5]m$. In fact, these outputs cor-

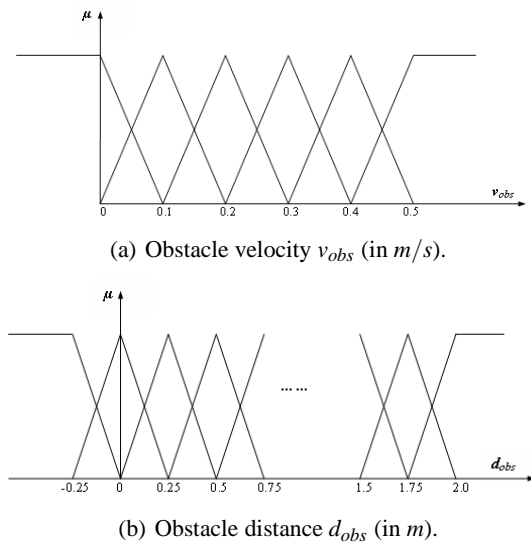


Figure 2: Membership functions used for the input space.

respond to the length of the step. Consequently, at each step time, the Fuzzy Q-Learning algorithm needs to choose one output among five possible outputs for each activated rules. It must be pointed out that the chosen output is included into a discrete set, but the real output $Y(t)$ is a real number due a fuzzification. During the simulation, the size of the obstacle is constant but the velocity of the obstacle may be modified. At each episode, initialization of some parameters are necessary. The initial distance between the biped robot and the obstacle is always equal to $2.5m$. The velocity of the obstacle is chosen randomly into the interval $[0, 0.4]m/s$. During one episode, the step length of the robot is computed using the FQL algorithm described in section 2. Consequently, the biped robot moves step by step towards the obstacle during the episode. The episode is finished whether the robot steps over the obstacle (success) or if the robot crashes into obstacle (failure). The discount factor γ and the learning rate parameter β are equal to 0.8 and 0.1 respectively. This parameters have been chosen empirically after several trials in order to assure a good convergence of FQL algorithm. The probability P_{ϵ} is equal to 0.1 and means that the random exploration is privileged during the learning phase.

3.4 Reinforcement Signal

The reinforcement signal provides an information in terms of reward or punishment. Consequently, the reinforcement signal informs the learning agent about the quality of the chosen action. In our case, the learning agent must find a succession of action allowing

to the biped robot to step over an obstacle. But here the obstacle is a dynamic object which moves towards the biped robot. Consequently, the reinforcement information have to take into account of the velocity of the moving obstacle. In addition, the position of the foot just before the stepping-over is very important as well. On the base of these considerations, we designed reinforcement signal in two parts.

Firstly, if $x_{rob} < x_{obs}$ where x_{rob} and x_{obs} give the positions of the robot and of the obstacle respectively:

- $r = 0$, if the robot is still far from obstacle,
- $r = 1$, if the position of the robot is appropriate to cross the obstacle at next step,
- $r = -1$, if the robot is too close to the obstacle.

In this first case, r is computed with the following equation:

$$r = \begin{cases} 0 & \text{if } (x_{rob} \leq (x_{obs} - 1.2v_{obs}\Delta t)) \\ 1 & \text{if } (x_{rob} > (x_{obs} - 1.2v_{obs}\Delta t)) \\ & \text{AND } (x_{rob} \leq (x_{obs} - 1.1v_{obs}\Delta t)) \\ -1 & \text{if } (x_{rob} > (x_{obs} - 1.1v_{obs}\Delta t)) \end{cases} \quad (8)$$

x_{rob} and x_{obs} are updated after each action. $v_{obs}\Delta t$ represents the distance covering by obstacle during the time Δt . As the duration of the step is always equal 1s, Δt is always equal to 1s.

Secondly, if $(x_{rob} \geq x_{obs})$:

- $r = -2$, if the robot crashes into the obstacle at the next step,
- $r = 2$, if the robot crosses the obstacle at the next step.

In this last case, r is given by equation (9).

$$r = \begin{cases} -2 & \text{if } (x_{rob} \leq (x_{obs} + L_{obs})) \\ 2 & \text{if } (x_{rob} > (x_{obs} + L_{obs})) \end{cases} \quad (9)$$

Where L_{obs} is the size of the obstacle.

4 SIMULATION RESULTS

In this section, we present the main results related the footstep planning based on FQL approach by using MATLAB software. It must be noticed that our goal is to design a control strategy allowing to give a path planning into a dynamic environment for biped robot but we do not take into account the dynamic of the biped robot. We consider only discrete information allowing to compute the landing position of the foot. In addition, we consider only flat obstacles in the following simulations.

4.1 Training Phase

During the training phase, the goal of the learning agent is to find the best rules in order that the biped robot crosses the obstacle. On the base of the previous description, we trained the Q-matrix during 10000 episodes. After a full training, we test the footstep planning approach with 1000 velocity samples covering uniformly the input range $[0, 0.4]m/s$.

Table 1 gives results about successes rate for four sizes of the obstacle. The rate success corresponds to the ratio between the number of successes and the totality of trials (1000). And the figure 3 shows an example of the repartition between the successes and the failures over an input range v_{obs} and when L_{obs} is equal to $0.2m$. When the robot can step over the obstacle successfully, the results is 1 otherwise it is 0.

Table 1: Rate success according to obstacle size.

Size (m)	0.1	0.2	0.3	0.4
Successes rate (%)	65.6	31.3	21.7	4.8

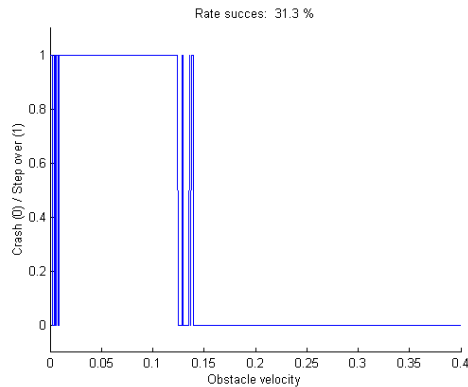


Figure 3: Successes rate when the size of the obstacle is equal to $0.2m$.

It must be pointed out that more the size is large, more the successes rate is weak. And like figure 3 shows it, there is a threshold ($0.12m/s$ approximately when $L_{obs} = 0.2m$) where our footstep planning never finds a solution. Consequently, the velocity of the obstacle must be limited if we want the biped crosses the obstacle successfully.

4.2 Footstep Planning Examples

Figure 4 shows a footstep sequence when the robot crosses an obstacle. The size of the obstacle is equal to $0.2m$ and its velocity is constant during all the simulation. Rectangles indicate the obstacle and the spots

indicate the two positions of the feet (left and right) for each step. Table 2 gives the step length for all the steps. It must be pointed out that when the biped robot is close to the obstacle, then the length of the step decreases in order to prepare the stepping over. Finally, the last step allows to avoid obstacle without collision.

Table 2: Length of the step L_{step} when $v_{obs} = 0.1m/s$ and $L_{obs} = 0.2m$.

Step	1	2	3	4	5	6
L_{step}	0.50	0.22	0.50	0.45	0.13	0.50

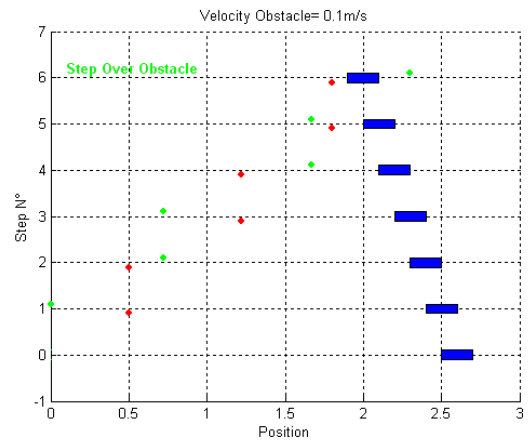


Figure 4: Successful footstep planning when $v_{obs} = 0.1m/s$ and $L_{obs} = 0.2m$.

It is pertinent to note that one of the most interesting point in our approach is its abilities to operate when the velocity of the obstacle is not constant. Figure 5 shows the footstep sequence when the obstacle moves with a random velocity. The velocity of the obstacle is carried out by the sum of a constant value which is equal to $0.1m/s$ and a random value included into $[-0.1..0.1]m/s$. Table 3 gives V_{obs} and L_{step} for each step. The size of the obstacle is equal to $0.1m$. It must be pointed out that the control strategy allows to adapt automatically the length of the step according to the obstacle velocity thanks to FQL algorithm. For 1000 trials realized in the same conditions, the successes rate is equal to 85% approximatively. This is very interesting because our strategy allows to increase the robustness of the footstep planning.

5 CONCLUSIONS

In this paper we have presented a footstep planning strategy for biped robots allowing them to step over

Table 3: Length of the step when v_{obs} is random and $L_{obs} = 0.1m$.

Step	1	2	3	4	5	6
v_{obs}	0.14	0.05	0.16	0.10	0.16	0.04
L_{step}	0.50	0.23	0.37	0.44	0.10	0.50

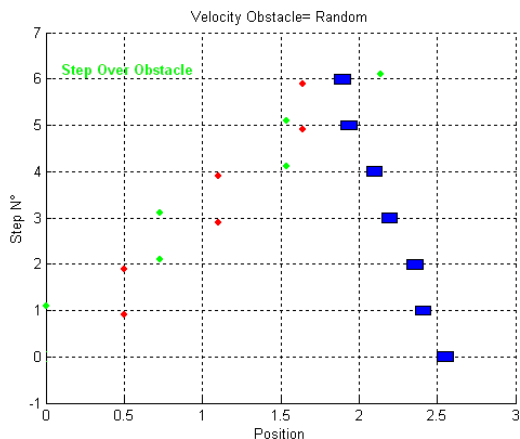


Figure 5: successful footstep planning when v_{obs} is random, $L_{obs} = 0.1m$.

dynamic obstacles. Our footstep planning tactic is based on a fuzzy Q-learning concept. The most appealing interest of our approach is its outstanding robustness related to the fact that the proposed footstep planning is operational for both constant and variable velocity of the obstacle.

Futures works will be focus on the improvement of our footstep planning strategy:

- First, our actual control strategy does not take into account the duration of the step. However, this parameter is very important with dynamic obstacles. Therefore, our goal is to enhance the proposal footstep planning in order to take care about both the length and the duration of the step,
- Second, in some cases, biped robot can not step over obstacle: for example when the size of the obstacle is too large. Consequently, the footstep planning must be able to propose a path planning in order to make the robot avoid obstacle.
- Third, in long-term, our goal is to design more general footstep planning based on both local footstep planning and global path planning,
- Finally, experimental validation may be consider on real humanoid robot. But in this case, it is necessary to design the joint trajectories based on the position of feet.

REFERENCES

M. Hackel. Humanoid Robots: Human-like Machines. *I-Tech Education and Publishing, Vienna, Austria*, June 2007 .

A. Carlos, P. Filho. Humanoid Robots: New Developments. *I-Tech Education and Publishing, Vienna, Austria*, June 2007.

Y. Ayza, K. Munawar, M. B. Malik, A. Konno and M. Uchiyama. A Human-Like Approach to Footstep Planning. *Humanoid Robots, I-Tech Education and Publishing, Vienna, Austria*, June 2007, pp.296–314

J. Chestnutt, J. J. Kuffner. A Tiered Planning Strategy for Biped Navigation. *Int. Conf. on Humanoid Robots (Humanoids'04), Santa Monica, California*, 2004.

K. Sabe, M. Fukuchi, J. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara. Obstacle Avoidance and Path Planning for Humanoid Robots using Stereo Vision. *Int. Conf. on Robotics Automation (ICRA)*. 2004, 592–597.

J.J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, H. Inoue. Footstep Planning Among Obstacles for Biped Robots. *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2001, 500–505.

J.J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, H. Inoue. Online Footstep Planning for Humanoid Robots. *Proceedings of IEEE/RSJ Int. Conf. on Robotics and Automation (ICRA)*, 2003, 932–937

J. Chestnutt, M. Lau, G. Cheung, J.J. Kuffner, J. Hodgins, T. Kanade. Footstep Planning for the Honda Asimo Humanoid. *Proceedings of IEEE Int. Conf. on Robotics Automation (ICRA)*, 2005, pp. 629–634

C. Watkins, P. Dayan. Q-learning. *Machine Learning*, 8, 1992, 279–292.

R.S. Sutton, A.G. Barto. Reinforcement Learning: An Introduction. *MIT Press, Cambridge, MA*, 1998.

P. Y. Glorionec. Reinforcement Learning: an Overview. *European Symposium on Intelligent Techniques (ESIT)*, 2000,17–35.

P.Y. Glorionec, L. Jouffe. Fuzzy Q-Learning *Proc. of FUZZ-IEEE'97, Barcelona*, 1997.

L. Jouffe. Fuzzy inference system learning by reinforcement methods. *IEEE Trans. on SMC, Part C*, August 1998, Vol. 28 (3).

DYNAMICAL MODELS FOR OMNI-DIRECTIONAL ROBOTS WITH 3 AND 4 WHEELS

Hélder P. Oliveira, Armando J. Sousa, A. Paulo Moreira and Paulo J. Costa
Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias s/n 4200-465, Porto, Portugal
{helder.oliveira, asousa, amoreira, paco}@fe.up.pt

Keywords: Identification, simulation, modeling and omni-directional mobile robots.

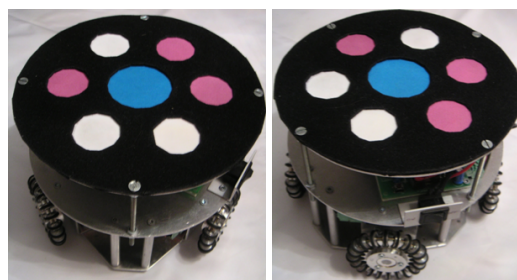
Abstract: Omni-directional robots are becoming more and more common in recent robotic applications. They offer improved ease of maneuverability and effectiveness at the expense of increased complexity. Frequent applications include but are not limited to robotic competitions and service robotics. The goal of this work is to find a precise dynamical model in order to predict the robot behavior. Models were found for two real world omni-directional robot configurations and their parameters estimated using a prototype that can have 3 or 4 wheels. Simulations and experimental runs are presented in order to validate the presented work.

1 INTRODUCTION

Omni-directional robots are becoming a much sought solution to mobile robotic applications. This kind of holonomic robots are interesting because they allow greater maneuverability and efficiency at the expense of some extra complexity. One of the most frequent solutions is to use some of Mecanum wheels (Diegel et al., 2002) and (Salih et al., 2006). A robot with 3 or more motorized wheels of this kind can have almost independent tangential, normal and angular velocities. Dynamical models for this kind of robots are not very common due to the difficulty in modeling the several internal frictions inside the wheels, making the model somewhat specific to the type of wheel being used (Williams et al., 2002).

Frequent mechanical configurations for omni-directional robots are based on three and four wheels. Three wheeled systems are mechanically simpler but robots with four wheels have more acceleration with the same kind of motors. Four wheeled robots are expected to have better effective floor traction, that is, less wheel slippage – assuming that all wheels are pressed against the floor equally. Of course four wheeled robots also have a higher costs in equipment, increased energy consumption and may require some kind of suspension to distribute forces equally among the wheels.

In order to study and compare the models of the 3 and 4 wheeled robots, a single prototype was built that can have both configurations, that is, the same mechanical platform can be used with 3 wheels and then it can be disassembled and reassembled with a 4



(a) Three wheeled robot. (b) Four wheeled robot.

Figure 1: Omni-directional robot.

wheel configuration, see figure 1.

Data from experimental runs is taken from overhead camera. The setup is taken from the heritage of the system described in (Costa et al., 2000) that currently features 25 fps, one centimeter accuracy in position (XX and YY axis) and about 3 sexagesimal degrees of accuracy in the heading of the robot.

In order to increase the performance of robots, there were some efforts on the studying their dynamical models (Campion et al., 1996)(Conceição et al., 2006)(Khosla, 1989)(Tahmasebi et al., 2005)(Williams et al., 2002) and kinematic models (Campion et al., 1996) (Leow et al., 2002)(Loh et al., 2003)(Muir and Neuman, 1987)(Xu et al., 2005). Models are based on linear and non linear dynamical systems and the estimation of parameters has been the subject of continuing research (Conceição et al., 2006)(Olsen and Petersen, 2001). Once the dynamical model is found, its parameters have to be estimated. The most common method for identification

of robot parameters are based on the Least Squares method and Instrumental Variables.

However, the systems are naturally non-linear (Julier and Uhlmann, 1997), the estimation of parameters is more complex and the existing methods (Ghaharamani and Roweis, 1999)(Gordon et al., 1993)(Tahmasebi et al., 2005) have to be adapted to the model's structure and noise.

1.1 Structure

This paper starts by presenting the mechanical prototype and studied mechanical configurations for the 3 and 4 wheeled robots. Finding the dynamical model is discussed and then, an initial approach in estimating model parameters for each robot is done. The need for additional accuracy drives the comparative study on relative importance of the estimated parameters. An additional experiment is done for estimating final numerical values for the configurations of 3 and 4 wheels. Conclusions and future work are also presented.

2 MECHANICAL CONFIGURATIONS

Figures 2 and 3 present the configuration of the three and four wheeled robots respectively, as well as all axis and relevant forces and velocities of the robotic system. The three wheeled system features wheels separated by 120 degrees.

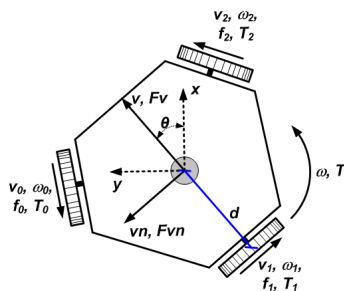


Figure 2: Three Wheeled Robot.

Figures 2 and 3 show the notation used throughout this paper, detailed as follows:

- x, y, θ - Robot's position (x, y) and θ angle to the defined front of robot;
- d [m] - Distance between wheels and center robot;
- v_0, v_1, v_2, v_3 [m/s] - Wheels linear velocity;
- $\omega_0, \omega_1, \omega_2, \omega_3$ [rad/s] - Wheels angular velocity;
- f_0, f_1, f_2, f_3 [N] - Wheels traction force;

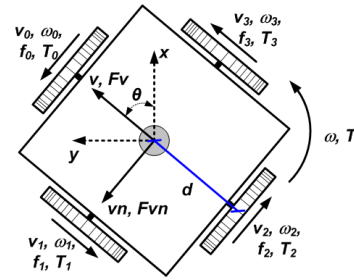


Figure 3: Four Wheeled Robot.

- T_0, T_1, T_2, T_3 [N · m] - Wheels traction torque;
- v, vn [m/s] - Robot linear velocity;
- ω [rad/s] - Robot angular velocity;
- F_v, F_{vn} [N] - Robot traction force along v and vn ;
- T [N · m] - Robot torque (respects to ω).

3 MODELS

3.1 Kinematic

The well known kinematic model of an omnidirectional robot located a (x, y, θ) can be written as $v_x(t) = dx(t)/dt$, $v_y(t) = dy(t)/dt$ and $\omega(t) = d\theta(t)/dt$ (please refer to figures 2 and 3 for notation issues). Equation 1 allows the transformation from linear velocities v_x and v_y on the static axis to linear velocities v and vn on the robot's axis.

$$X_R = \begin{bmatrix} v(t) \\ vn(t) \\ \omega(t) \end{bmatrix}; X_0 = \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix}$$

$$X_R = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot X_0 \quad (1)$$

3.1.1 Three Wheeled Robot

Wheel speeds v_0, v_1 and v_2 are related with robot's speeds v, vn and ω as described by equation 2.

$$\begin{bmatrix} v_0(t) \\ v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} -\sin(\pi/3) & \cos(\pi/3) & d \\ 0 & -1 & d \\ \sin(\pi/3) & \cos(\pi/3) & d \end{bmatrix} \cdot \begin{bmatrix} v(t) \\ vn(t) \\ \omega(t) \end{bmatrix} \quad (2)$$

Applying the inverse kinematics is possible to obtain the equations that determine the robot speeds related the wheels speed. Solving in order of v, vn and ω , the following can be found:

$$v(t) = (\sqrt{3}/3) \cdot (v_2(t) - v_0(t)) \quad (3)$$

$$vn(t) = (1/3) \cdot (v_2(t) + v_0(t)) - (2/3) \cdot v_1(t) \quad (4)$$

$$\omega(t) = (1/(3 \cdot d)) \cdot (v_0(t) + v_1(t) + v_2(t)) \quad (5)$$

3.1.2 Four Wheeled Robot

The relationship between the wheels speed v_0, v_1, v_2 and v_3 , with the robot speeds v, vn and ω is described by equation 6.

$$\begin{bmatrix} v_0(t) \\ v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & d \\ -1 & 0 & d \\ 0 & -1 & d \\ 1 & 0 & d \end{bmatrix} \cdot \begin{bmatrix} v(t) \\ vn(t) \\ \omega(t) \end{bmatrix} \quad (6)$$

It is possible to obtain the equations that determine the robot speeds related with wheels speed but the matrix associated with equation 6 is not square. This is because the system is redundant. It can be found that:

$$v(t) = (1/2) \cdot (v_3(t) - v_1(t)) \quad (7)$$

$$vn(t) = (1/2) \cdot (v_0(t) - v_2(t)) \quad (8)$$

$$\omega(t) = (v_0(t) + v_1(t) + v_2(t) + v_3(t)) / (4 \cdot d) \quad (9)$$

3.2 Dynamic

The dynamical equations relative to the accelerations can be described in the following relations:

$$M \cdot \frac{dv(t)}{dt} = \sum F_v(t) - F_{Bv}(t) - F_{Cv}(t) \quad (10)$$

$$M \cdot \frac{dvn(t)}{dt} = \sum F_{vn}(t) - F_{Bvn}(t) - F_{Cvn}(t) \quad (11)$$

$$J \cdot \frac{d\omega(t)}{dt} = \sum T(t) - T_{B\omega}(t) - T_{C\omega}(t) \quad (12)$$

where the following parameters relate to the robot as follows:

- M [kg] - mass;
- J [$kg \cdot m^2$] - inertia moment;
- F_{Bv}, F_{Bvn} [N] - viscous friction forces along v and vn ;
- $T_{B\omega}$ [$N \cdot m$] - viscous friction torque with respect to the robot's rotation axis;
- F_{Cv}, F_{Cvn} [N] - Coulomb frictions forces along v and vn ;
- $T_{C\omega}$ [$N \cdot m$] - Coulomb friction torque with respect to robot's rotation axis.

Viscous friction forces are proportional to robot's speed and as such $F_{Bv}(t) = B_v \cdot v(t)$, $F_{Bvn}(t) = B_{vn} \cdot vn(t)$ and $T_{B\omega}(t) = B_\omega \cdot \omega(t)$, where B_v, B_{vn} [$N/(m/s)$] are the viscous friction coefficients for directions v and vn and B_ω [$N \cdot m/(rad/s)$] is the viscous friction coefficient to ω .

The Coulomb friction forces are constant in amplitude $F_{Cv}(t) = C_v \cdot \text{sign}(v(t))$, $F_{Cvn}(t) = C_{vn} \cdot \text{sign}(vn(t))$ and $T_{C\omega}(t) = C_\omega \cdot \text{sign} \omega(t)$, where C_v, C_{vn} [N] are Coulomb friction coefficient for directions v e vn and C_ω [$N \cdot m$] is the Coulomb friction coefficient for ω .

3.2.1 Three Wheeled Robot

The relationship between the traction forces and rotation torque of the robot with the traction forces on the wheels is described by the following equations:

$$\sum F_v(t) = (f_2(t) - f_0(t)) \cdot \sin(\pi/3) \quad (13)$$

$$\sum F_{vn}(t) = -f_1(t) + (f_2(t) + f_0(t)) \cdot \cos(\pi/3) \quad (14)$$

$$\sum T(t) = (f_0(t) + f_1(t) + f_2(t)) \cdot d \quad (15)$$

The traction force on each wheel is estimated by traction torque, which can be determined using the motor current, as described in the following equations:

$$f_j(t) = T_j(t)/r \quad (16)$$

$$T_j(t) = l \cdot K_t \cdot i_j(t) \quad (17)$$

- l - Gearbox reduction;
- r [m] - Wheel radius;
- K_t [$N \cdot m/A$] - Motor torque constant;
- i_j [A] - Motor current (j=motor number).

3.2.2 Four Wheeled Robot

The relationship between the traction forces and rotation torque of the robot with the traction forces on the wheels, is described by the following equations:

$$\sum F_v(t) = f_3(t) - f_1(t) \quad (18)$$

$$\sum F_{vn}(t) = f_0(t) - f_2(t) \quad (19)$$

$$\sum T(t) = (f_0(t) + f_1(t) + f_2(t) + f_3(t)) \cdot d \quad (20)$$

As above, the traction force in each wheel is estimated using the wheels traction torque, which is determined by the motor current, using equations 16 and 17, where $j=0,1,2,3$.

3.3 Motor

The prototype uses brushless motors for the locomotion of the robot. The model for brushless motors is the similar to the common DC motors, based on (Pillay and Krishnan, 1989).

$$u_j(t) = L \cdot \frac{di_j(t)}{dt} + R \cdot i_j(t) + K_v \cdot \omega_{mj}(t) \quad (21)$$

$$T_{mj}(t) = K_t \cdot i_j(t) \quad (22)$$

- L [H] - Motor inductance;
- R [Ω] - Motor resistor;
- K_v [$V/(rad/s)$] - EMF motor constant;
- u_j [V] - Motor voltage (j=motor number);
- ω_{mj} [rad/s] - Motor angular velocity (j=motor number);
- T_{mj} [$N \cdot m$] - Motor torque (j=motor number).

4 PARAMETER ESTIMATION

The necessary variables to estimate the model parameters are motor current, robot position and velocity. Currents are measured by the drive electronics, position is measured by using external camera and velocities are estimated from positions.

The parameters that must be identified are the viscous friction coefficients (B_v, B_{vn}, B_ω), the Coulomb friction coefficients (C_v, C_{vn}, C_ω) and inertia moment J . The robot mass was measured, and it was 1.944 kg for the three wheeled robot and 2.34 kg for the four wheeled robot.

4.1 Experience 1 - Steady State Velocity

This method permits to identify the viscous friction coefficients B_ω and the Coulomb friction coefficients C_ω . The estimation of the coefficient ω was only implemented because inertia moment is unknown, and it is necessary to have an initial estimate of these coefficients. The experimental method relies on applying different voltages to the motors in order to move the robot according his rotation axis - the tests were made for positive velocities. Once reached the steady state, the robot's speed ω and rotation torque T can be measured. The robot speed is constant, so, the acceleration is null, and as such equation 12 can be re-written as follows:

$$\sum T(t) = B_\omega \cdot \omega(t) + C_\omega \quad (23)$$

This linear equation shows that it is possible to test different values of rotation speed and rotation torques in multiple experiences and estimate the parameters.

4.2 Experience 2 - Null Traction Forces

This method allows for the estimation of the viscous friction coefficients (B_v, B_{vn}), the Coulomb friction coefficients (C_v, C_{vn}) and the inertia moment J . The experimental method consists in measuring the robot acceleration and speed when the traction forces were null. The motor connectors were disconnected and with a manual movement starting from a stable position, the robot was pushed through the directions v , vn and rotated according to his rotation axis. During the subsequent deceleration, velocity and acceleration were measured. Because the traction forces were null during the deceleration equations 10, 11, and 12 can be re-written as follows:

$$\frac{dv(t)}{dt} = -\frac{B_v}{M} \cdot v(t) - \frac{C_v}{M} \quad (24)$$

$$\frac{dvn(t)}{dt} = -\frac{B_{vn}}{M} \cdot vn(t) - \frac{C_{vn}}{M} \quad (25)$$

$$\frac{d\omega(t)}{dt} = -\frac{B_\omega}{J} \cdot \omega(t) - \frac{C_\omega}{J} \quad (26)$$

These equations are also a linear relation and estimation of all parameters is possible.

The inertia moment J is estimated using the values obtained previously in section 4.1. To do this, equation 26 must be solved in order of J :

$$J = -\frac{\omega(t)}{(d\omega(t)/dt)} \cdot B_\omega - \frac{1}{(d\omega(t)/dt)} \cdot C_\omega \quad (27)$$

4.3 DC Motor Parameters

The previous electrical motor model (equation 21) includes an electrical pole and a much slower, dominant mechanical pole - thus making inductance L value negligible. To determinate the relevant parameters K_v and R , a constant voltage is applied to the motor. Under steady state condition, the motor's current and the robot's angular velocity are measured. The tests are repeated several times for the same voltage, changing the operation point of the motor, by changing the friction on the motor axis.

In steady state, the inductance L disappears of the equation 21, being rewritten as follows:

$$u_j(t) = R \cdot i_j(t) + K_v \cdot \omega_{mj}(t) \quad (28)$$

As seen in equation 29, by dividing (28) by $i_j(t)$, a linear relation is obtained and thus estimation is possible.

$$\frac{u_j(t)}{i_j(t)} = K_v \cdot \frac{\omega_{mj}(t)}{i_j(t)} + R \quad (29)$$

5 RESULTS

5.1 Robot Model

By combining previously mentioned equations, it is possible to show that model equations can be rearranged into a variation of the state space that can be described as:

$$(dx(t)/dt) = A \cdot x(t) + B \cdot u(t) + K \cdot \text{sign}(x) \quad (30)$$

$$x(t) = [v(t) \quad vn(t) \quad w(t)]^T \quad (31)$$

This formulation is interesting because it shows exactly which part of the system is non non-linear.

5.1.1 Three Wheeled

Using equations on section 3.2, 13 to 17 and 28, the equations for the three wheeled robot model are:

$$A = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix} \quad (32)$$

$$A_{11} = -\frac{3 \cdot K_t^2 \cdot l^2}{2 \cdot r^2 \cdot R \cdot M} - \frac{B_v}{M}$$

$$A_{22} = -\frac{3 \cdot K_t^2 \cdot l^2}{2 \cdot r^2 \cdot R \cdot M} - \frac{B_{vn}}{M}$$

$$A_{33} = -\frac{3 \cdot d^2 \cdot K_t^2 \cdot l^2}{r^2 \cdot R \cdot J} - \frac{B_w}{J}$$

$$B = \frac{l \cdot K_t}{r \cdot R} \cdot \begin{bmatrix} -\sqrt{3}/(2 \cdot M) & 0 & \sqrt{3}/(2 \cdot M) \\ 1/(2 \cdot M) & 1/M & 1/(2 \cdot M) \\ d/J & d/J & d/J \end{bmatrix} \quad (33)$$

$$K = \begin{bmatrix} -C_v/M & 0 & 0 \\ 0 & -C_{vn}/M & 0 \\ 0 & 0 & -C_w/J \end{bmatrix} \quad (34)$$

5.1.2 Four Wheeled

Using equations on section 3.2 and equations 16 to 20 and 28 we get the following equations to the four wheeled robot model.

$$A = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix} \quad (35)$$

$$A_{11} = -\frac{2 \cdot K_t^2 \cdot l^2}{r^2 \cdot R \cdot M} - \frac{B_v}{M}$$

$$A_{22} = -\frac{2 \cdot K_t^2 \cdot l^2}{r^2 \cdot R \cdot M} - \frac{B_{vn}}{M}$$

$$A_{33} = -\frac{4 \cdot d^2 \cdot K_t^2 \cdot l^2}{r^2 \cdot R \cdot J} - \frac{B_w}{J}$$

$$B = \frac{l \cdot K_t}{r \cdot R} \cdot \begin{bmatrix} 0 & -1/M & 0 & 1/M \\ 1/M & 0 & -1/M & 0 \\ d/J & d/J & d/J & d/J \end{bmatrix} \quad (36)$$

$$K = \begin{bmatrix} -C_v/M & 0 & 0 \\ 0 & -C_{vn}/M & 0 \\ 0 & 0 & -C_w/J \end{bmatrix} \quad (37)$$

5.2 Experimental Data for Robot Model

Experience 1 was conducted using an input signal corresponding to a ramped up step. This way wheel slipping was avoided, that is, wheel - traction problems don't exist.

Shown in Figure 4 are the experimental plots regarding the 4 wheeled system. Due to space constraints only T vs. ω and results of the experiment 2 along the v direction are shown.

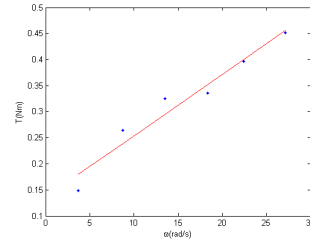
The motor model was presented earlier in equation 21.

Experimental tests to the four motors were made to estimate the value of resistor R and the constant K_v . The numerical value of the torque constant K_t is identical to the EMF motor constant K_v .

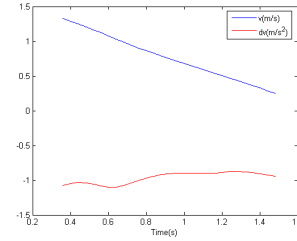
Figure 5 plots experimental runs regarding motor 0. Other motors follow similar behavior.

5.3 Numerical Results

Table 1 presents the experimental results to the friction coefficients and inertial moment. From the experimental runs from all 4 motors, the parameters found are $K_v = 0.0259 \text{ V}/(\text{rad/s})$ and $R = 3.7007 \Omega$.



(a) Experience 1 - T and ω .



(b) Experience 2 - Direction v .

Figure 4: Experimental Results for the four wheeled robot.

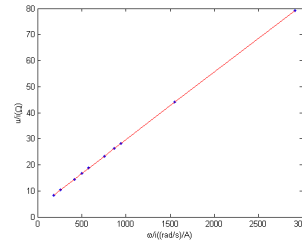


Figure 5: Experimental tests for motor 0.

Table 1: Friction coefficients and inertia moment.

Parameters	3 wheels	4 wheels
$J(\text{kg} \cdot \text{m}^2)$	0.015	0.016
$B_v(\text{N}/(\text{m/s}))$	0.503	0.477
$B_{vn}(\text{N}/(\text{m/s}))$	0.516	0.600
$B_\omega \text{N} \cdot \text{m}/(\text{rad/s})$	0.011	0.011
$C_v(\text{N})$	1.906	1.873
$C_{vn}(\text{N})$	2.042	2.219
$C_\omega(\text{N} \cdot \text{m})$	0.113	0.135

5.4 Sensitivity Analysis

To understand which model parameters have more influence on the robot's dynamics, a comparison was made between the matrices of the models.

The model equation 30 is a sum of fractions. Analyzing the contribution of each parcel and of the variable portion within each fraction, a sensitivity analysis is performed, one estimated parameter at a time.

1. Matrix A , robot moving along v direction;

- Three wheeled robot:

$$\left(\frac{3 \cdot K_t^2 \cdot l^2}{2 \cdot r^2 \cdot R \cdot M} \right) = \frac{K_{a1}}{R} = 3.3110$$

$$(B_v/M) = K_{a2} \cdot B_v = 0.3245$$

- Four wheeled robot:

$$\left(\frac{2 \cdot K_t^2 \cdot l^2}{r^2 \cdot R \cdot M} \right) = \frac{K_{a1}}{R} = 3.6676$$

$$(B_v/M) = K_{a2} \cdot B_v = 0.2041$$

 2. Matrices B and K , robot moving along v direction with constant voltage motor equal to 6V;

- Three wheeled robot:

$$\left(\frac{\sqrt{3} \cdot l \cdot K_t}{2 \cdot r \cdot R \cdot M} \right) \cdot 12 = \frac{K_b}{R} \cdot 12 = 5.7570$$

$$(C_v/M) = K_k \cdot C_v = 0.8728$$

- Four wheeled robot:

$$\left(\frac{l \cdot K_t}{r \cdot R \cdot M} \right) \cdot 12 = \frac{K_b}{R} \cdot 12 = 5.5227$$

$$(C_v/M) = K_k \cdot C_v = 0.7879$$

The same kind of analysis could be taken further by analyzing other velocities (vn and ω). Conclusions reaffirm that motor parameters have more influence in the dynamics than friction coefficients. This means that it is very important to have an accurate estimation of the motor parameters. Some additional experiences were designed to improve accuracy. The method used previously does not offer sufficient accuracy to the estimation of R . This parameter R is not a physical parameter and includes a portion of the non-linearity of the H bridge powering the circuit that, in turn, feeds 3 rapidly switching phases of the brushless motors used. In conclusion, additional accuracy in estimating R is needed.

5.5 Experience 3 - Parameter Estimation Improvement

The parameter improving experience was made using a step voltage with an initial acceleration ramp.

As seen in 5.1 the model was defined by the equation 30 and we can improve the quality of the estimation by using the Least Squares method. The system model equation can be rewritten as:

$$y = \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \theta_3 \cdot x_3 \quad (38)$$

Where $x_1 = x(t)$, $x_2 = u(t)$, $x_3 = 1$ and $y = dx(t)/dt$. The parameters θ are estimated using:

$$\theta = (x^T \cdot x)^{-1} \cdot x^T \cdot y \quad (39)$$

$$x = [x_1(1) \dots x_1(n) \ x_2(1) \dots x_2(n) \ x_3(1) \dots x_3(n)]^T \quad (40)$$

Estimated parameters can be skewed and for this reason instrumental variables are used to minimize the error, with vector of states defined as

$$z = [\bar{x}_1(1) \dots \bar{x}_1(n) \ x_2(1) \dots x_2(n) \ x_3(1) \dots x_3(n)]^T \quad (41)$$

The parameters θ are now calculated by:

$$\theta = (z^T \cdot z)^{-1} \cdot z^T \cdot y \quad (42)$$

Three experiments were made for each configuration of 3 and 4 wheels, along v , vn and ω . For the v and vn experiments values C_v and C_{vn} are kept from previous analysis. For the ω experiment, the value of the R parameter used is the already improved version from previous v and vn experimental runs of the current section.

The numerical value of R for each motor was estimated for each motor and then averaged to find $R=4.3111 \ \Omega$. The results are present on followings tables. Table 2 shows values estimated by the experiment mentioned in this section.

Table 2: Parameters estimated using the method 3.

Parameters	3 wheels	4 wheels
$J(kg \cdot m^2)$	0.0187	0.0288
$B_v(N/(m/s))$	0.5134	0.5181
$B_{vn}(N/(m/s))$	0.4571	0.7518
$B_\omega N \cdot m/(rad/s)$	0.0150	0.0165
$C_\omega(N \cdot m)$	0.0812	0.1411

The final values for friction and inertial coefficients are averaged with results from all 3 experimental methods and the numerical values found are presented in Table 3.

Table 3: Parameters of dynamical models.

Parameters	3 wheels	4 wheels
$d(m)$	0.089	
$r(m)$	0.0325	
l	5	
$K_v(V/(rad/s))$	0.0259	
$R(\Omega)$	4.3111	
$M(kg)$	1.944	2.34
$J(kg \cdot m^2)$	0.0169	0.0228
$B_v(N/(m/s))$	0.5082	0.4978
$B_{vn}(N/(m/s))$	0.4870	0.6763
$B_\omega(N \cdot m/(rad/s))$	0.0130	0.0141
$C_v(N)$	1.9068	1.8738
$C_{vn}(N)$	2.0423	2.2198
$C_\omega(N \cdot m)$	0.0971	0.1385

5.6 Model Validation Experiences

The models were validated with experimental tests on using a step voltage with an initial acceleration ramp.

Due to space constraints in this document, figures 6 to 9 show plots for some of the runs only. Other runs confirm the global validity of the model as simulation follows reality closely.

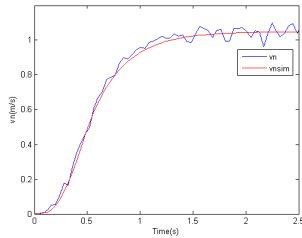


Figure 6: 3 wheel model validation - velocity along v_n .

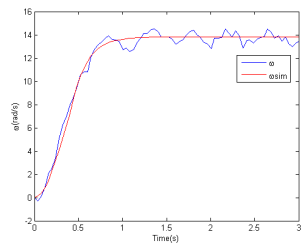


Figure 7: 3 wheel model validation - angular velocity ω .

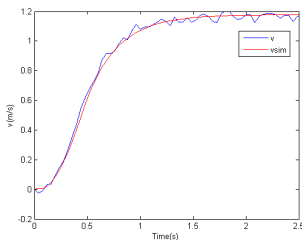


Figure 8: 4 wheel model validation - velocity along v .

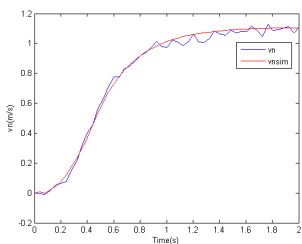


Figure 9: 4 wheel model validation - velocity along v_n .

6 CONCLUSIONS

This paper presents models for mobile omni-directional robots with 3 and 4 wheels. The derived model is non-linear but maintains some similarities with linear state space equations. Friction coefficients are most likely dependent on robot and wheels construction and also on the weight of the robot. The model is derived assuming no wheel slip as in most standard robotic applications.

A prototype that can have either 3 or 4 omni-directional wheels was used to validate the presented model. The test ground is smooth and carpeted. Experience data was gathered by overhead camera capable of determining position and orientation of the robot with good accuracy.

Experiences were made to estimate the parameters of the model for the prototypes. The accuracy of the presented model is discussed and the need for additional experiences is proved. The initial estimation method used two experiences to find all parameters but a third experience is needed to improve the accuracy of the most important model parameters. Sensitivity analysis shows that the most important model parameters concern motor constants.

Observing estimated model parameters, the four wheel robot has higher friction coefficients in the v_n direction when compared to the v direction. This means of course higher maximum speed for movement along v axis and higher power consumption for movements along the v_n direction. This difference in performance points to the need of mechanical suspension to even wheel pressure on the ground.

The found model was shown to be adequate for the prototypes in the several shown experimental runs.

7 FUTURE WORK

The work presented is part of a larger study. Future work will include further tests with different prototypes including prototypes with suspension. The model can also be enlarged to include the limits for slippage and movement with controlled slip for the purpose of studying traction problems. Dynamical models estimated in this work can be used to study the limitations of the mechanical configuration and allow for future enhancements both at controller and mechanical configuration level. This study will enable effective full comparison of 3 and 4 wheeled systems.

REFERENCES

- Campion, G., Bastin, G., and Dandrea-Novel, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1):47–62. 1042-296X.
- Conceição, A. S., Moreira, A. P., and Costa, P. J. (2006). Model identification of a four wheeled omnidirectional mobile robot. In *Controlo 2006, 7th Portuguese Conference on Automatic Control*, Instituto Superior Técnico, Lisboa, Portugal.
- Costa, P., Marques, P., Moreira, A. P., Sousa, A., and Costa, P. (2000). Tracking and identifying in real time the robots of a f-180 team. In *Manuela Veloso, Enrico Pagello and Hiroaki Kitano, Robocup-99: Robot Soccer World Cup III. Springer, LNAI*, pages 289–291.
- Diegel, O., Badve, A., Bright, G., Potgieter, and Tlale, S. (2002). Improved mecanum wheel design for omnidirectional robots. In *Proc. 2002 Australasian Conference on Robotics and Automation*, Auckland.
- Ghaharamani, Z. and Roweis, S. T. (1999). Learning nonlinear dynamical systems using an em algorithm. In *M. S. Kearns, S. A. Solla, D. A. Cohn, (eds) Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press*, 11.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F on Radar and Signal Processing*, 140(2):107–113. 0956-375X.
- Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the kalman filter to nonlinear systems. *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*.
- Khosla, P. K. (1989). Categorization of parameters in the dynamic robot model. *IEEE Transactions on Robotics and Automation*, 5(3):261–268. 1042-296X.
- Leow, Y. P., H., L. K., and K., L. W. (2002). Kinematic modelling and analysis of mobile robots with omnidirectional wheels. In *Seventh International Conference on Control, Automation, Robotics And Vision (ICARCV'02)*, Singapore.
- Loh, W. K., Low, K. H., and Leow, Y. P. (2003). Mechatronics design and kinematic modelling of a singularityless omnidirectional wheeled mobile robot. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 3, pages 3237–3242.
- Muir, P. and Neuman, C. (1987). Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In *Proceedings 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 1772–1778.
- Olsen, M. M. and Petersen, H. G. (2001). A new method for estimating parameters of a dynamic robot model. *IEEE Transactions on Robotics and Automation*, 17(1):95–100. 1042-296X.
- Pillay, P. and Krishnan, R. (1989). Modeling, simulation, and analysis of permanent-magnet motor drives, part 11: The brushless dc motor drive. *IEEE transactions on Industry applications*, 25(2):274–279.
- Salih, J., Rizon, M., Yaacob, S., Adom, A., and Mamat, M. (2006). Designing omni-directional mobile robot with mecanum wheel. *American Journal of Applied Sciences*, 3(5):1831–1835.
- Tahmasebi, A. M., Taati, B., Mobasser, F., and Hashtrudi-Zaad, K. (2005). Dynamic parameter identification and analysis of a phantom haptic device. In *Proceedings of 2005 IEEE Conference on Control Applications*, pages 1251–1256.
- Williams, R. L., I., Carter, B. E., Gallina, P., and Rosati, G. (2002). Dynamic model with slip for wheeled omnidirectional robots. *IEEE Transactions on Robotics and Automation*, 18(3):285–293. 1042-296X.
- Xu, J., Zhang, M., and Zhang, J. (2005). Kinematic model identification of autonomous mobile robot using dynamical recurrent neural networks. In *2005 IEEE International Conference Mechatronics and Automation*, volume 3, pages 1447–1450.

POSITION CONTROL METHOD OF A NON-CONTACTING CONVEYANCE SYSTEM FOR STEEL STRIP

Yeongseob Kueon, Hyoung Jin Yoon[†]

*Process Control and Rolling Technology Research Gr., POSCO Tech. Res. Lab., S. Korea
yskueon@posco.co.kr*

Yoon Su Baek[†]

*[†]Department of Mechanical Engineering, Yonsei University, Seoul, S. Korea
amuger@yonsei.ac.kr*

Keywords: Electromagnetic force, non-contacting conveyance, steel strip, position control.

Abstract: Electromagnetic application system to improve the surface quality of steel strip is getting popular because customers demand better surface quality of steel strip. To realize such a requirement, non-contact conveyance methods by means of air floater and electromagnetic levitation and propulsion were considered. However, air floating method is not easy to control of the position of steel strip since the system is highly nonlinear. And thus, the application of a magnetic levitation and propulsion to steel strip conveyance is suitable. Sensors measuring positions of steel strip also need to be non-contact in order to maintain non-contact and simple characteristics of the system. This paper proposes the method of the spatial position estimation of steel plate without using sensors. This method simplifies non-contact conveyance system and cuts down expenses of the system. Spatial positions of steel strip can be estimated by currents supplied for electromagnet to maintain a fixed air gap. Estimated positions are then fed back into the control system to do position control. Computer simulation and experimental results are provided to verify the suitability of the proposed system performance and concept.

1 INTRODUCTION

In 1990s, various kinds of research activities were conducted to convey steel strips by means of non-contacting methods. One of them was electromagnetic conveyance technology. At first, most researchers were focused on reducing vibration of steel strip. (Liu and Yao, 2002). Later on, University of Tokyo conducted very promising research of levitation and propulsion of steel strip via electromagnetic force. (Hayashiya, and et al, 1999).

Steel making industries produce and treat large amounts of thin steel strips in cold rolling processes to obtain high quality steel strips using various ways. Actually, steel strips are processed at high speed in continuous cold rolling process lines. Because of this, vibration and position deviation of steel strip are the main hazardous problems which cause surface defects and lower productivity. The non-contact operation and the quick response mechanism can be considered to solve the above mentioned

problems. Applications of electromagnetic force can be one of the useful technical approaches. (Liu and Yao, 2002).

Electromagnetic application system to improve the surface quality of steel strip is getting popular because customers demand better surface quality of steel strip. To realize such a requirement, non-contact conveyance methods by means of air floater and electromagnetic levitation and propulsion were considered. However, air floating method is not easy to control of the position of steel strip since the system is highly nonlinear. And thus, the application of a magnetic levitation and propulsion to steel strip conveyance is suitable. Sensors measuring positions of steel strip also need to be non-contact in order to maintain non-contact and simple characteristics of the system. This paper proposes the method of the spatial position estimation of steel plate without using sensors. This method simplifies non-contact conveyance system and cuts down expenses of the system. Spatial positions of steel strip can be estimated by currents supplied for electromagnet to

maintain a fixed air gap. Estimated positions are then fed back into the control system to do position control. Computer simulation and experimental results are provided to verify the suitability of the proposed system performance and concept. (Gerber, 2002)

2 BACKGROUND

Non-contacting conveyance system by means of electromagnetic force can be seen in Figure 1. The system should generate normal, thrust, and guidance forces in order to maintain steel strip under control. The system is now then designed based on the above concept.

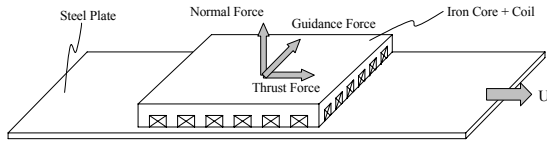


Figure 1: Non-Contacting System Concept.

Figure 2 shows the schematic diagram of the non-contact conveyance system. $N \cdot i$ shows the magnetomotive force and Φ means magnetic flux. Reluctance, \mathfrak{R} , can be a clearance at the middle of E-shaped core. Reluctance, \mathfrak{R} , can be expressed as equation (1) in terms of the area, A , at the middle of E-shaped core. (Roters, 1951).

$$\mathfrak{R} = \frac{z}{\mu_0 A} \quad (1)$$

where μ_0 is the permeability in the air which can be set as $4\pi \times 10^{-7}$ H/m.

As shown in Figure 1, reluctance on both sides can be twice as much as the center part of E-shaped core, since the area on both sides is half of the middle part. The equivalent reluctance, \mathfrak{R}_{eq} , can be obtained as shown in equation (2).

$$\mathfrak{R}_{eq} = \mathfrak{R} + \frac{1}{\frac{1}{2\mathfrak{R}} + \frac{1}{2\mathfrak{R}}} = 2\mathfrak{R} \quad (2)$$

From equation (2) and Ohm's law, applied magnetic field, \mathfrak{I} , can be expressed by equation (3),

$$\mathfrak{I} = \int \mathbf{H} \cdot d\mathbf{l} = N \cdot i \quad (3)$$

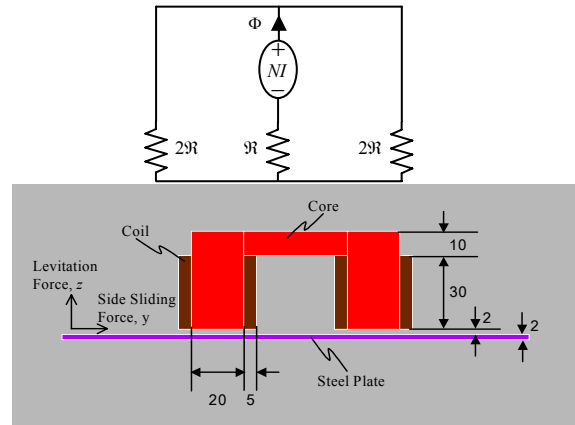


Figure 2: Schematic Diagram of the System.

Where N and i shows number of turns and applied current to the coil. Magnetic flux can be computed by using applied magnetic flux \mathfrak{I} and equivalent reluctance \mathfrak{R}_{eq} with Ohm's law as follows (Trumper, Weng, Ritter, 1999),

$$\Phi = \frac{\mathfrak{I}}{\mathfrak{R}_{eq}} = \frac{Ni}{\mathfrak{R}_{eq}} \quad (4)$$

Hence, flux linkage with N turns of coils can be expressed as shown in equation (5).

$$\lambda = N\Phi = \frac{N^2 i}{\mathfrak{R}_{eq}} \quad (5)$$

Applied current, i , to the electromagnet can induce some amount of force to the steel strip. The induced force to the steel strip can be expressed by magnetic force, f_e .

$$f_e = \frac{\partial W_c}{\partial z} \quad (6)$$

Where magnetic energy, W_c , can be obtained as follows,

$$W_c = \int_0^i \lambda dt \quad (7)$$

Equation (7) can be obtained by integrating equation (5) with respect to time and differentiate partially with respect to the moving direction. f_e is induced force with respect to applied current i and clearance z .

$$f_e = -\frac{\mu_0 AN^2}{4} \cdot \left(\frac{i}{z}\right)^2 \quad (8)$$

Induced force, f_e , can be negative when the electromagnet attracts the thin steel strip. In other words, attractive force caused by the electromagnet can be expressed by negative force comparing with repulsive force (Choi and Baek 2002).

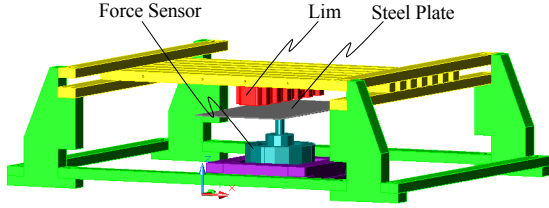


Figure 3: Configuration of the System.

3 SYSTEM DYNAMICS

This paper assumes that the steel strip should keep constant clearance and perpendicular position during its movement. The position estimation of the steel strip cannot be estimated correctly since current applied to the electromagnet is changed according to the inclined steel strip. In this paper, two assumptions were used to derive the equation of motion between the steel strip and the electromagnet. They are as follows: one is that the clearance is always constant, and the other is that the steel strip is always perpendicular to its moving direction.

The initial condition of the system is satisfied when $a=0$. F_A is same as F_B and applied current has also the same amount at the initial condition. If a is not zero, F_A and F_B are not the same and applied current is also not the same, since the distance between the electromagnet and the center of gravity of the steel strip is changed and the force cannot be balanced any more. In other words, the forces applied to the steel strip from the electromagnet should be changed to keep the steel strip perpendicular in accordance with moving distance. The moving distance or the position of the steel strip can be estimated by the above mentioned things. (Nasar and Boldea, 1976).

Figure 4 shows that the steel strip has moved to the amount of a from the initial position toward x-direction.

F_A and F_B means the attractive forces to the points A and B, respectively. In this case, following two equations can be derived from the force and moment balance equations.

$$\sum F_z = 0 ; F_A + F_B = F_{steel} \quad (9)$$

$$\sum M_y = 0 ; F_B(n+a) - F_A(n-a) = 0 \quad (10)$$

The above two equations can be expressed in equation (11), where F_{steel} is steel strip weight, M_{steel} steel strip mass, and g the acceleration of gravity.

$$F_{steel} = M_{steel} \times g \quad (11)$$

From equations (9) and (10), the following equation can be derived with respect to a ,

$$a = q \cdot \frac{F_B - F_A}{F_{steel}} \quad (12)$$

The equations about the applied current to each coil can be derived by using equation (13),

$$F_n = \frac{\mu_0 AN^2}{4} \left(\frac{i_n}{z}\right)^2 \quad (n = A, B) \quad (13)$$

By substituting equation (13) into equation (12), the moving position a can be expressed with respect to the applied current as shown in equation (13),

$$a = \frac{\mu_0 AN^2 q}{4} \cdot \frac{i_B^2 - i_A^2}{F_{steel}} \quad (14)$$

or the following equation can be derived from equation (8),

$$a = q \cdot \frac{i_B^2 - i_A^2}{i_B^2 + i_A^2} \quad (15)$$

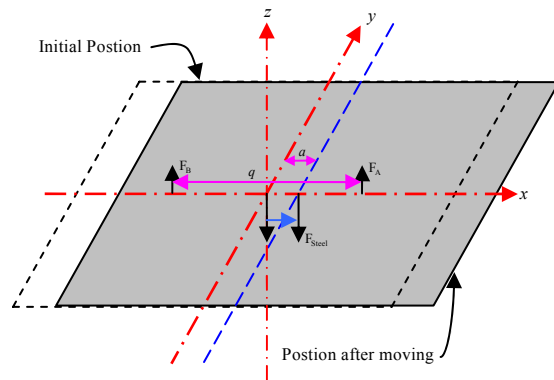


Figure 4: Movement of the Strip.

4 CONTROL SYSTEM

Levitation force (Normal force) can be described by y and i_{pc} , where y is the air gap between electromagnetic core and steel strip and i_{pc} is DC offset, respectively. Thrust force can be expressed by equation (16) and (17), by assuming that the frequency of power is fixed and AC power alone takes charge of control. (Fujisaki, 2001).

$$F_y = f(y, i_{DC}) \quad (16)$$

$$F_x = f(i_{AC}) \quad (17)$$

where, i_{AC} is the maximum value of AC current.

Equations of motion can be obtained by the following:

$$M\ddot{x} = F_x - K_{dx}\dot{x} \quad (18)$$

$$M\ddot{y} = F_y - Mg - K_{dy}\dot{y}$$

Where, M is the mass of the strip, F_x is thrust force, g is gravity, K_{dx} is the friction coefficient in the x direction, K_{dy} is the friction coefficient in the y direction.

From equations of motion, the following equation can be obtained.

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \mathbf{d} \quad (19)$$

where,

$$\mathbf{x} = \begin{bmatrix} x \\ v_x \\ y \\ v_y \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} F_x \\ F_y \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -g \end{bmatrix},$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{K_{dx}}{M} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{K_{dy}}{M} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{1}{M} & 0 \\ 0 & 0 \\ 0 & \frac{1}{M} \end{bmatrix} \quad (20)$$

Control inputs can be derived by the following equations:

$$F_x = K_{xp}(x_d - x) + K_{xD}(v_{xd} - v_x) + K_{xI} \int (x_d - x)dt + M_x a_{xd} + K_{dx} v_{xd} \quad (21)$$

$$F_y = K_{yp}(y_d - y) + K_{yD}(v_{yd} - v_y) + K_{yI} \int (y_d - y)dt + Mg + M_y a_{yd} + K_{dy} v_{yd} \quad (22)$$

where, v_x and v_y are velocities in the direction of x and y, respectively. Feedback gains are as follows: (Choi and Baek, 2002)

$$K_P = \begin{bmatrix} K_{xP} & 0 & 0 & 0 \\ 0 & 0 & K_{yP} & 0 \end{bmatrix},$$

$$K_I = \begin{bmatrix} K_{xI} & 0 & 0 & 0 \\ 0 & 0 & K_{yI} & 0 \end{bmatrix},$$

and

$$K_D = \begin{bmatrix} K_{xD} & 0 & 0 & 0 \\ 0 & 0 & K_{yD} & 0 \end{bmatrix} \quad (23)$$

Now, currents can be applied as follows:

$$I_A = F_y + F_x \cdot a \cdot \cos(2\pi ft + 0) \quad (24)$$

$$I_B = F_y + F_x \cdot a \cdot \cos\left(2\pi ft + \frac{\pi}{3}\right) \quad (25)$$

$$I_C = F_y + F_x \cdot a \cdot \cos\left(2\pi ft + \frac{2\pi}{3}\right) \quad (26)$$

where, a is AC magnitude weighting factor.

The overall control system can be designed as shown in figure 5.

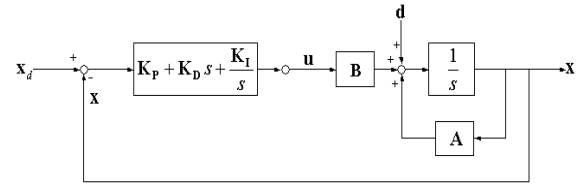


Figure 5: Control System Diagram.

5 POSITION ESTIMATION OF STEEL STRIP

The center of gravity of the steel strip is positioned in the middle of two equal spaced electromagnets. In this experiment, the steel strip is moving to the x-axis while it is lifted. Figures 6 through 8 show the experimental results. Figure 6 shows that the lifted steel strip can keep the constant clearance and stable. Figure 7 depicts the fluctuating current during the steel strip movements to the x-axis. Figure 8 is the compared positions of the steel strip as the steel strip moves to the x-axis. These graphs are the estimated position, the measured and filtered value by laser position sensor. (Nakagawa, 2000).

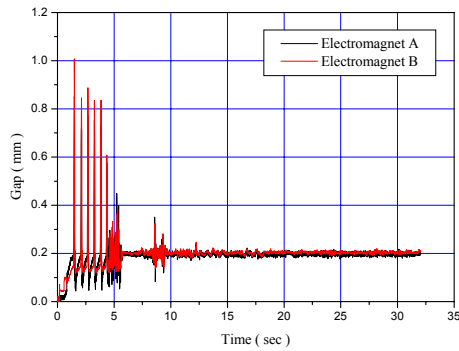


Figure 6: Air gap.

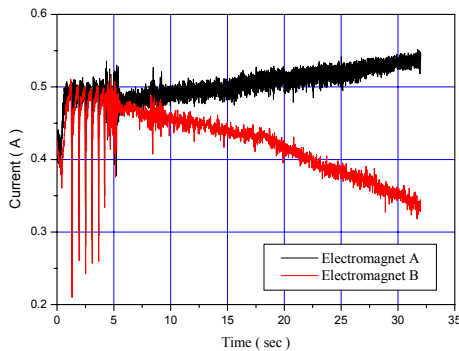


Figure 7: Current.

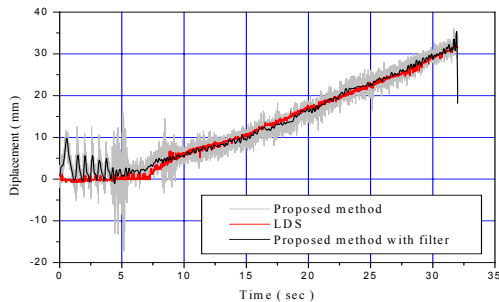


Figure 8: Position of a Moving Steel Strip.

6 RESULTS

A non-contact conveyance of the steel strip by electromagnets has been proposed to show that the applied current can be changed in accordance with the movement of the levitated steel strip. The position estimation method of the steel strip by the applied current has also been proposed and tested in the constructed non-contact steel strip conveyance system. The estimated position of the steel strip has been compared the measured one by a laser position sensor. The estimated position of the

steel strip shows satisfactory results comparing with the measured one. Non-contact sensors are very expensive and some of them make system complicated. To eliminate sensors, this paper proposes the method of the spatial position estimation of steel strip without sensors. This method simplifies non-contact conveyance system and cuts down expenses. The spatial position of steel strip with currents supplied for electromagnet was estimated and used to maintain a fixed air gap. And the theoretical analysis was verified by experiments and shows good control performance.

REFERENCES

- H. Hayashiya, N. Araki, J. E. Paddison, H. Ohsaki, and E. Masada, 1996, IEEE Trans. Magn., Vol. 32, pp.5052-5054.
- Herbert C. Roters, 1951, *Electromagnetic Devices*, John Wiley & sons.
- Cheng-Tseng Liu and Sung-Yi Yao, 2002, IEEE Transactions on Magnetics, Vol. 38, No. 5.
- Howard L. Gerber, 2002, IEEE 2002.
- Keisuke Fujisaki, 2001, IEEE Transactions on Industry Applications, Vol. 37, No. 4.
- Keisuke Fujisaki, 2002, IEEE 2002.
- Siegfried Latzel, 2000, IEEE 2000.
- Cheng-Tsung Liu and Sung-Yi Yao., 2002. Electromagnetic Field and Force Analyses of a Non-contacting Conveyance System for Steel Mill Application, IEEE Transactions on Magnetics, VOL., 38, NO. 5.
- Keisuke Fujisaki., 2000. Application of Electromagnetic Force to Run Out Table., IEEE.
- K. Fujisake, T. Ueyama, and K. Wajima, 1996. Electromagnets Applied to Thin Steel Plate, IEEE Transaction on Magnetics, Vol. Mag-32.
- Shinya Hasegawa, Takayuki Obata, Yasuo Oshinoya, and Kazuhisa Ishibashi, 2002, "Study on Noncontact Support and Transportaion of a Rectangular Thin Steel Plate, Proc. Schl. Eng. Tokai Univ., Ser. E, 27, pp. 1 – 12.
- Toshiko Nakagawa, Mikio Hama, and Tadashi Furukawa, 2000, IEEE Trans. on Magnetics, Vol. 36, No. 5, pp. 3686 – 3689.
- David L. Trumper, Ming-chih Weng, and Robert J. Ritter, 1999, Proceedings of the IEEE International Conference on Control Applications, pp. 551 – 557.
- Herbert C. Roters, 1941, *Electromagnetic Devices*, John Wiley & Sons, Inc., .
- S. A. Nasar and I. Boldea, 1976, *Linear Motion Electric Machines*, John Wiley & Sons.
- Jung Soo Choi and Yoon Su Baek, 2002, KSME International Journal, Vol. 16, No. 12, pp. 1643 – 1651, 2002.

STABLE STATES TRANSITION APPROACH

A New Strategy for Walking Robots Control in Uncertain Environments

* Anca Petrișor, ** Nicu George Bîzdoacă, * Adrian Drighiciu
*Faculty of Electromechanical Engineering, University of Craiova, Romania
apetrisor@em.ucv.ro, nicu@robotics.ucv.ro, adrighiciu@em.ucv.ro

** Ilie Diaconu, * Sonia Değeratu, ** Gabriela Canureci, * Gabriela Petropol Serb
**Faculty of Automation, Computers and Electronics, University of Craiova, Romania
diaconu@robotics.ucv.ro, sdeğeratu@em.ucv.ro, gpetropol@em.ucv.ro

Keywords: Walking robot, mathematical model, control strategy, variable causality dynamic system, uncertain environment.

Abstract: A new strategy for walking robots control in uncertain environments, called Stable States Transition Approach (SSTA), is proposed in this paper. All the controls, both the steps succession and the evolution inside each step are established by the evolution environment and by the objective proposed during the evolution. There are no predetermined types of legs movements; they are on-line determined during the robot evolution, irrespective of the ground shape. To apply this strategy it was necessary the robot interpretation as a Variable Causality Dynamic Systems (VCDS). Experimental results are implemented and verified in RoPa, a platform for simulation and design of walking robot control algorithms, to demonstrate the efficacy of the proposed control method.

1 INTRODUCTION

In the last years, many researches combine results from the fields of robotics and control systems, especially for wheeled and legged mobile robots (Fulgenzi, Spalanzani and Laugier, 2007), (Jung, Hsia and Bonitz, 2004).

Mobile robots control in uncertain environments represents still a challenge for real world applications. The robot should be able to gain its goal position facing the implicit uncertainty of the surrounding environment.

The walking robots, particularly the legged robots, allow many advantages with respect to the wheeled robots, especially regarding the autonomy in difficult environments. Unfortunately, a specific type of movement called legged locomotion, (Thirion 2001), (Cubero 2001), is characterized by strongly nonlinear mathematical models to allow describing both the fundamental aspects: leg movements and leg coordination. During the legged locomotion, the control algorithms must assure a stable movement that can be dynamic stable movement or static stable movement.

The problem of walking robots control in uncertain environments has been deeply studied in literature and several techniques have been developed.

Many control algorithms implemented on the existing walking robots, (CWR, 2003), are based on "state of the art" technologies to control the movements of articulated limbs and joint actuators. Some of them try to recreate movements of biological insects which execute various types of periodic gait patterns and adaptive gaits at very high speed, (Cubero, 2003). The walking control algorithms are often around some distributed architectures, (Schmucker, 1996), (Galt, 1999), by assembling a multitude of small processes which are executed concurrently.

Other approaches consider the robot having the necessary intelligence to operate in uncertain environments and use fuzzy logic or neural networks based techniques, (Thirion, 2001), (Gu 2001), (Nanayakkara, Watanabe, Izumi, and Kiguchi, 2001), advanced control schemas, genetic algorithms (Kiguchi, Watanabe, Izumi, and Fukuda, 2000), (Kumarawadu, Watanabe, Kiguchi and Izumi, 2002) etc., to develop the dynamic walking.

In this paper it is developed a new concept of walking called SSTA "Stable States Transition Approach" based on the variable causality mathematical model of the walking robot. According SSTA both the leg coordination and individual leg movements are entirely dependent on the robot goal and the environments only.

The control structure of SSTA, presented in this paper, is proposed in order to apply the best control with respect to safety issues and convergence to the goal.

Simulation results show how the developed walking control algorithm allows the robot to navigate safely, in uncertain environments, toward the goal and to modify its behavior according to the SSTA control strategy.

The paper is structured as follows: in Section II the geometrical structure of the walking robot is described in detail; in Section III the variable causality mathematical model of the walking robot is described and discussed. In Section IV the general block diagram of SSTA walking robot control is proposed. In Section V the algorithm for walking robot control in SSTA strategy is presented. In Section VI simulation results are shown and discussed. Last section closes the paper with conclusions and purposes for future activities.

2 GEOMETRICAL STRUCTURE OF THE WALKING ROBOT

It is considered the walking robot structure as depicted in Fig.1, having three normal legs L^1, L^2, L^3 and a head equivalent to another leg, L_0 , containing the robot centre of gravity, G , placed in its foot. The robot body RB is characterized by two position vectors O^0, O^1 and the leg joining points denoted R^1, R^2, R^3 . The joining point of the head, L^0 , is the central point $O^0, R^0=O^0$, so the robot body RB is univocally characterized by the set,

$$RB = \{O^0, O^1, \lambda^1, \lambda^2, \lambda^3, \lambda^0\} \quad (1)$$

where $\lambda^0 = 0$.

The robot position in the vertical plane is defined by the pair of the position vectors O^0, O^1 where $|O^1 - O^0| = 1$, or by the vector O^0 and the scalar θ , the angular direction of the robot body.

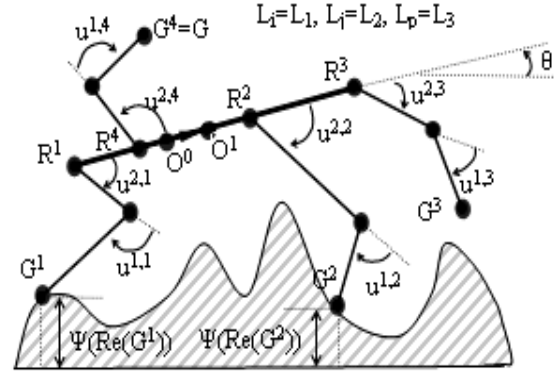


Figure 1: Geometrical structure of the robot.

The robot has a rigid body if the three scalars ($\lambda^i, i=1:4$) are time constant. The variable θ determines the robot body angle in vertical plane. The geometrical structure of the walking robot is defined by the relations

$$O^1 - O^0 = e^{j\theta} \quad (2)$$

$$R^k = O^0 + \lambda^k \cdot e^{j\theta} \quad (3)$$

from which,

$$R^{k_1} - R^{k_2} = (\lambda^{k_1} - \lambda^{k_2}) \cdot e^{j\theta}, k_1, k_2 \in \{1, 2, 3, 4\} \quad (4)$$

The robot position in the vertical plane is defined by the pair of the position vectors O^0, O^1 , where

$$|O^1 - O^0| = 1 \quad (5)$$

or by the vector O^0 and the scalar θ , the angular direction of the robot body.

Each of the four robot legs $L_k, k=1:4$ is characterized by an Existence Relation $ER(L)$ depending on specific variables,

$$ER(L_k): R^k - G^k - A^k = 0 \quad (6)$$

$$A^k = a^k \cdot e^{j\alpha^k} = -a^k \cdot e^{j\theta} \cdot e^{j\alpha^{1,k}} \cdot e^{j\alpha^{2,k}} \quad (7)$$

$$B^k = b^k \cdot e^{j\beta^k} = -b^k \cdot e^{j\theta} \cdot e^{j\beta^{1,k}} \cdot e^{j\beta^{2,k}} \quad (8)$$

$$R^k - G^k + e^{j\theta} \cdot AB^k = 0 \quad (9)$$

$$AB^k = -e^{j\alpha^{2,k}} [b^k + a^k \cdot e^{j\alpha^{1,k}}] \quad (10)$$

From this point of view, the walking robot is an object containing five fundamental components,

$$WR = \{L_1, L_2, L_3, L_4, RB\} \quad (11)$$

This determines a system of equations where the unknown variables selection depends on the robot status. This system is called Existence Relation of the walking robot denoted $ER(WR)$.

This means that in specific circumstances some variables are effects of the others but the causality ordering can be changed. For example, sometimes a junction is external controlled but it could become a free junction as the effect of the other causes.

The mathematical model of this object is a Variable Causality Dynamic Systems (VCDS) and will be analyzed from this point of view.

Irrespective of the leg numbers, any walking robot, evolving in a vertical plane with a rigid body, has only two legs as a support on the ground.

The ground, at the time moment t , is defined by an unknown equation.

$$z = \psi(x, t) \tag{12}$$

about which only some values are obtained

$$z^* = \psi(x^*, t^*) \tag{13}$$

as a result of feet testing at time instant t^* .

A pair of legs $\{L_i, L_j\}$, $i, j \in \{1, 2, 3\}, i \neq j$ constitutes the so called Active Pair of Legs (APL) if the robot body position is the same irrespective of the feet position of all the other legs different of L_i (the prime-active leg) and L_j (the second-active leg).

A robot is a fixed robot on the ground (FRG) if its position is constant in time when both all its commands and the ground are time invariant

$$\psi(x, t) = \psi(x), \forall t \tag{14}$$

A robot containing N proper legs can have only N_a numbers of APL,

$$N_a = C_N^2 = N(N-1)/2 \tag{15}$$

In this case $N=3$ so $N_a=3$. All the other legs that at a time instant do not belong to APL are called Passive Legs (PL).

A label is assigned to each possible APL. The APL label is expressed by a variable q called Index of Activity (IA) that can take N_a values, numbers or strings of characters. For example the string of characters, $q='ijp'$, points out that the pair $\{L_i, L_j\}$ is an APL and the leg L_p is a passive leg. Instead of strings of characters, the IA can take numerical values as for example,

$$q = 123 \Leftrightarrow i = 1; j = 2; p = 3, \Leftrightarrow \\ L_i = L_1; L_j = L_2; L_p = L_3;$$

3 VARIABLE CAUSALITY MATHEMATICAL MODEL OF THE WALKING ROBOT

A good description for walking robot behavior is as a VCDS. In such a system, all the variables that characterize its behavior (the terminal variables) are represented by a matrix X called the global variable of the system. In the case of the above robot, the matrix X is a 16×5 matrix. The first four columns of this matrix contain variables related to the legs L_k

$k=1:4$ and the fifth variable related to the robot body or other useful information.

For example, the k -column contains

$X^k = [u^{1,k}, u^{2,k}, R^k, G^k, s^k, \alpha^k, \beta^k, a^k, b^k, \lambda^k]$, $k=1:4$, where s^k expresses the state of the k leg L_k and the fifth column contains

$$X^5 = [O^0, \theta, \varepsilon^{12}, \varepsilon^{23}, \varepsilon^{31}, \dots]$$

where $\varepsilon^{12}, \varepsilon^{23}, \varepsilon^{31}$ express the stability indexes.

A distinction has to be pointed out between the walking robot as a physical object, which has a mathematical model, and different systemic input-output representations generated by this mathematical model.

These different systemic input-output representations refer specially to different VCDS extensions of the walking robot model subsystems. VCDS representations are used in the SSTA control algorithm of the walking robot.

According to SSTA, all the control actions are closed loop performed. Both the sequence of different types of movements and evolutions inside of each specific movement, depend on the general walking robot behavior objective and the environment only.

Even if, as a physical object, the walking robot has some command parameters, it behaves as a VCDS because of the internal kinematics restrictions that determine mechanical locks of the rigid body. For example in this paper the variables

$$u^{1,I}, u^{2,I}, I = \{1, 2, 3, 4\} \tag{16}$$

are the command parameters, as inputs to robot actuators, but discrepancies can appear between the values as desired values supplied by the control device and the realized values of these parameters. In addition, a physical robot can have the possibility of controlling its causal structure through a new variable cz . For example, the angle $u^{2,j}$ is set as free angle or other angles intentionally are set free. The values of the free variables depend on the kinematics restrictions or depend on the position parameters, intentionally some how modified. For example, it is possible externally to modify the position vector R^i or only its real or imaginary part $R^k = R^{k,x} + j \cdot R^{k,y}$, $k=1:4$. All these justify the interpretation of the walking robot as VCDS.

In the framework of VCDS description, inputs and outputs do not exist. All the variables are terminal variables satisfying the System Existence Relation (SER). As long as the system exists, the SER is true according to a causality ordering specified by the variable cz which acts as a true input variable.

In the VCDS approach with discrete time evolution, the global variable X is represented by three instants: X_k^d , X_{k-1} and X_k which respectively express: the desired value at the current step, the previous value and the actual realized value.

Depending on the actual value of the ordering structure signal cz_k , and the actual value of the index of activity signal q_k , only some components of the matrix X_k^d are effectively realized. The VCDS evolution equation of the WR is

$$X_k = F(X_{k-1}, X_k^d, q_k, cz_k) \quad (17)$$

The VCDS model of the walking robot is used in the SSTA control structure proposed by the authors. The behavior of the walking robot in different causal structure is analyzed in details in other papers.

4 SSTA CONTROL STRUCTURE

It is consider that the walking robot has to evolve in space along a direction Ox , which determines a vertical section in the plane xOz of the evolution environment as an unpredictable but measurable function, called also the ground,

$$z = \psi(x) \quad (18)$$

The evaluation of the curve $\psi(x)$ can be performed by using walking robot external tools or by using its legs for ground testing.

Evolution of the walking robot on unpredicted ground $\psi(x)$ implies performing very complex movements for walking robot legs. They must be coordinated in such a way to avoid ground collision during gaiting. Generally, in classical approaches, the legs movements are predetermined, specifically for different typical shapes of the ground.

The general block diagram of SSTA walking robot control is presented in Fig.2.

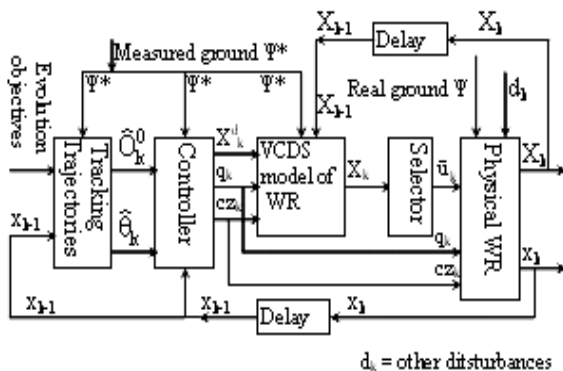


Figure 2: The general block diagram of SSTA walking robot control.

The evolution attitude refers, for example, to the forward or backward evolution, having some imposed fixed body angle, or variable body angle but with maximum stability.

As a physical object, at a time moment $t=kT$, the robot is controlled by the variables \hat{u}_k, q_k, cz_k , where \hat{u}_k represents the matrix of the desired values command angles of the proper legs L_1, L_2, L_3 , and of the head, assimilated as a leg, L_4 . So,

$$\hat{u}_k = \begin{bmatrix} u_k^{1,1} & u_k^{1,2} & u_k^{1,3} & u_k^{1,4} \\ u_k^{2,1} & u_k^{2,2} & u_k^{2,3} & u_k^{2,4} \end{bmatrix} \quad (19)$$

The variables q_k, cz_k , represent the values of the activity index q and respectively the causal ordering cz at the time moment $t=kT$. By d_k there are equivalently represented all the other disturbances acting on the physical robot evolving in the environment expressed by the function Ψ . The desired values \hat{u}_k are applied to the positioning systems, as a request, but they are not necessarily realized. This depends on the values of q_k, cz_k .

Applying to the physical robot the desired commands \hat{u}_k , under the conditions of q_k, cz_k, Ψ, d_k , the global variable X , takes the value X_k and the abscissa x of the robot centre point O^b takes the value x_k .

The values X_k, x_k will be utilized by the control algorithm in the next time step. The real evolution ground, expressed by a function Ψ is externally or internally expressed by a function Ψ^* , known, at least, around the actual position of the robot.

5 IMPLEMENTATION OF THE WALKING ROBOT CONTROL ALGORITHM IN SSTA STRATEGY

By SSTA strategy is assured the walking robots evolution in uncertain environments subordinated to two goals:

- achievement of the desired trajectory expressed by the functions $O_z^0 = f(x)$ and $\theta = \theta(x)$, where x is the ground abscissa and $O_x^0 = x$; it is considered the evolution from left to right;
- assurance of the system stability that is, in any moment of the evolution the centre of gravity has to be in the stability area.

Considering the walking robot as a variable causality dynamic system it is possible to realize this desideratum in different variants of assurance the steps succession. The steps succession supposes a series of elementary actions that are accomplished only if the stability condition exists.

Continuously, by sensorial means or using the passive leg, the robot has informations about its capacity of evolving on the ground. Every time it is considered that the legs i, j are on the ground and the the system is stable ($\varepsilon_{ij} \in [0,1]$). The passive leg G^p is which realises the walking.

By testing the ground is realized its division in lots representing the fields on x axis which constitute the abscissas of some points that can be touched by the G^p leg. The leg will always touch the ground only on an admitted lot.

A next support point given by the free G^p leg, is chosen so that to existe a next stable state ε_{ip} or ε_{jp} , taking into account the actual state of legs activity. For example, if $q=132$, passive leg (which tests) is $G^p = G^2$ and assures $\varepsilon_{12} \in [0,1]$ or $\varepsilon_{23} \in [0,1]$. When the change of legs activity is realised ($q=123$ or $q=321$ or $q=231$ etc.), the present passive leg G^p will become the leg i or the leg j .

In this paper, a variant of movements succession, composed by 12 steps, is proposed.

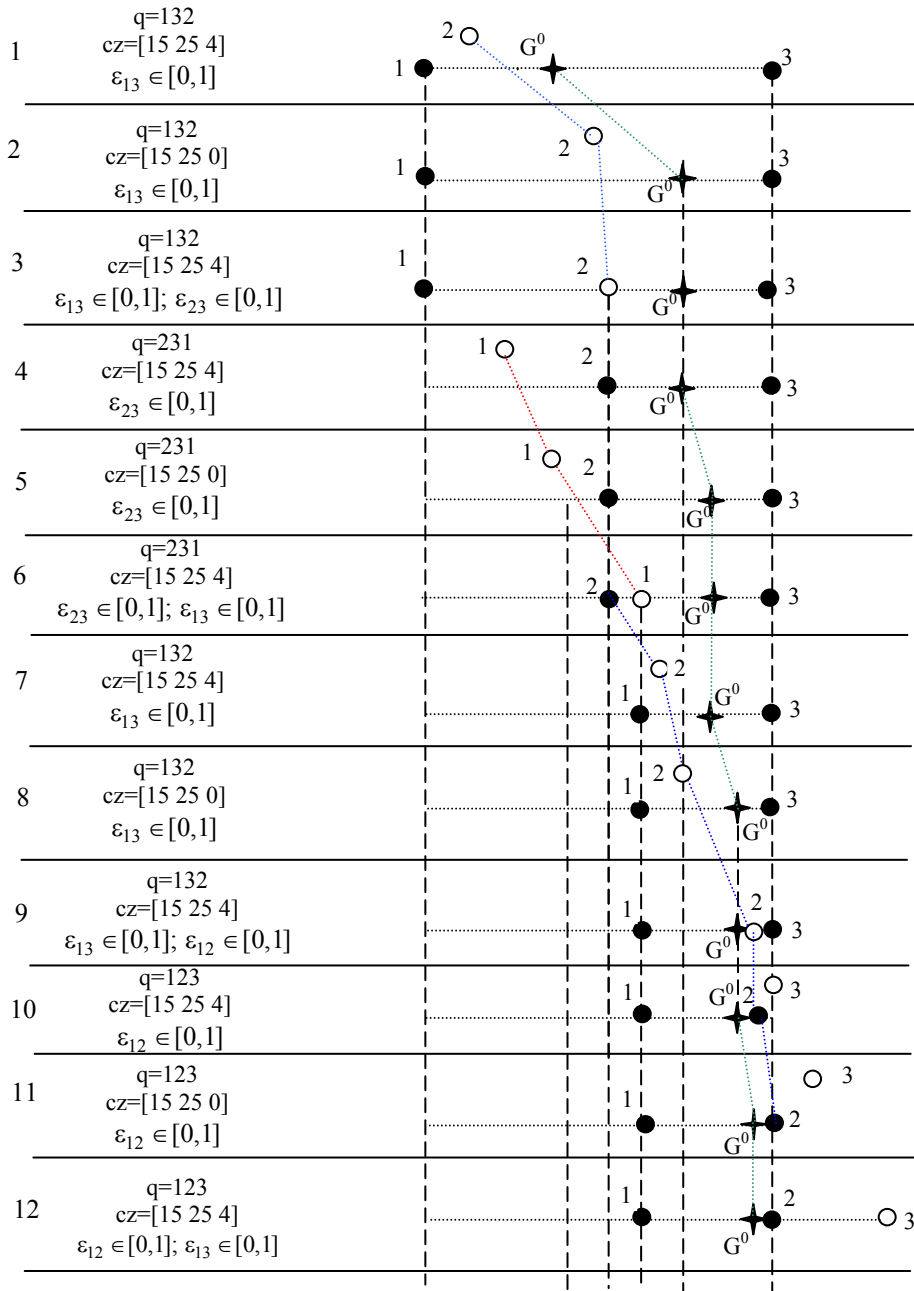


Figure 3: The graphical representation of SSTA walking strategy.

6 EXPERIMENTAL RESULTS

An experimental platform, called RoPa, has been conceived. The RoPa platform is a complex of MATLAB programs for simulation and control of walking robots evolving in uncertain environments according to SSTA control strategy.

A number of eight causality orderings of the robotic structure have been implemented on RoPa.

Figure 4 presents the interface of this application for the causality structure with four free joints. The four degrees of freedom are thus consumed: one to fulfil the kinematics restriction; one to ensure the desired value of the θ angle of the robot body and two for the desired values $\hat{O}^0(O_x^0, O_z^0)$ of the robot body.

The causal ordering is activated by selecting the causal variable $cz=[15\ 25\ 0]$.

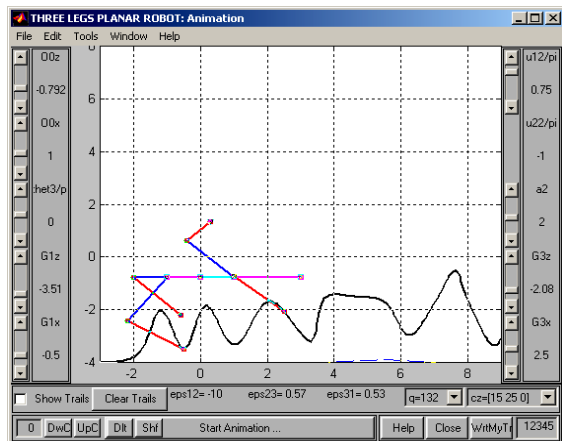


Figure 4: RoPa Graphic User Interface.

The stability of this evolution is graphical represented by a stability certificate of the evolution. This certificate attests the stability index of the active pair of legs in any moment.

7 CONCLUSIONS

The experiments performed on RoPa demonstrate the efficacy and adaptability of the proposed method when the walking robots evolve in uncertain environments. All the causal orderings are perfectly integrated in RoPa structure proving the correctness of the theoretical results.

The mathematical model developed in the paper becomes an element of the VCDS walking robot

model. The robustness of this mathematical model was proved by many experimental results.

Further investigations will be directed towards a hexapod robot performing a task in uncertain environment.

REFERENCES

- Cubero S., 2001. A 6-Legged Hybrid Walking and Wheeled Vehicle. 7-th International Conference on Mechatronics and Machine Vision in Practice, USA.
- CWR, 2003. The Climbing and Walking Robots, Home Page, www.uwe.ac.uk/clawar.
- Fulgenzi, C., Spalanzani A., Laugier C., 2007. Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid, *Robotics and Automation*, 2007, pp.1610-1616.
- Galt S., Luk L., 1999. Intelligent walking gait generation for legged robots. *Proc. 2-th International Conference on Climbing and Walking Robots*, pp.605-613.
- Jung, S., Hsia, T.C., Bonitz, R., 2004. Force Tracking Impedance Control of Robot Manipulators Under Unknown Environment. *IEEE Transactions on Control Systems Technology*, 12(3), 474-483.
- Kiguchi, K., Watanabe, K., Izumi, K., Fukuda, T., 2000. Application of Multiple Fuzzy-Neuro Force Controllers in an Unknown Environment Using Genetic Algorithms, *Proc. of IEEE International Conference on Robotics and Automation*, pp. 2106-2111.
- Kiguchi, K., Miyaji, H., Watanabe, K., Izumi, K., Fukuda, T., 2000. Design of Neuro Force Controllers for General Environments Using Genetic Programming, *Proc. of the Fourth Asian Fuzzy Systems Symposium (AFSS2000)*, pp. 668--673.
- Nanayakkara, T., Watanabe K., Izumi, K., Kiguchi, K., 2001. Evolutionary Learning of a Fuzzy Behavior Based Controller for a Nonholonomic Mobile Robot in a Class of Dynamic Environments, *Journal of Intelligent and Robotic Systems*, Vol. 32, No. 3, pp. 255--277.
- Nanayakkara, T., Watanabe, K., Kiguchi, K., Izumi, K., 2001. Fuzzy Self-Adaptive Radial Basis Function Neural Network-Based Control of a Seven-Link Redundant Industrial Manipulator, *Advanced Robotics*, Vol. 15, No. 1, pp. 17--43.
- Sisil Kumarawadu, Keigo Watanabe, Kazuo Kiguchi, and Kiyotaka Izumi, Neural Network-Based Optimal Adaptive Tracking Using Genetic Algorithms, *Proc. of 4th Asian Control Conference*, pp105-110.
- Schmuacer U., Schneider A. Ihme T., 1996. Six Legged Robot for Service Operations. *Proc. of EROBOT'96; IEEE Computer Society Press*, pp:135-142.
- Thirion, B., Thiry, L., 2001. Concurrent Programming for the Control of Hexapod Walking. *7-th International Conference on Mechatronics and Machine Vision in Practice*, USA.

ROBOT GOES BACK HOME DESPITE ALL THE PEOPLE

Paloma de la Puente, Diego Rodriguez-Losada, Luis Pedraza and Fernando Matia
DISAM - Universidad Politecnica de Madrid, Jose Gutierrez Abascal 2, Madrid, Spain
paloma.puente.yusty@alumnos.upm.es, diego.rlosada@upm.es

Keywords: Mobile Robots Navigation, Localization and Mapping, Reactive Control, Dynamic Points.

Abstract: We have developed a navigation system for a mobile robot that enables it to autonomously return to a start point after completing a route. It works efficiently even in complex, low structured and populated indoor environments. A point-based map of the environment is built as the robot explores new areas; it is employed for localization and obstacle avoidance. Points corresponding to dynamical objects are removed from the map so that they do not affect navigation in a wrong way. The algorithms and results we deem more relevant are explained in the paper.

1 INTRODUCTION

Autonomous navigation around indoor environments is a difficult task for a mobile robot to achieve, especially if there are people passing by frequently. A good start point is presented in (Borenstein et al., 1996) by breaking down the general problem of robot navigation into three questions: "Where am I?", "Where am I going?", "How should I get there?". So, the first (and main) problem encountered when dealing with this issue is the necessity of knowing where the robot is at every moment. As the robot moves, errors in odometry information increase significantly hence making it essential that these data be corrected. Different probabilistic methods for performing this correction using measurements received from stereoceptive sensors (such as laser range-finders or sonars) have thus far been developed, being those capable of building a map at the same time for proper representation of the environment the most effective and popular ones.

As for the second question, the goal to be reached is often defined by the user. It may be given by higher level tasks depending on the particular application.

The last question is challenging as well. A first step is motion control, which is better addressed by means of a closed-loop controller using position feedback (Siegwart and Nourbakhsh, 2004). With a regulator of this kind, path planning comes to computing a sequence of passing points leading to the target. Once a nominal trajectory has been obtained, safe navigation requires reactive control, for the robot

should be able to change its behavior if a situation that endangers its mission appears. Regarding this, several strategies have been used in the literature to face obstacle avoidance. Some of the proposed solutions (Feiten et al., 1994), (Yang and Li, 2002) consist of sending special drive and steer velocity commands when an obstacle is detected. The latter and other authors do so through fuzzy control. An interesting and generic approach is an iterative algorithm found in (Lamiroux et al., 2004) for real time deformation of previously collision free paths when operating with nonholonomic robots.

Dynamic objects in the environment bring about further difficulties in map building and reactive control. If the problem is simplified and observed features are represented as permanent in the map, it is still useful for localization purposes but there will be discrepancies with reality. It also increases the map's size unnecessarily and may result in the robot avoiding obstacles which are no longer there. (R.Siegwart et al., 2002) tackle this issue applying the EM algorithm and making use of an a priori map of the environment. They also address other aspects of robot navigation in populated exhibitions, remarking the importance of introducing novel combinations and adaptations of different preexisting approaches. (Hhnel et al., 2003) developed a statistical method to identify measurements corresponding to dynamic objects and perform localization and building of occupancy grid maps, all in the context of the EM algorithm.

In this paper we present a system which allows

the robot to build a point-based map of its static surroundings while being teleoperated from somewhere; this map is afterwards used by the robot to localize itself and make its way back to its original pose, avoiding any obstacles which may be near the initial path. Experiments in highly crowded and cluttered environments have been carried out successfully.

This work is to be used in an ambitious project concerning the autonomous setup of an interactive robot at museums and trade fairs. The robot's name is Urbano and it already counts on a robust localization system based upon geometrical features over SLAM-EKF that originates 2D precise maps in real time (Rodríguez-Losada, 2004). Here we expose an efficient less complex model with the specific demonstrator of the returning home utility. This work has been partially founded by DPI-2004-07907-C02-01.

The paper is organized as follows. Section 2 includes a description of the algorithms corresponding to localization and mapping. In Section 3 we present the movement and reactive control implemented algorithms. Section 4 accounts for experimental results we have obtained. Finally, Section 5 contains our conclusions and future working lines.

2 LOCALIZATION AND MAPPING

2.1 Localization and Map Building

The solution adopted for this problem is a Maximum Incremental Probability algorithm whose foundation is the Extended Kalman Filter(EKF). The system elaborates and continually updates a map built from the observations acquired from the laser measurements.

The state vector used is defined as the robot's global pose, $[x_R, y_R, \theta_R]^T$, whereas the odometry measurements (incremental, so referred to the robot's local coordinate system) represent the system inputs, $\vec{u} = [u_x, u_y, \theta_u]^T$. Time subscripts are omitted in the latter so as to simplify notation; measurements obtained last are always the ones considered. According to this, the state equation is put as:

$$\vec{x}_{R_k} = \vec{x}_{R_{k-1}} \oplus \vec{u} = f(\vec{x}_{R_{k-1}}, \vec{u}) \quad (1)$$

where \oplus represents the composition of relative transformations. Odometry measurements can be modelled as a gaussian variable, $\vec{u}_k \sim N(\hat{\vec{u}}_k, Q)$.

2.1.1 Predictor Equations

Applying the definition of the \oplus operator, the predicted state, \vec{x}_{R_k} , will be given by:

$$\begin{pmatrix} \vec{x}_{R_k} \\ \vec{y}_{R_k} \\ \vec{\theta}_{R_k} \end{pmatrix} = \begin{pmatrix} \hat{x}_{R_{k-1}} + u_x \cos \hat{\theta}_{R_{k-1}} - u_y \sin \hat{\theta}_{R_{k-1}} \\ \hat{y}_{R_{k-1}} + u_x \sin \hat{\theta}_{R_{k-1}} + u_y \cos \hat{\theta}_{R_{k-1}} \\ \hat{\theta}_{R_{k-1}} + \theta_u \end{pmatrix} \quad (2)$$

And the state's covariance prediction, \vec{P}_k , is computed from:

$$\vec{P}_k = F_x \hat{P}_{k-1} F_x^T + F_u Q F_u^T \quad (3)$$

being $F_x = \frac{\delta f}{\delta \vec{x}} |_{\vec{x}_{R_k}}$, $F_u = \frac{\delta f}{\delta \vec{u}} |_{\vec{x}_{R_k}}$ (we use the best state estimation obtained up to now)

The algorithm is initiated with $\hat{x}_0 = 0, \hat{P}_0 = 0$.

2.1.2 Corrector Equations and Data Association

Ideally, each observation $\vec{o}_i = [o_{ix}, o_{iy}]^T$ (or laser measurement received) is implicitly related to the previous estimation of the state by means of composition with it and comparison to the corresponding $\vec{l}_j = [x_{lj}, y_{lj}]^T$ point in the map. The resultant expression is known as the innovation of observation i :

$$\vec{h}_{ij} = \vec{x}_R \oplus \vec{o}_i - \vec{l}_j = h(\vec{x}_R, \vec{o}_i, \vec{l}_j) = \vec{0} \quad (4)$$

which combined with \oplus definition is the same as:

$$\vec{h}_{ij} = \begin{pmatrix} -x_{lj} + \vec{x}_R + o_{ix} \cos \vec{\theta}_R - o_{iy} \sin \vec{\theta}_R \\ -y_{lj} + \vec{y}_R + o_{ix} \sin \vec{\theta}_R + o_{iy} \cos \vec{\theta}_R \end{pmatrix} = \vec{0} \quad (5)$$

If we denote $H_{x_{ijk}} = \frac{\delta h_{ij}}{\delta \vec{x}} |_{\vec{x}_{R_k}, \vec{o}_i}$, $H_{z_{ijk}} = \frac{\delta h_{ij}}{\delta \vec{o}_i} |_{\vec{x}_k, \vec{o}_i}$ for every iteration k , the covariance of each innovation is given by:

$$S_{ijk} = H_{x_{ijk}} \vec{P}_k H_{x_{ijk}}^T + H_{z_{ijk}} R H_{z_{ijk}}^T \quad (6)$$

where R is the covariance of laser measurements.

We have followed the *Nearest Neighbor* strategy to pair each observation with its correspondent map point. Computation of the Mahalanobis distance for innovation h_{ijk} once we have got its covariance matrix S_k will let us select that map point which minimizes such distance. If Mahalanobis test is passed for that association, then it is taken into account at the correction step; otherwise it will be added as a new point of the map vector. Among all the h_{ij} , $H_{x_{ij}}$ and $H_{z_{ij}}$ matrices obtained for an observation at a certain iteration, only those corresponding to the map point, if any, associated to it will be kept to correct the estimation. They will be denoted $h_{i_{min}}$, $H_{x_{i_{min}}}$, $H_{z_{i_{min}}}$. As more associations are made, there are more measurements to

be used. This is contemplated by filling other matrices containing joint information from all of them:

$$\vec{h} = [h_{1_{min}}, \dots, h_{t_{min}}]^T \quad (7)$$

$$H_x = [H_{x1_{min}}, \dots, H_{xt_{min}}]^T \quad (8)$$

$$H_z = \begin{pmatrix} H_{z1_{min}} & & \\ & \ddots & \\ & & h_{t_{min}} \end{pmatrix} \quad (9)$$

where t is the total number of associations. Now we get the global S matrix and the Kalman gain from

$$S = H_x \tilde{P} H_x^T + H_z R H_z^T \quad (10)$$

$$K = \tilde{P} H_x^T S^{-1} \quad (11)$$

The corrected values are updated from this information and the prediction values obtained from 2 and 3:

$$\hat{x} = \tilde{x} - Kh; \quad (12)$$

$$\hat{P} = (I - KH_x) \tilde{P} \quad (13)$$

2.2 Removal of Dynamic Points

The algorithm presented above incorporates to the map every observed point which cannot be associated to a pre-existing one. If we stick to it, points corresponding to a dynamic object detected at a given moment are included in the map when they are first seen (just as any other object is) and nothing is done to avoid them being there forever. Here we propose a method that removes map points far from being observed at a given moment, before the correction and update of the map take place.

Firstly, an evolvent polygon of the laser measurements is constructed in a recursive way. The general procedure is summarized in the following lines:

- A segment from the first measurement to the last is taken.
- The furthest observation from that segment is sought among the others.
- If the separation between the segment and that observation is high enough, the process is repeated between the first observation and the one selected and then between this observation and the last one.
- When for one segment there is no observation at a greater distance than a threshold, its limits are stored in a vector containing the polygon vertices.

Some other simplifications are made in order not to generate too many vertices. Map points that remain quite inside the polygon cannot correspond to currently observed objects because if so, they would either have been included as polygon vertices or they would be very close to them (due to the threshold and the simplifications mentioned above).

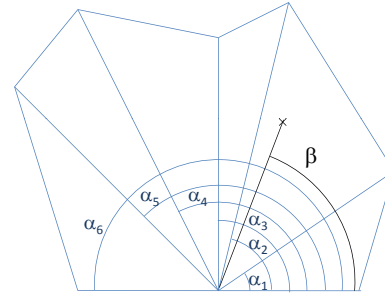


Figure 1: Angles to be measured for seeing if a point is inside an unconvex polygon.

To find out whether a map point is inside the polygon, we compute a series of angles ($\alpha_1 \dots \alpha_v$ in figure 1) as soon as the vertices extraction is over. Then, for every point in the map which is not too near the polygon's border we determine its coordinates in the laser reference system and then compute the value of the β angle represented in the same figure. Selecting those alphas β lies in between (α_1 and α_2 in the figure's case) we obtain the corresponding vertices and use their distance to the robot to establish if that map point should be erased. We use a conservative criterion that leaves outside the map only those points at a smaller distance than the closest of both vertices. This prevents any map points being removed incorrectly.

The computational cost of creating the polygon and finding the values of the needed angles is upper bounded, since the number of observations provided by the laser is constant. Notice that only one angle has to be computed for each point in the map, which makes it a not very cumbersome algorithm.

3 MOTION CONTROL

3.1 Motion Controller

To get the robot moving from one point to another we have employed gain scheduling, implementing a controller in agreement to the divide and conquer approach. The error is a linear combination of three angles. In first place is the difference between the present orientation of the robot and the one it should

have to look straight ahead to the next goal. The second angle is that one the robot should turn to have the same global orientation as the current trajectory segment it is at. The third angle is similar to the second although in this case it is the next trajectory segment that is considered. Weights of 0.3, 0.2 and 0.5 have been used to produce smooth anticipative movement following a given trajectory. We define several intervals for that global error and vary a constant value of drive velocity and the gain of a proportional controller for the steer velocity within each of them.

In the experiments described here, the given trajectory is obtained by saving the robot's pose every time it moves a constant distance.

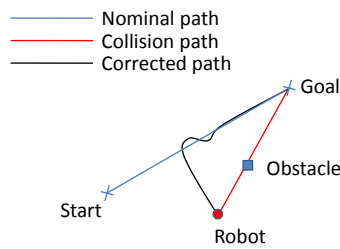


Figure 2: Possible paths if the robot is not near the defined trajectory.

If the robot were to be quite separated from the defined trajectory, this controller would take it to the goal not across the nominal path but across one in which there might be an obstacle (figure 2).

The corrected path is obtained by forcing the robot to approach the nominal path before heading for the target. For this purpose we have included control laws that regulate angles δ_1 or δ_2 in figure 3 depending on which side of the path the robot is at. We use angles in $[-\pi, \pi]$ along all our work, which is what the `ForceInRange` function has been defined for. These laws are employed instead of the initial ones only when the robot is far enough from the defined path.

When the robot is told to go back, it turns 180 at first and then come into action the rules that have just been commented. At a certain distance from home, the robot begins to slow down until it reaches its final pose. Then it turns to adopt the orientation it had when it abandoned home.

3.2 Path Deformation

To get a trajectory free of obstacles we displace the points of the nominal trajectory which are affected by objects detected in the environment. To begin with, we make sure that consecutive points in the initial trajectory are at a short enough distance, let it be 0.1m. For each point i we follow the same steps. At first, a point

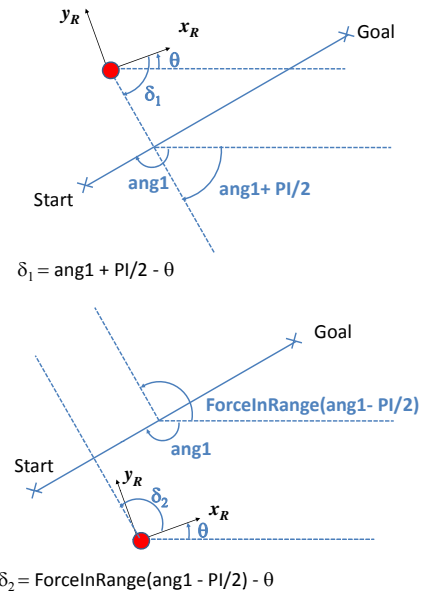


Figure 3: Angles through which control is executed.

p is defined on the normal to the segment joining that point of the trajectory and the next one so that it is 1m away from the former. The vector going from point i to point p will be referred to as v_1 . We define a vector v_2 with origin at the considered trajectory point and end at the subsequent obstacles detected. The orthogonal projection of this vector v_2 onto v_1 determines point p_2 . Its distance to the trajectory point i is d , which may be positive or negative if the obstacle is at one side or another of the trajectory.

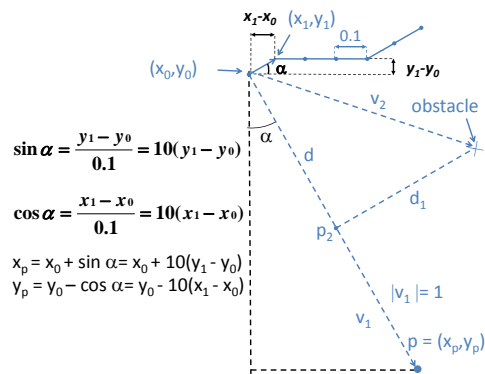


Figure 4: Definition of point p and other magnitudes for the trajectory point $i = 0$.

The distance from p_2 to the obstacle will be called d_1 . All these magnitudes are represented in figure 4. Only those objects resulting in a small enough value of d_1 take part in the deformation relative to that

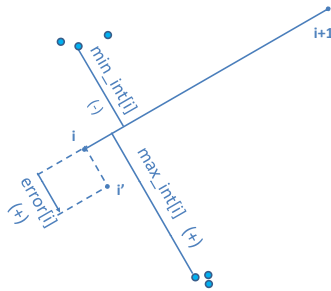


Figure 5: Displacement applied to each point of the nominal trajectory.

path point. Correcting d by considering the influence of the robot radius and afterwards obtaining its minimum and maximum values, an area of permitted movement can be determined. We will call these two distances $\text{min_int}[i]$ and $\text{max_int}[i]$, respectively.

They are initialized with the maximum displacement allowed, for the case of no presence of obstacles and other similar situations. Their mean will be the displacement, $\text{error}[i]$, to apply to that path point in order to leave it at an intermediate distance between the limits found for each of both sides (see figure 5). To generate a smoother deformation we use the average of that error value and the previous and next ones:

$$\text{error2}[i] = (\text{error}[i - 1] + \text{error}[i] + \text{error}[i + 1]) / 3$$

4 EXPERIMENTS AND RESULTS

The system has been tested in different environments, using simulators (one we have developed ourselves and also MobileSim, by Activemedia Robotics) and real robot data from text files at the beginning, and the robot Urbano afterwards. Urbano stands on a B21r platform and has a laser SICK LMS200 mounted on top. The architecture of the system is the one in fig. 6.

The most significant experiments we have conducted took place at our laboratory, which has very narrow areas and people often coming to and fro. The

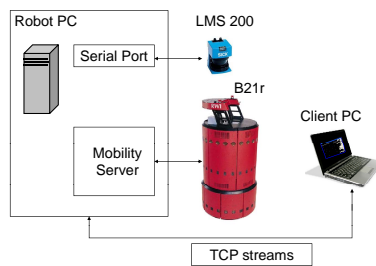


Figure 6: Urbano's distribution schema.

robot was teleoperated from one end of the laboratory to the other, finding a lot of visitors while building the map. When we had brought it to the main entrance, we told him to return home. Since that moment all of its behavior is absolutely autonomous. In the way back, other people were seen and Urbano was able to avoid them. Some other slight corrections were made in order to follow a path which got apart from close obstacles. One of the maps we obtained is the one in figure 7. The green path is the odometry corresponding to the teleoperation mode. The pink path is the one representing dead-reckoning in the way back. The blue trajectory is the correction for the first stage of the experiment, while the red one is the correction obtained when the robot was coming back home. Cyan points are those which were removed from the map. Most of them were clearly identified with people's successive positions when walking in front of the robot.

Another experiment was performed in the building where lessons are imparted at a time in which there were plenty of students coming out from their classes. The resultant map and the actual environment are shown in figure 8.

5 CONCLUSIONS

The system presented here has two main components, one having to do with localization and map building and one related to control and path deformation.

The implemented localization and mapping algorithm is a Maximum Incremental Probability method based on the Extended Kalman Filter (EKF). Its main drawback is the fact that it does not consider the uncertainty in the map itself, but it allows for a higher degree of simplicity and has proven an appropriate behavior on the experimental conditions of a wide variety of tests apart from those exposed in this paper (using data obtained by several real robots in different indoor environments, some of them having large odometry errors). The elimination of points makes more realistic and reliable maps which represent the last structure observed by the robot. If the same map were used in another experiment, instead of making the robot build a new one in real time, it would get adapted to the configuration of the environment at that moment. This strategy also prevents mistakes in data association and reduces maps' size when possible.

The developed control module enables the robot to achieve a final target without hitting any obstacles. A high level technique keeps the path to be followed away from the objects detected by the laser, and the controller does not let the robot get far from this collision free path. Obstacles at lower height than the

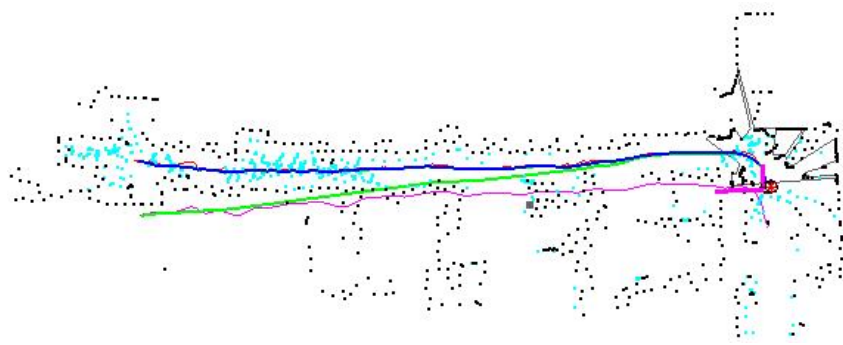


Figure 7: Real time built map of our laboratory.

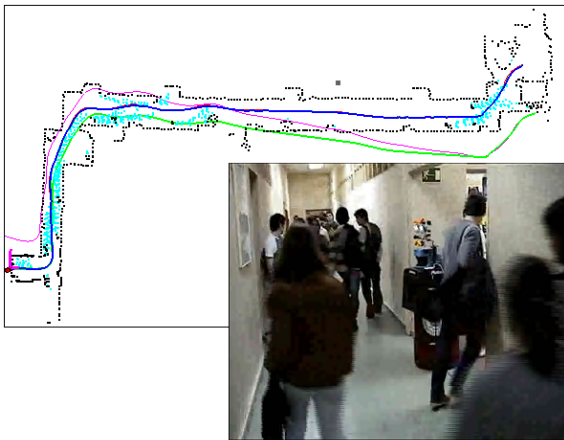


Figure 8: Real time built map of an area plenty of students.

laser sensor cannot be seen. Consequently, not only will these obstacles be left out of the map and cause it to be less accurate but, what is much worse, they will not be considered in path deformation either. As a precaution, Urbano has got some ultrasound sensors at an intermediate height which make the robot stop if they detect it is about to crash, but it might not be enough under some circumstances. This caveat gives rise to one of our next working lines.

6 FUTURE WORK

As previously mentioned, we now orientate our work towards the integration of knowledge models for the autonomous setup of an interactive robot at museums and trade fairs. The reference of this project, called Robonauta, is: DPI2007-66846-c02-01.

To achieve this important objective, some lines related to this work and to Urbano's navigation in general are:

- Extension of geometrical models. Employing a

wrist that can precisely position the 2D scanner in different angles for data acquisition, we aim at constructing robust algorithms to create 3D maps which will make the robot's navigation safer.

- Improvements in control, so that the robot can move faster and follow its path more accurately.
- Path definition by means of people tracking or by means of voice commands spoken to the robot.
- Incorporation of new conducts which will enable the robot to follow a learning process by itself.

REFERENCES

- Borenstein, J., Everett, H., and L.Feng (1996). *Navigating Mobile Robots: Systems and Techniques*. A.K Peters.
- Feiten, W., R.Bauer, and Lawitzky, G. (1994). Robust Obstacle Avoidance in Unknown and Cramped Environments. In *IEEE Int. Conf. Robotics and Automation*.
- Hhnel, D., R.Triebel, W.Burgard, and S.Thrun (2003). Map Building with Mobile Robots in Dynamic Environments. In *IEEE Int. Conf. Robotics and Automation*.
- Lamiroux, F., Bonnafous, D., and O.Lefebvre (2004). Reactive Path Deformation for Nonholonomic Mobile Robots. *IEEE Transactions on Robotics*.
- Rodríguez-Losada, D. (2004). *SLAM Geométrico en Tiempo Real para Robots Móviles en Interiores basado en EKF*. PhD thesis, ETSII-Universidad Politécnica de Madrid.
- R.Siegwart, R.Philippsen, and B.Jensen (2002). <http://robotics.epfl.ch>.
- Siegwart, R. and Nourbakhsh, I. (2004). *Introduction to Autonomous Mobile Robots*. MIT Press.
- Yang, S. and Li, H. (2002). An Autonomous Mobile Robot with Fuzzy Obstacle Avoidance Behaviors and a Visual Landmark Recognition System. In *7th ICARCV*.

USING STEREO VISION AND TACTILE SENSOR FEATURES For Grasp Planning Control

Madjid Boudaba¹, Nicolas Gorges², Heinz Woern² and Alicia Casals³

¹TES Electronic Solution GmbH, Zettachring 8, 70567 Stuttgart, Germany
madjid.boudaba@tesbv.com

²Institute of Process Control and Robotics, University of Karlsruhe
Engle-Bunte-Ring 8-Gebaeude 40.28, 76131 Karlsruhe, Germany
gorges@ira.uka.de, woern@ira.uka.de

³GRINS: Research Group on Intelligent Robots and Systems
Technical University of Catalonia, Pau Gargallo 5, 08028 Barcelona, Spain
alicia.casals@upc.edu

Keywords: Stereo vision, Tactile sensors, Grasp planning, Features matching.

Abstract: Planning the grasp positions either from vision or tactile sensors one can expect various uncertainties. This paper describes a scheme that match visual stereo and tactile data based on stereo vision and tactile sensors. For grasp planning, initially, the grasping positions are generated from stereo features, then the feedback of tactile features is used to match these positions. The result of the matching algorithm is used to control the grasping positions. The grasping process proposed is experimented with an anthropomorphic robotic system.

1 INTRODUCTION

In recent years, considerable research in robotic grasping systems has been published. The proposed system works by using the principle of *sensing-planning-action*. To place our approach in perspective, we review existing methods for sensor based planning for grasping. The existing literature can be broadly classified into three categories; vision based, tactile based and both vision-tactile based. For all categories, the extracted image features are key factors, they can range from geometric primitives such as edges, lines, vertices and circles to optical flow estimates. The first category uses visual image features to estimate the grasping points and from them define the robot's motion with respect to the object position and orientation before performing a grasp (Yoshimi and Allen, 1994), (Maekawa et al., 1995), (Smith and Papanikolopoulos, 1996), (Sanz et al., 1998), (Kragic et al., 2001), and (Morales et al., 2002). The second category uses tactile image features to estimate the characteristics of the area in contact with the object (Berger and Khosla, 1991), (Chen et al., 1995), (Perrin et al., 2000), and (Lee and Nicholls, 2000). The last category uses data fusion from both vision and tactile sensors in order to control grasping tasks efficiently (Namiki and Ishikawa, 1999), and (Allen et al., 1999).

This paper is an extension of our previous work (Boudaba and Casals, 2006) and (Boudaba and Casals, 2007) on grasp planning using visual features. In this work, we demonstrate the utility of matching both visual and tactile image features in the context of grasping, or fingers position controlling. In our approach, we avoid using any object model, and instead, we work directly from image features to plan the grasping points. In order to avoid finger positioning errors, matching, by back projecting these tactile features into visual features is required to compute the similarity transformation that relates the grasping region with the sensitive touching area. To achieve a high level of grasping position matching efficiency, two matching schemes are considered in this paper. The first establishes grasp points correspondences between the left and right images of the stereo head. In this scheme, only the grasp positions are back projected into one side of the stereo image. A second scheme is a region matching where the whole sensitive touching area with the object is used in the back projection into the visual image. All the points belonging to the sensitive area of a tactile sensor are back-projected into the grasp region of visual features. The processing in each match is completely independent and can be performed at its own rate. Our approach based on features matching can play the

critical role of forcing the fingertips to move towards the desired positions before the grasping is executed.

2 GRASPING SYSTEM DESCRIPTION

In robotic grasping tasks, when data from several sensors are available simultaneously, it is generally necessary to precisely analyze all of them along the entire grasping process (see Figure 1). The object being extracted from a video sequence requires encoding its contour individually in a layered manner and provide at the receiver's side some enhanced visual information. In the same way, from the data being extracted from a tactile sensor, the tactile layer processes and provides the tactile information at its receiver's side. The architecture of the whole grasping system is organized into several modules, which are embedded in a distributed MCA2 (Modular Controller Architecture Version 2) software framework (Scholl et al., 2001). There are mainly three modules involved in this development; the stereo vision, tactile sensors, and grasp planning. In MCA2, every module is structured in a data vector that allows the module to receive and send the data from/to other modules, or to take any part of an output data and permute and copy it to other modules. In order to control its current functionality, every module has fully or partially access to the input data of the other modules depending on the tasks involved. For instance, the grasp planning module has full access to the input data of the sensory system and has partially access to the robot hand. Because the architecture of the system has a global planning to access to all the data available to the system, the grasp planning module can locally plan the grasping positions without having a global view of the robot's environment. For instance, the robot hand needs some information supplied by the global planning module such as grasp configurations for the object to be grasped.

2.1 Visual Layer: Feature Extraction

We consider visual features extraction in the context of a stereo head (see Figure 2). First, however, we recall the epipolar geometry technique which is motivated by considering the search of corresponding points in the stereo image pair. Since we are dealing with a stereo head, we need to extract features well suited for determining the grasp points on the first image either from the left or right side of a stereo

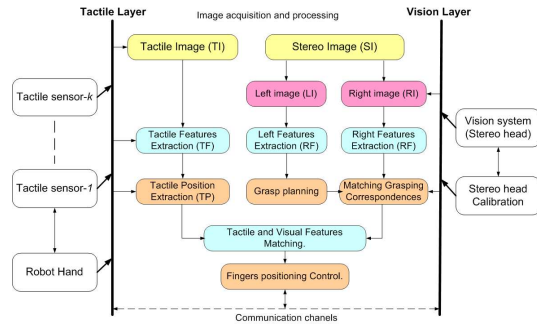


Figure 1: Grasping system description.

pair and then computing their correspondences in the second image. Given the estimated object pose, placed on the table, the full observability of the object is then projected into the left and right image planes. The visual layer takes these images and calibration data as input (see Figure 1) and provides as output a set of visual features. Segmentation is used to separate the object from the background and other objects in the environment by determining the coordinates of a closed rectangular bounding box. After segmenting the region corresponding to the object, features belonging to the object contour are extracted. A function is then constructed for parameters regrouping object features together.

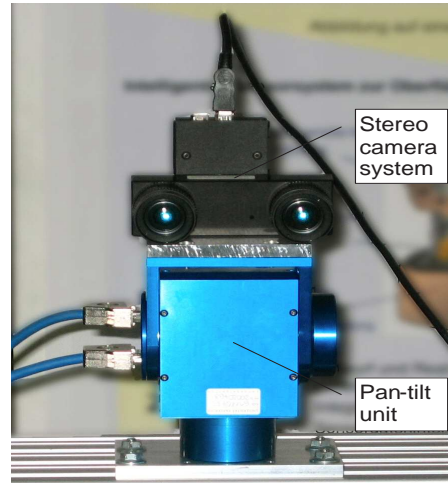


Figure 2: Stereo camera system and a Pan-tilt unit.

We denote by V a function regrouping visual parameters that is defined by

$$V = \{glist, gparam, com\} \quad (1)$$

where $glist$, $gparam$ and com are the visual features. During image processing, V is maintained as a doubly linked list of grasping region and intervening their

parameters as $g_1 g_{param_1}, \dots, g_m g_{param_m}$. A grasping region g_i is defined by its ending points g_{ui} and g_{vi+1} , and its orientation ϕ with respect to the object's center of mass com . The resulting parameters of V fully describe the two-dimensional location of features with respect to the image plane. The visual features obtained can be used as input data for both, grasp planning and grasp position matching. For more details about this topic, we refer the reader to our previous work (Boudaba and Casals, 2007).

2.2 Tactile Layer: Feature Extraction

Unlike vision which provides global features of the object, tactile sensor provides local features when the fingertip is in contact with the object. The tactile layer shown in Figure 1 takes as input the data extracted from a set of tactile sensor (or so called Group Of Tactile sensor (GOT)) and the configuration of the robot hand and provides as output a set of tactile features. To simplify the problem, tactile features are treated as visual features using the basic results from different approaches. For the purpose of sensor features matching, extracting edge features are of interest and will be discussed in section 4. Figure 3 illustrates the PCB tactile sensor module with its memory and data control units. The sensor module (from Weiss Robotics, (K.Weiss and Woern, 2005)) consists of a sensitive area organized in matrix of 4x7 sensor cells with a spacial resolution of 3.8 mm. By using four modules, (two in each gripper finger), the parallel gripper shown in Figure 3 is equipped with a total number of 112 sensor cells.

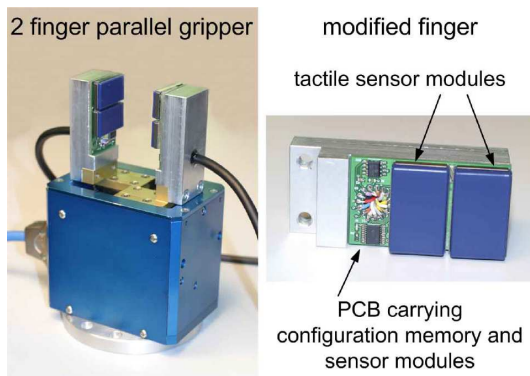


Figure 3: Gripper equipped with tactile sensor modules.

The data of the tactile sensor matrix corresponds to a two-dimensional planar image. We analyze this image using moments up to the 2nd order (Hu, 1962). The two-dimensional $(p + q)^{th}$ order moment $m_{p,q}$ of an image is defined as the following double sum over

all image pixels (x,y) and their values $f(x,y)$:

$$m_{pq} = \sum_x \sum_y f(x,y) x^p y^q \quad p, q \geq 0 \quad (2)$$

The moment $m_{0,0}$ constitutes the resulting force exerted on the sensor. The center of gravity $cog = (x_c, y_c)^T$ of this force can be computed as follows:

$$x_c = \frac{m_{10}}{m_{00}}, \quad y_c = \frac{m_{01}}{m_{00}} \quad (3)$$

The center of gravity of each tactile sensor matrix determines a contact point of the gripper.

3 GRASP POSITION MATCHING

The Grasping system can be explained in more detail through a set of tasks. In order to complete the grasp matching process, it is necessary to find the relationship between their Cartesian coordinate frames (see Figure 4).

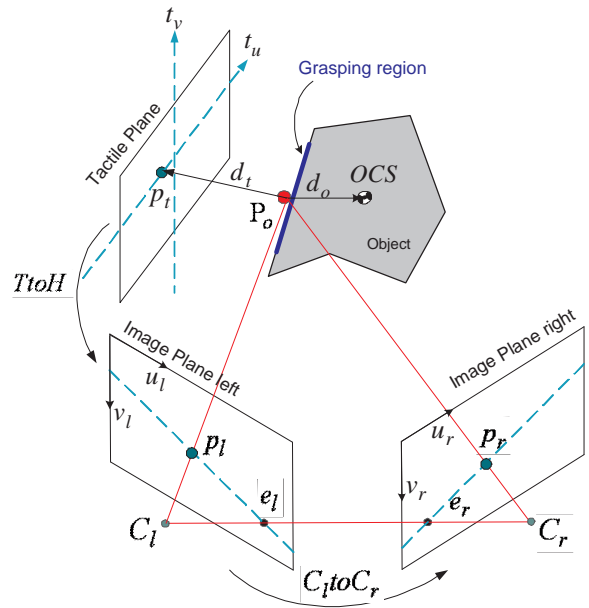


Figure 4: Sensor frames relationship for grasping.

We define these frames as follows:

- **HCS.** Head Coordinate System has a stand alone configuration of stereo head. The fixation of the head is assured by controlling the pan-tilt angle. The offset to the object coordinate system (OCS) is constant.

- **GCS.** Gripper Coordinate System (also known as the end effector frame) is determined by controlling the angular configuration of the robot arm. The robot arm moves over itself and the measurements given by all joints enable the system to determine the Tool-Center-Point (TCP) relative to the robot coordinate system.
- **OCS.** Object Coordinate System is fixed on the table and does not change its position and orientation during calibration. Once the object pose relative to the stereo head (**HCS**) is determined, the **GCS** relative to the object pose is determined by using *robot hand - stereo head* calibration.
- **TCS.** Tactile Coordinate System. The location of **TCS** is terminated by controlling the **GCS** configuration.
- **WCS.** World Coordinate System. The location of the object position and orientation **OCS**, **HCS** and the robot base station are determined relative to the **WCS**.

In the remaining of this work, the kinematics of the robot arm and robot hand are ignored. So far, for grasping using features matching, we have established two mapping relationships between feature frames. The first mapping implies finding the grasp points correspondence between left and right image of the stereo head (C_l to C_r). The second mapping implies matching the two apparent features into the tactile and stereo frames (**TtoH**).

3.1 Stereo Images Matching: C_l to C_r

We adopt some notations similar, but not identical to the work of (Hartley and Zisserman, 2000) on multiple view geometry (see Figure 4). Considering a point P_o in contact with the object, its distance to the center of mass, d_o is measured and its projection into the stereo head and tactile frames is given by (p_l, p_r) and p_t , respectively. The subscripts, o , c , l , r , and t are referred to object, contact point, left and right frames, and tactile frame, respectively.

Let $p_r = (u_r, v_r, 1)^T$ and $p_l = (u_l, v_l, 1)^T$ denote the projection of P_o on the right and left images, respectively. The epipolar plane defined by the three points P_o , C_l and C_r intersects the two image planes in two epipolar lines ep_r and ep_l . The line connecting the two centers of projection $[C_l, C_r]$ intersects the image planes at the conjugate points e_r and e_l which are called epipoles. Using the projective coordinates, the epipolar constraints can be written:

$$p_l^T \mathbf{F} p_r = 0 \quad (4)$$

where \mathbf{F} is the so-called fundamental matrix which consists of a 3×3 unknown entries and can be expressed as follows:

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \quad (5)$$

In the calibrated environment, the 9 unknown entries of \mathbf{F} can be captured in an algebraic representation as defined by

$$\mathbf{F} = \mathbf{C}_r^{-T} \mathbf{E} \mathbf{C}_l^{-1} \quad (6)$$

where the fundamental matrix \mathbf{F} encapsulates both the intrinsic and the extrinsic parameters of the stereo head, while the essential matrix $\mathbf{E} = [\mathbf{T}]_{\times} \mathbf{R}$ which compactly encodes the extrinsic parameters of the stereo head can be composed of the baseline vector $\mathbf{t} = [C_r - C_l] = (t_x, 0, t_z)^T$ and the angular rotation β about the y -axis that renders the left image parallel to the right one, then we have:

$$\mathbf{E} = \begin{bmatrix} 0 & -t_z & 0 \\ t_z & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix} \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (7)$$

C_l and C_r are the intrinsic parameter matrices of the left and right cameras defined by

$$\mathbf{C}_l = \begin{bmatrix} f_{ul} & 0 & C_{ul} \\ 0 & f_{vl} & C_{vl} \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{C}_r = \begin{bmatrix} f_{ur} & 0 & C_{ur} \\ 0 & f_{vr} & C_{vr} \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

where u_{0l} and v_{0l} (resp. u_{0r} and v_{0r}) are the coordinates of the principle point (in pixels) of the left (resp. right) camera. (f_x, f_y) are the focal length in x and y direction.

For more details about camera calibration and related topics, we refer to the work of (Faugeras and Toscani, 1986) and (Tsai and Lenz, 1989).

3.2 Tactile and Stereo Matching: **TtoH**

By dealing with the contact constraint, the minimum distance between a fingertip (tactile) and the object can be expressed by a parameter d_t . So keeping a fingertip in touch with the object, the condition $d_t = 0$ must be maintained, and tactile features are extracted and measured into the tactile frames. Matching these tactile features with visual features implies the computation of a similarity transformation relating the

grasping region to actual sensitive touching area. Tactile and visual features are then related by the following transformation

$$s_i = \mathbf{T}t\mathbf{o}\mathbf{H}v_i \quad (9)$$

where s_i and v_i are points on the tactile and visual image features, respectively. $\mathbf{T}t\mathbf{o}\mathbf{H}$ is the similarity transformation given by

$$\mathbf{T}t\mathbf{o}\mathbf{H} = \begin{bmatrix} s \cos \alpha & -\sin \alpha & t_x \\ \sin \alpha & s \cos \alpha & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

where s , α , and $[t_x, t_y, 1]$ are scaling, rotation angle and translation vector of the tactile image with respect to the visual image, respectively.

In the calibration cases, the parameters of (10) can be computed directly using homogeneous transformation matrices between frames as shown in Figure 4.

4 IMPLEMENTATION

The implementation of our algorithm for grasping position matching using stereo vision and tactile sensor can be divided into two parts. First is related to the grasp planning using stereo vision. The second part of this implementation is related features matching between stereo vision and tactile sensor.

4.1 Grasp Planning using a Stereo Head

As stated before, the first implementation consists of computing the grasp points correspondence in the stereo vision. More formally, let $G = \{G_{v_1}, G_{v_2}, \dots, G_{v_k}\}$ be a set of valid grasps and $G_{v_i}^l = (g_{u_i}^l, g_{v_i}^l, 1)^T$ be its i th-determined grasp point on the left image, next step is to compute its correspondence on the right image, $G_{v_i}^r = (g_{u_i}^r, g_{v_i}^r, 1)^T$. To do this, we first need to establish a mapping relationship between a line and point by exploiting the epipolar constraint defined by (4).

Let

$$L_{r_i} = \mathbf{F}G_{v_i}^l, \quad L_{l_i} = \mathbf{F}^T G_{v_i}^r \quad (11)$$

be the mapping equations where \mathbf{F}^T is the transpose of \mathbf{F} and $i := 1, 2, \dots, k$ is the number of grasping points. $G_{v_i}^l$ (resp. $G_{v_i}^r$) is the determined grasp point

on the left (resp. right) image and L_{r_i} (resp. L_{l_i}) is its corresponding epipolar line on the right (resp. left) image. By exploiting the epipolar constraint (4), the grasping points are constrained to lie along the epipolar lines L_{r_i} and L_{l_i} , respectively.

If both grasping points satisfy the relation $G_{v_i}^l \mathbf{F} G_{v_i}^r = 0$ then the lines defined by these points are coplanar. This is a necessary condition for the grasp points to correspond.

Given the parameters of a line and a grasp point in one image, the maximum deviation of a point from the line can be computed as follows:

$$d_{l_i}^2 = \text{norm}(L_{l_i}, G_{v_i}^l), \quad d_{r_i}^2 = \text{norm}(L_{r_i}, G_{v_i}^r) \quad (12)$$

where $d_{l_i}^2$ (resp. $d_{r_i}^2$) is the maximum deviation of a grasp point on the left (resp. right) image.

We can estimate a cost function with respect to a parameter t as follows:

$$C(t) = d_{l_i}^2 + d_{r_i}^2 \quad (13)$$

The minimum threshold (t_{min}) corresponds to the t_i where the cost function is minimum.

4.2 Features Matching

The second implementation consists of computing the similarity between the stereo and the tactile images features. To compare image features, the Hausdorff metric based on static features matching is used (Huttenlocher et al., 1993).

Given two feature sets: $\mathbf{S} = \{s_1, s_2, \dots, s_q\}$ and $\mathbf{V} = \{v_1, v_2, \dots, v_q\}$, the Hausdorff distance from the point set \mathbf{S} to point set \mathbf{V} is defined as

$$h(\mathbf{S}, \mathbf{V}) = \max_{s_i \in \mathbf{S}} \min_{v_j \in \mathbf{V}} \|s_i - v_j\| \quad (14)$$

where $\|s_i - v_j\|$ corresponds to the sum of the pixel difference and indices i and j correspond to the size of a searching window.

The matching process is evaluated according to the output of the function (14). The matching that results in the lowest cost is the one that matches the closest grasp planning. Since we want to guide the gripper toward the grasping points previously generated by the grasp planning, the solution consists of reducing the cost function (or so called grasp error) by

Table 1: Parameters measure of grasping positions and cost function: obj1, obj2.

Threshold: $t = 2$ corresponds to the maximum deviation.							
Object	Left grasp	(x_l, y_l)	d_l	Right grasp	(x_r, y_r)	d_r	C
obj1	$G_{v_1}^l$	358.500 365.500	2.934	$G_{v_1}^r$	313.500 377.000	2.940	17.252
	$G_{v_2}^l$	359.000 400.000	5.008	$G_{v_1}^r$	312.500 411.500	5.018	50.261
obj2	$G_{v_1}^l$	355.000 226.000	1.726	$G_{v_1}^r$	299.500 237.500	1.731	5.9768
	$G_{v_2}^l$	363.000 314.500	0.922	$G_{v_1}^r$	309.000 323.500	0.926	1.7028
	$G_{v_3}^l$	309.500 316.500	5.251	$G_{v_1}^r$	255.000 321.500	5.259	55.234

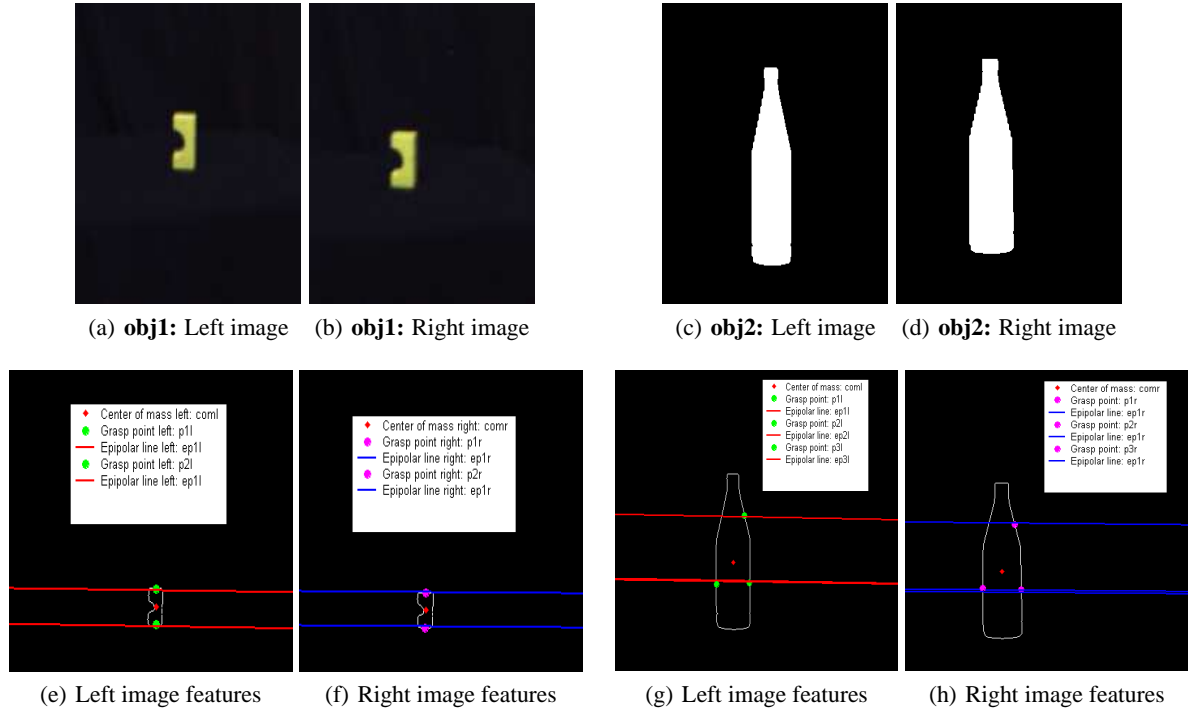


Figure 5: Result of two and three-fingered grasp planning algorithms using stereo images.

moving the tactile sensors toward these points. The cost of a solution is expressed as the total sum of contact displacements over the surface of the object from an initial contact configuration. If the result of matching is outside a given margin, then the grasp controller should launch a new measurement via joint angle and position sensors.

4.3 Matching Algorithm

- **Input:** images: im_1, im_2 . features: V_1, V_2 . Number of fingers: k . Fundamental matrix: \mathbf{F} . Threshold t . Size of window: 7×7 pixel.
- **Output:** Grasping points: $(G_{v_i}^l, G_{v_i}^r)$, with $i := 1, \dots, k$. Matching: $h(\mathbf{S}, \mathbf{V})$

■ Process:

1. Perform the features extraction tasks

- for $i := 1$ to 2 do
 - extract features: $V_i := im_i$.
2. Perform the grasp planning tasks
- select V_1 on which the grasp will be performed.
 - Get valid grasping point G_v from (1)
3. Perform the grasping point correspondences
- select V_2 on which the grasping correspondences will be performed
 - for $i := 1$ to k do
 - Compute $G_{v_i}^l := G_{v_i}$
 - Compute L_{r_i} and $G_{v_i}^r$ using (11)
4. Perform the matching function
- for $i, j := 1$ to 7, 7 do
 - Compute $h(\mathbf{S}, \mathbf{V}) \leq r$ and $h(\mathbf{S}, \mathbf{V}) \geq r$
 - Compute pixel difference $h(\mathbf{S}, \mathbf{V})$ from (14)

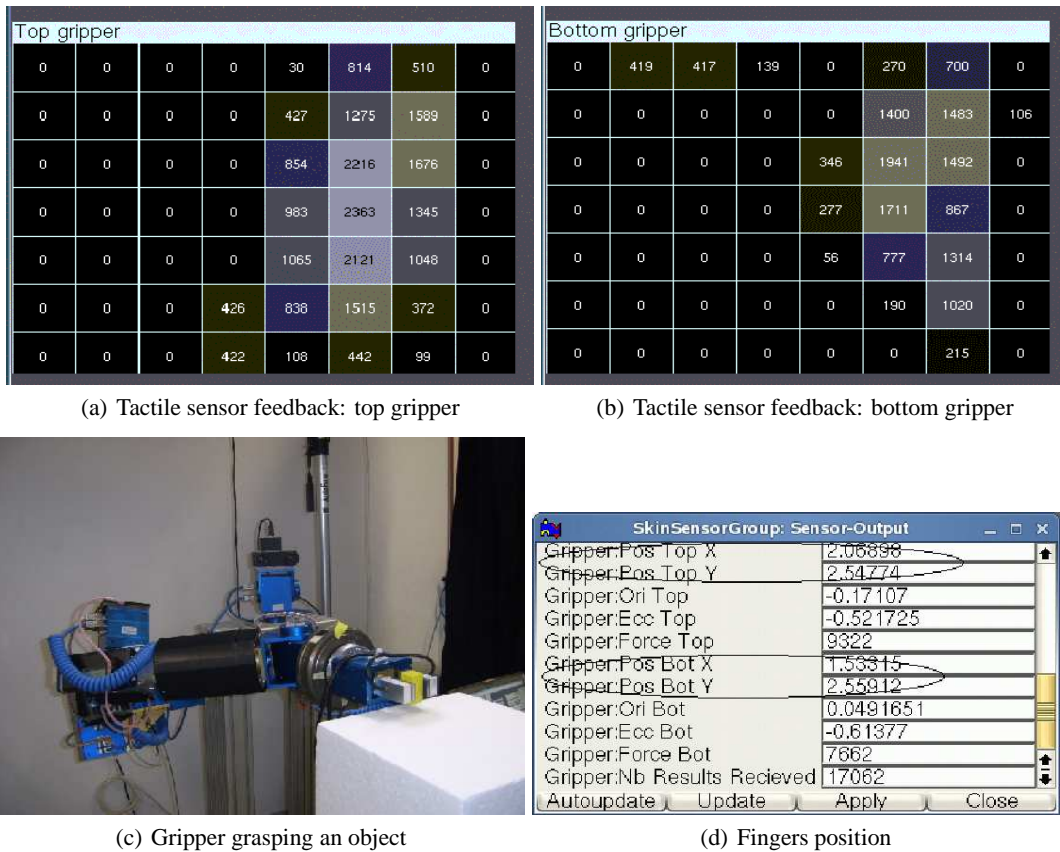


Figure 6: The Experiment setup. (a)-(b) Tactile sensor feedback giving the sensitive area in contact with the object. (c) Object grasped with two fingers parallel gripper. (d) Tactile sensor output giving the top/bottom position of the gripper.

5. Perform the cost function

- Compute d_i^2 and $d_{r_i}^2$ using (12)
- Compute $C(t)$ with (13)

6. End.

4.4 Experimental Results

The algorithm was implemented on our experimental system, which consists of a 7 DOF manipulator arm, a robot hand with two fingers, each one equipped with a tactile sensor module mounted directly to the finger tip, and a vision system (see Figure 6). This first prototype of anthropomorphic robot system developed by the German Research Foundation (see (Boudaba et al., 2005)) is used as platform and demonstrator for a coming generation of service robots. The grasping configuration is based on a stand-alone stereo head (MEGA-D from Videre Design) mounted on a pan-tilt controller unit equipped with a pair of 4.8 mm lenses and a fixed baseline of about 9cm. We have experimented our approach with two different kind of objects placed on a fixed table with a fixed position

and orientation (static object). Figure 5 illustrates the results obtained from our matching algorithm using stereo vision. The performance of our results (see Table 1) is validated according to a cost function C defined in the stereo images as the errors between grasping points. The cost that results in the lower value is the one that matches the closest grasp planning. Figures 6(b)-(d) illustrate the feedback of tactile sensor giving the top/bottom position of fingers with respect to the tactile image plane in (b) while (c)-(d) showing the top/bottom sensitive area in touch with the object.

5 CONCLUSIONS

The implementation of our algorithms for grasping points matching using stereo vision and tactile sensor have been detailed. Two schemes for grasping points matching have been included in this work. In the first scheme, stereo vision matching was used to find the grasp points correspondence between left and right images. It is shown that the quality of matching

depends strongly on the precise computation of the intrinsic and extrinsic parameters of the stereo head calibration. The performance of our results is evaluated according to a cost function defined in the stereo images as the errors between a pair of grasping points. In the second scheme, the tactile sensor provides the sensitive area of a fingertip in contact with an object which was used with the grasp region to compute the similarity between both features. Using these two matching schemes, we were able to fuse the visual grasp region with the tactile features and capabilities of reducing or avoiding the grasp positioning errors (or so called controlling the grasp planning) before executing a grasps.

REFERENCES

- Allen, P., Miller, A., Oh, P., and Leibowitz, B. (1999). Integration vision, force and tactile sensing for grasping. *Int. Journal of Intell. Mechatronics*, 4(1):129–149.
- Berger, A. D. and Khosla, P. K. (1991). Using tactile data for real-time feedback. *International Journal of Robotics Research (IJR'91)*, 2(10):88–102.
- Boudaba, M. and Casals, A. (2006). Grasping of planar objects using visual perception. In *Proc. IEEE 6th International Conference on Humanoid Robots (HUMANOIDS'06)*, pages 605–611, Genova, Italy.
- Boudaba, M. and Casals, A. (2007). Grasp configuration matching using tactile and visual information. In *Proc. IEEE 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO'07)*, volume Vol. II, pages 121–127, Angers, France.
- Boudaba, M., Casals, A., Osswald, D., and Woern, H. (2005). Vision-based grasping point determination on objects grasping by multifingered hands. In *Proc. IEEE 6th International Conference on Field and Service Robotics*, pages 261–272, Australia.
- Chen, N., Rink, R. E., and Zhang, H. (1995). Edge tracking using tactile servo. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 84–99.
- Faugeras, O. D. and Toscani, G. (1986). The calibration problem for stereo. In *In Proc. IEEE Conference on Computer Vision and Pattern Recognition, CVPR'86*, pages 15–20, Miami Beach, FL, USA.
- Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, England, UK.
- Hu, M.-K. (1962). Visual Pattern Recognition by Moment Invariants. *IEEE Transactions on Information Theory*, 8(2):179–187.
- Huttenlocher, D., Klanderma, D., and Rucklidge, A. (1993). Comparing images using the hausdorff distance. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15(9):850–863.
- Kragic, D., Miller, A., and Allen, P. (2001). Real-time tracking meets online grasp planning. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'2001)*, pages 2460–2465, Seoul, Korea.
- K.Weiss and Woern, H. (2005). The working principle of resistive tactile sensor cells. In *Proceedings of the IEEE Int. Conf. on Mechatronics and Automation*, pages 471–476, Ontario, Canada.
- Lee, M. H. and Nicholls, H. R. (2000). Tactile sensing for mechatronics - a state of the art survey. *Mechatronics*, 9:1–31.
- Maekawa, H., Tanie, K., and Komoriya, K. (1995). Tactile sensor based manipulation of an unknown object by a multifingered hand with rolling contact. In *Proc. IEEE International Conference on Robotics and Automation*, pages 743–750.
- Morales, A., Sanz, P., and del Pobil, A. (2002). Heuristic vision-based computation of three-finger grasps on unknown planar objects. In *Proceedings of the IEEE/RST Int. Conf. on Intelligent Robots and Systems*, pages 1693–1698, Lausanne, Switzerland.
- Namiki, A. and Ishikawa, M. (1999). Optimal grasping using visual and tactile feedback. In *Proceedings of the IEEE/RST Int. Conf. on Intelligent Robots and Systems*, pages 589–596.
- Perrin, D., Smith, C., Masoud, O., and Papanikolopoulos, N. (2000). Unknown object grasping using pressure models. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 575–582.
- Sanz, P., del Pobil, A., Iesta, J., and Recatal, G. (1998). Vision-guided grasping of unknown objects for service robots. In *ICRA'98*, page 30183025, Leuven, Belgium.
- Scholl, K. U., Albiez, J., and Gassmann, B. (2001). MCA - An Expandable Modular Controller Architecture. *3rd Real-Time Linux Workshop*.
- Smith, C. and Papanikolopoulos (1996). Vision-guided robotic grasping: Issues and experiments. In *ICRA'96*, pages 3203–3208.
- Tsai, R. Y. and Lenz, R. K. (1989). A new techniques for fully autonomous and efficient 3d robot hand/eye calibration. *IEEE Trans. on Robotics and Automation*, 5(3):345–358.
- Yoshimi, B. H. and Allen, P. K. (1994). Visual control of grasping and manipulation tasks. In *Proc. IEEE International Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 575–582, Las Vegas, USA.

TRAFFIC SIGN RECOGNITION WITH CONSTELLATIONS OF VISUAL WORDS

Toon Goedemé

*De Nayer Instituut, Jan De Nayerlaan 5, 2860 Sint-Katelijne-Waver, Belgium
Toon.Goedeme@denayer.wenk.be*

Keywords: Traffic sign recognition, local features, SURF, embedded.

Abstract: In this paper, we present a method for fast and robust object recognition. As an example, the method is applied to traffic sign recognition from a forward-looking camera in a car. To facilitate and optimise the implementation of this algorithm on an embedded platform containing parallel hardware, we developed a voting scheme for constellations of visual words, i.e. clustered local features (SURF in this case). On top of easy implementation and robust and fast performance, even with large databases, an extra advantage is that this method can handle multiple identical visual features in one model.

1 INTRODUCTION

One of the key upcoming technologies making cars safer to drive with is automatic recognition of road signs. An on-board camera installed in the car observes the road ahead. Intelligent computer vision algorithms are being developed that enable the detection and recognition of various objects in these images: traffic lane markings, pedestrians, obstacles, ... Here, we focus on the recognition of traffic signs. Based on recognition results, certain alerts can be sent to the driver, e.g. a warning if the current speed of the car is higher than the speed limit declared in the traffic sign. Because that the appearance of a certain traffic sign is fixed (even described by law), the detection is quite a bit easier than e.g. the detection of pedestrians. Nevertheless, in real-life experiments substantial appearance variation is measured, mainly due to different light conditions, viewpoint changes, ageing of the traffic sign, deformations and even vandalism. We can conclude that a robust method is needed.

The remainder of this text is organised as follows. Section 2 gives an overview of relevant related work. In section 3, our algorithm is described. Some *real life* experiments are presented in section 4. The paper ends with a conclusion in section 5.

2 RELATED WORK

Real-time road sign recognition has been a research topic for many years. This problem is often addressed

in a two-stage procedure involving detection and classification. In contrast, our solution is an all-in-one operation which more likely leads to a faster algorithm. An other difference with related work in the field is that our solution does not rely on template matching (Rosenfeld and Kak, 1976), colour (Zhu and Liu, 2006), the detection of geometrical basis shapes (Garcia-Garrido et al., 2006), or canny edges (Sandoval et al., 2000). Moreover, our solution is not limited to certain traffic sign shapes (Ballerini et al., 2005). We use clusters of local image features (SURF) to robustly describe the appearance of the traffic sign.

A few years ago, a major revolution in the object recognition field was the appearance of the idea of local image features (Tuytelaars et al., 1999; Lowe, 1999). Indeed, looking at local parts instead of the entire pattern to be recognised has the inherent advantage of robustness to partial occlusions. In both template and query image, local regions are extracted around interest points, each described by a descriptor vector for comparison. The development of robust local feature descriptors, like e.g. Mindru's generalised colour moment based ones (Mindru et al., 1999), added robustness to illumination and changes in viewpoint.

Many researchers proposed algorithms for local region matching. The differences between approaches lie in the way in which interest points, local image regions, and descriptor vectors are extracted. An early example is the work of Schmid and Mohr (Schmid et al., 1997), where geometric invariance

was still under image rotations only. Scaling was handled by using circular regions of several sizes. Lowe *et al.* (Lowe, 1999) extended these ideas to real scale-invariance. More general affine invariance has been achieved in the work of Baumberg (Baumberg, 2000), that uses an iterative scheme and the combination of multiple scales, and in the more direct, constructive methods of Tuytelaars & Van Gool (Tuytelaars *et al.*, 1999; Tuytelaars and Gool, 2000), Matas *et al.* (Matas *et al.*, 2002), and Mikolajczyk & Schmid (Mikolajczyk and Schmid, 2002). Although these methods are capable to find very qualitative correspondences, most of them are too slow for use in a real-time application as the one we envision here. Moreover, none of these methods are especially suited for the implementation on an embedded computing system, where both memory and computing power must be as low as possible to ensure reliable operation at the lowest cost possible.

The classic recognition scheme with local features, presented in (Lowe, 1999; Tuytelaars and Gool, 2000), and used in many applications such as in our previous work on robot navigation (Goedemé *et al.*, 2005; Goedemé *et al.*, 2006), is based on finding one-on-one matches. Between the query image and a model image of the object to be recognised, bijective matches are found. For each local feature of the one image, the most similar feature in the other is selected.

This scheme contains a fundamental drawback, namely its disability to detect matches when multiple identical features are present in an image. In that case, no guarantee can be given that the most similar feature is the correct correspondence. Such pattern repetitions are quite common in the real world, though, especially in man-made environments. To reduce the number of incorrect matches due to this phenomenon, in classic matching techniques a criterion is used such as comparing the distance to the most and the second most similar feature (Lowe, 1999). Of course, this practice throws away a lot of good matches in the presence of pattern repetitions.

In this paper, we present a possible solution to this problem by making use of the *visual word* concept. Visual words are introduced (Sivic and Zisserman, 2003; Li and Perona, 2005; Zhang *et al.*, 2005) in the context of object classification. Local features are grouped into a large number of clusters with those with similar descriptors assigned into the same cluster. By treating each cluster as a *visual word* that represents the specific local pattern shared by the keypoints in that cluster, we have a visual word vocabulary describing all kinds of such local image patterns. With its local features mapped into visual words, an

image can be represented as a *bag of visual words*, as a vector containing the (weighted) count of each visual word in that image, which is used as feature vector in the classification task.

In contrast to the in categorisation often used bag-of-words concept, in this paper we present the *constellation-of-words* model. The main difference is that not only the presence of a number of visual words is tested, but also their relative positions.

3 ALGORITHM

Fig. 1 gives an overview of the algorithm. It consists of two phases, namely the model construction phase (upper row) and the matching phase (bottom row).

First, in a model photograph (*a*), local features are extracted (*b*). Then, a vocabulary of visual words is formed by clustering these features based on their descriptor. The corresponding visual words on the image (*c*) are used to form the model description. The relative location of the image centre (the *anchor*) is stored for each visual word instance (*d*).

The bottom row depicts the matching procedure. In a query image, local features are extracted (*e*). Matching with the vocabulary yields a set of visual words (*f*). For each visual word in the model description, a vote is cast at the relative location of the anchor location (*g*). The location of the object can be found based on these votes as local maxima in a voting Hough space (*h*). Each of the following subsections describes one step of this algorithm in detail.

3.1 Local Feature Extraction

We chose to use SURF as local feature detector, instead of the often used SIFT detector. SURF (Bay *et al.*, 2006; Fasel and Gool, 2007) is developed to be substantially faster, but at least as performant as SIFT.

3.1.1 Interest Point Detector

In contrast to SIFT (Lowe, 1999), which approximates Laplacian of Gaussian (LoG) with Difference of Gaussians (DoG), SURF approximates second order Gaussian derivatives with box filters, see fig. 2. Image convolutions with these box filters can be computed rapidly by using integral images as defined in (Viola and Jones, 2001). Interest points are localised in scale and image space by applying a non-maximum suppression in a 3×3 neighbourhood. Finally, the found maxima of the determinant of the approximated Hessian matrix are interpolated in scale and image space.

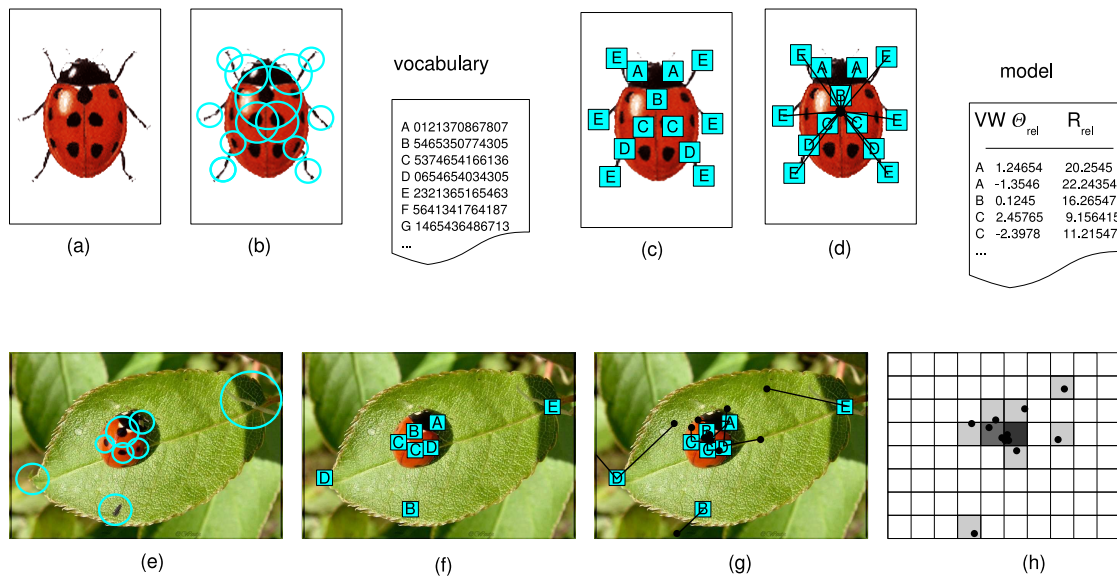


Figure 1: Overview of the algorithm. Top row (model building): (a) model photo, (b) extracted local features, (c) features expressed as visual words from the vocabulary, (d) model description with relative anchor positions for each visual word. Bottom row (matching): (e) query image with extracted features, (f) visual words from the vocabulary, (g) anchor position voting space based on relative anchor position, (h) Hough voting space.



Figure 2: Left: two filters based on Gaussian derivatives. Right: their approximation using box filters.

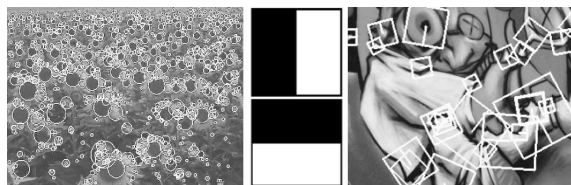


Figure 3: Middle: Haar wavelets. Left and right: examples of extracted SURF features.

3.1.2 Descriptor

In a first step, SURF constructs a circular region around the detected interest points in order to assign a unique orientation to the former and thus gain invariance to image rotations. The orientation is computed using Haar wavelet responses in both x and y direction as shown in the middle of fig. 3. The Haar wavelets can be easily computed via integral images, similar to the Gaussian second order approximated box filters. Once the Haar wavelet responses are computed, they are weighted with a Gaussian centred at the in-

terest points. In a next step the dominant orientation is estimated by summing the horizontal and vertical wavelet responses within a rotating wedge, covering an angle of $\frac{\pi}{3}$ in the wavelet response space. The resulting maximum is then chosen to describe the orientation of the interest point descriptor. In a second step, the SURF descriptors are constructed by extracting square regions around the interest points. These are oriented in the directions assigned in the previous step. Some example windows are shown on the right hand side of fig. 3. The windows are split up in 4×4 sub-regions in order to retain some spatial information. In each sub-region, Haar wavelets are extracted at regularly spaced sample points. In order to increase robustness to geometric deformations and localisation errors, the responses of the Haar wavelets are weighted with a Gaussian, centred at the interest point. Finally, the wavelet responses in horizontal d_x and vertical directions d_y are summed up over each sub-region. Furthermore, the absolute values $|d_x|$ and $|d_y|$ are summed in order to obtain information about the polarity of the image intensity changes. The resulting descriptor vector for all 4×4 sub-regions is of length 64. See fig. 4 for an illustration of the SURF descriptor for three different image intensity patterns.

More details about SURF can be found in (Bay et al., 2006) and (Fasel and Gool, 2007).

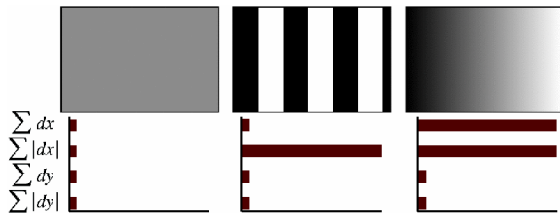


Figure 4: Illustrating the SURF descriptor.

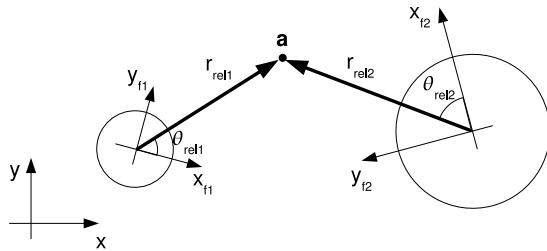


Figure 5: The position of the anchor point is stored in the model as polar coordinates relative to the visual word scale and orientation.

3.1.3 Visual Words

As explained before, the next step is forming a vocabulary of visual words. This is accomplished by clustering a big set of extracted SURF features. It is important to build this vocabulary using a large number of features, in order to be representative for all images to be processed.

The clustering itself is easily carried out with the k-means algorithm. Distances between features are computed as the Euclidean distance between the corresponding SURF descriptors. Keep in mind that this model-building phase can be processed off-line, the real-time behaviour is only needed in the matching step.

In the fictive ladybug example of fig. 1, each visual word is symbolically presented as a letter. It can be seen that the vocabulary consists of a file linking each visual word symbol with a mean descriptor vector of the corresponding cluster.

3.2 Model Construction

All features found on a model image are matched with the visual word vocabulary, as shown in fig. 1 (c). In addition to the popular bag-of-words models, which consist of a set of visual words, we add the relative constellation of all visual words to the model description.

Each line in the model description file consists of the symbolic name of a visual word, and the rela-

tive coordinates (r_{rel}, θ_{rel}) to the anchor point of the model item. As anchor point, we chose for instance the centre of the model picture. These coordinates are expressed as polar coordinates, relative to the individual axis frame of the visual word. Indeed, each visual word in the model photograph has a scale and an orientation because it is extracted as a SURF feature. Fig. 5 illustrates this. The resulting model is a very compact description of the appearance of the model photo. Many of these models, based on the same visual word vocabulary, can be saved in a compact database. In our traffic sign recognition application, we build a database of all different traffic signs to be recognised.

3.3 Matching

This part of the algorithm is time-critical. We are spending lots of efforts in speeding up the matching procedure, in order to be able to implement it on an embedded system.

The first operation carried out on incoming images is extracting SURF features, exactly as described in section 3.1. After local feature extraction, matching is performed with the visual words in the vocabulary. We used Mount's ANN (Approximate Nearest Neighbour) (Arya et al., 1998) algorithm for this, which is very performant. As seen in fig. 1 (f), some of the visual words of the object are recognised, amidst other visual words.

3.3.1 Anchor Location Voting

Because each SURF feature has a certain scale and rotation, we can reconstruct the anchor pixel location by using the feature-relative polar coordinates of the object anchor. For each instance in the object model description, this yields a vote for a certain anchor location. In fig. 1 (g), this is depicted by the black lines with a black dot at the computed anchor location.

Ideally, all these locations would coincide at the correct object centre. Unfortunately, this is not the case due to mismatches and noise. Moreover, if there are two identical visual words in the model description of an object (as in the ladybug example for words A, C and D), each detected visual word of that kind in the query image will cast to different anchor location votes, of which only one can be correct.

3.3.2 Object Detection

For all different models in the database, anchor location votes can be quickly computed. Next task is to decide where a certain object is detected. Because a certain object can be present more than once in the



Figure 6: Some model photos from the traffic sign library.

query image, it is clear that a simple average of the anchor position votes is not a sufficient technique, even if robust estimators like RANSAC are used to eliminate outliers. Therefore, we construct a *Hough space*, a matrix which is initiated at zero and incremented at each anchor location vote, fig. 1 (*h*). The local maxima of the resulting Hough matrix are computed and interpreted as detected object positions.

4 EXPERIMENTS

For preliminary experiments, we implemented this algorithm using Octave and an executable of the SURF extractor. Fig. 7 shows some typical results of different phases of the algorithm. The test images were acquired by taking 640×480 digital photographs at random natural road scenes.

In fig. 6, a number of model photographs are shown. Each of such images, having a resolution of about 100×100 pixels, yielded a thorough description of the traffic sign design in a model description containing on the average 52 features, what boils down to a model file size of only 3.2 KB.

Fig. 7 shows a typical output during the detection stage. In a query image, local features are extracted. These are matched with the visual word vocabulary. Then, for each traffic sign model, for the matching visual words the relative anchor position is computed. This stage is visualised in the figure. As can be noticed, in the centre of the traffic sign, many anchor votes are coinciding.

The traffic signs were detected by finding local maxima in the Hough space, for this example visualised in fig. 8. We performed experiments on 35 query images and were able to detect 82% of the trained types. Detection failures were mostly due to the fact that the signs were too far away and hence too small, and severe occlusions by other objects.



Figure 7: Typical experimental results. In a query image, all matching visual words are shown for the *pedestrian crossing* sign (white squares). From each visual word, the anchor location is computed (black line pointing from visual word centre to anchor). The zoom on the right shows clearly the recognised sign: many anchor lines coincide.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an algorithm for object detection based on the concept of visual word constellation voting. The preliminary experiments proved the performance of this approach. The method has the advantages that it is computing power and memory efficient and that it can handle pattern repetitions in the models.

We applied this method successfully on automatic traffic sign recognition.

As told before, our aim in this work is an embedded implementation of this algorithm. The Octave implementation presented here is only a first step towards that. But we believe the proposed approach has a lot of advantages. The SURF extraction phase can mostly be migrated to a parallel hardware imple-

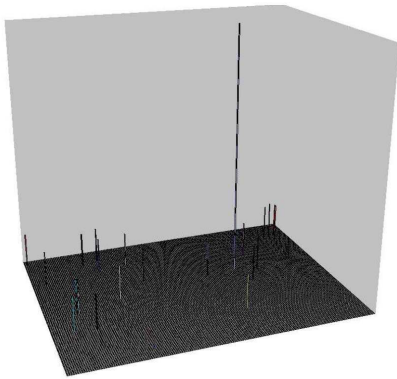


Figure 8: Hough space for the anchor positions of the example in fig. 7.

mentation on FPGA. Visual word matching is sped up using the ANN-libraries, making use of Kd-trees. Of course a large part of the memory is used by the (mostly sparse) hough space. A better description of the voting space will lead to a great memory improvement of the algorithm.

REFERENCES

- Arya, S., Mount, D., Netanyahu, N., Silverman, R., and Wu, A. (1998). An optimal algorithm for approximate nearest neighbor searching. In *J. of the ACM*, vol. 45, pp. 891-923.
- Ballerini, R., Cinque, L., Lombardi, L., and Marmo, R. (2005). *Rectangular Traffic Sign Recognition*. Springer Berlin / Heidelberg.
- Baumberg, A. (2000). Reliable feature matching across widely separated views. In *Computer Vision and Pattern Recognition, Hilton Head, South Carolina*, pp. 774-781.
- Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Speeded up robust features. In *ECCV*.
- Fasel, B. and Gool, L. V. (2007). Interactive museum guide: Accurate retrieval of object descriptions. In *Adaptive Multimedia Retrieval: User, Context, and Feedback, Lecture Notes in Computer Science, Springer, volume 4398*.
- Garcia-Garrido, M., Sotelo, M., and Martin-Gorostiza, E. (2006). Fast traffic sign detection and recognition under changing lighting conditions. In *Intelligent Transportation Systems Conference 2006*, pp. 811 - 816.
- Goedemé, T., Tuytelaars, T., Nuttin, M., and Gool, L. V. (2006). Omnidirectional vision based topological navigation. In *International Journal of Computer Vision and International Journal of Robotics Research, Special Issue: Joint Issue of IJCV and IJRR on Vision and Robotics*.
- Goedemé, T., Tuytelaars, T., Vanacker, G., Nuttin, M., and Gool, L. V. (2005). Feature based omnidirectional sparse visual path following. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2005, Edmonton*, pages 1003–1008.
- Li, F.-F. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Proc. of the 2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 524531.
- Lowe, D. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer Vision*.
- Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference, Cardiff, Wales*, pp. 384-396.
- Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In *ECCV, vol. 1*, 128–142.
- Mindru, F., Moons, T., and Gool, L. V. (1999). Recognizing color patterns irrespective of viewpoint and illumination. In *Computer Vision and Pattern Recognition, vol. 1*, pp. 368-373.
- Rosenfeld, A. and Kak, A. (1976). Digital picture processing. In *Computer Science and Applied Mathematics, Academic Press, New York*.
- Sandoval, H., Hattori, T., Kitagawa, S., and Chigusa, Y. (2000). Angle-dependent edge detection for traffic signs recognition. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, pp. 308-313.
- Schmid, C., Mohr, R., and Bauckhage, C. (1997). Local grey-value invariants for image retrieval. In *International Journal on Pattern Analysis and Machine Intelligence, Vol. 19, no. 5*, pp. 872-877.
- Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Proc. of 9th IEEE Intl Conf. on Computer Vision, Vol. 2*.
- Tuytelaars, T. and Gool, L. V. (2000). Wide baseline stereo based on local, affinely invariant regions. In *British Machine Vision Conference, Bristol, UK*, pp. 412-422.
- Tuytelaars, T., Gool, L. V., D'haene, L., and Koch, R. (1999). Matching of affinely invariant regions for visual servoing. In *Intl. Conf. on Robotics and Automation*, pp. 1601-1606.
- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*.
- Zhang, M., Marszalek, S., Lazebnik, S., and Schmid, C. (2005). Local features and kernels for classification of texture and object categories: An in-depth study. In *Technical report, INRIA*.
- Zhu, S. and Liu, L. (2006). Traffic sign recognition based on color standardization. In *IEEE International Conference on Information Acquisition 2006*, pp. 951-955, Weihai, China.

A KNOWLEDGE-BASED COMPONENT FOR HUMAN-ROBOT TEAMWORK

Pedro Santana, Luís Correia

*LabMAG, Computer Science Department, University of Lisbon, Portugal
m35353@alunos.di.fc.ul.pt, Luis.Correia@di.fc.ul.pt*

Mário Salgueiro, Vasco Santos

*R&D Division, IntRoSys, S.A., Portugal
mjs@introsys.eu, vps@introsys.eu*

José Barata

*UNINOVA, New University of Lisbon, Portugal
jab@uminova.pt*

Keywords: Human-robot teamwork, multi-agent systems, cooperative workflows, knowledge-based systems, demining.

Abstract: Teams of humans and robots pose a new challenge to teamwork. This stems from the fact that robots and humans have significantly different perceptual, reasoning, communication and actuation capabilities. This paper contributes to solving this problem by proposing a knowledge-based multi-agent system to support design and execution of stereotyped (i.e. recurring) human-robot teamwork. The cooperative workflow formalism has been selected to specify team plans, and adapted to allow activities to share structured data, in a frequent basis, while executing. This novel functionality enables tightly coupled interactions among team members. Rather than focusing on automatic teamwork planning, this paper proposes a complementary and intuitive knowledge-based solution for fast deployment and adaptation of small scale human-robot teams. In addition, the system has been designed in order to provide information about the mission status, contributing this way to the human overall mission awareness problem. A set of empirical results obtained from simulated and real missions demonstrates the capabilities of the system.

1 INTRODUCTION

As highly appealing the idea of humans and robots enrolling in teamwork might seem, their significantly different perceptual, reasoning, and actuation capabilities make the task a daunting one. Typically, human-robot teamwork (Tambe, 1997; Scerri et al., 2002; Sierhuis et al., 2005; Nourbakhsh et al., 2005; Sycara and Sukthankar, 2006) solutions grow from work on multi-robot and multi-agent systems adapted to include humans. This paper proposes to see the problem from the other end, i.e. to include robots as participants on human-centred operational procedures, supported by knowledge management concepts usual in human organisations. Both views are complementary rather than mutually exclusive.

Human teamwork operational procedures are typically knowledge intensive tasks (Schreiber et al., 2000), which can be approximately represented by a set of templates. Knowledge engineering methodolo-

gies can be used to grasp and formalise domain experts' knowledge into the form of templates. These templates specify stereotyped (i.e. recurring) team plans, which need to be adapted to the situation at hand. Here, much of the work on automatic teamwork (re)planning (Sycara and Sukthankar, 2006) can be employed. Nonetheless, visual interfaces through which humans can manually adapt the plan are paramount. See for instance that most of the clues to detect and solve exceptions to a plan in complex situations are not observable without complex tacit human knowledge, which is continuously evolving as the mission unfolds. Bearing this in mind, a formalism that directly maps the plan and its visual representation is essential. In addition to meet this requirement, workflows also have the advantage of being common for the representation of activities in human organisations, thus providing a natural integration of robots in human knowledge intensive tasks.

Another benefit of considering a knowledge-based

component for human-robot teamwork is that actual human-robot teams encompass few elements, meaning that the cost of having one human taking care of major strategic decisions for the entire team is possibly less than having the system taking uninformed strategic decisions. In addition, providing humans with a visual description of the mission state is essential to improve their *overall mission awareness* (refer to (Drury et al., 2003) for a thorough study on the awareness topic in the human-robot interaction domain). The need for this improvement is suggested by the limitations of common map-centric and video-centric interfaces on fostering mission awareness (Drury et al., 2007).

The paper is organised as follows: Section 2 presents the knowledge-based concepts of the proposed approach, whereas Section 3 describes the multi-agent system for human-robot teamwork. In Section 4, a case study is described, and a set of empirical results, obtained from both simulated and real experiments, are discussed. Finally, conclusions and pointers to future work are given in Section 5.

2 KNOWLEDGE-BASED APPROACH

Under the assumption of knowledge-based human-robot teamwork, domain knowledge must be acquired, formalised, adapted, and employed for the coordination of team members performing a mission. Such knowledge is mainly composed of *mission templates* specified by a domain expert in terms of workflows. Mission templates are then adapted and instantiated by the mission coordinator (a human) to the actual team on field, in order to build an operational team plan. That is, *physical entities* (i.e. robots and humans) are assigned to *participants* in the mission template. The proposed approach is composed of four major steps, namely:

Mission Template Specification. Mission templates are knowledge intensive tasks specifications, i.e. domain knowledge, maintained in a knowledge base supported by a well specified ontology. Mission templates are non operational team plans, in the sense that: (1) on field adaptations to the template are expected; and (2) no knowledge of which physical entity (i.e. human or robot) will play the role of a given participant is known beforehand.

Mission Template Adaptation. On field, the mission coordinator selects and adapts templates according to the environment and work to be done by the team. An example of an adaptation is the

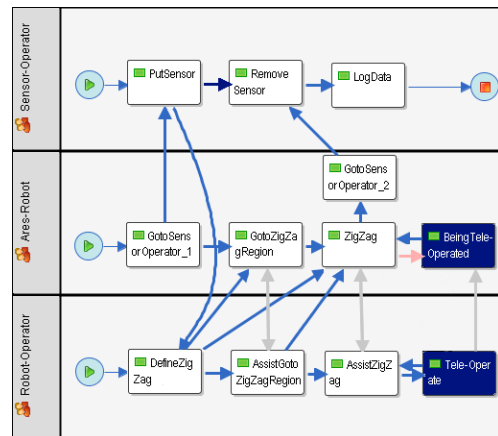


Figure 1: Partial view of the WFDM tool. Dark boxes represent active activities. Transitions, data-flow links, and exceptions are represented by blue, grey, and pink arrows, respectively. Each row corresponds to a team participant.

addition of a new activity to deal with a specific exception. In the process, some of the adapted mission templates are added to the mission templates library for reuse.

Mission Template Instantiation. In this step the mission coordinator instantiates the adapted mission template towards an operational team plan, by recruiting physical entities to the team.

Team Plan Execution. Finally, the operational team plan is distributed to each team member and executed.

Mission specification, adaptation, instantiation, and monitoring are performed in the Workflow Design and Monitor (WFDM) tool (see Fig. 1). It was developed by the authors over the Together Workflow Editor (TWE) community edition¹ tool for workflow design applied to human organisations.

2.1 Data-Flow Links

Typically, workflows describe sequences of activities, which can exchange data at transition time. This limits their application in domains where activities must exchange data in a tightly coupled way, i.e. during their execution. To cope with this limitation, the concept of *cooperative workflows* has been proposed in (Godart et al., 2000). Although in cooperative workflows, activities can share data while executing, being mostly business management oriented, the exchanged data is performed sporadically and in the form of documents.

¹TWE homepage: <http://www.together.at/>

The introduction of robots as team members adds new challenges to the execution of cooperative workflows. Robots require the use of structured data, i.e. with an explicit semantics. In addition, since many of the interactions have the purpose of allowing one participant (typically a human) to modulate the behaviour of another one (typically a robot), messages must be exchanged in a frequent basis. An example of a tightly coupled interaction is when a robotic team member is being teleoperated by a human team member.

Bearing this in mind, the cooperative workflow formalism is here extended with *data-flow links*, which allow activities to exchange structured messages, i.e. according to the ontology, in an asynchronous and frequent basis.

In addition to task-dependent interactions, subordination relationships also play a relevant role in human-robot teamwork. For this purpose, some activity parameters are defined at system level. For example, one team member (typically human) must be able to terminate another team member's (typically robotic) activity. This termination order is sent through a data-flow link. This feature can be seen in Fig. 1, in which the robot is teleoperated only while the human operator considers necessary.

3 MULTI-AGENT SYSTEM FOR TEAMWORK

This section describes the multi-agent system supporting the creation, adaptation, instantiation, and finally, execution of team plans. The system is built over the Java-based multi-agent platform JADE (Bellifemine et al., 1999), which provides two main facilities, namely: a yellow pages service for agent registration and lookup, plus an inter-agent messaging infrastructure.

Fig. 2 illustrates the major components of the multi-agent system. The coordinator represents human operators/experts responsible for formalising, adapting, instantiating, and monitoring a mission. The robotic and human participants represent robots and humans, respectively, involved in the mission's execution.

The explicit separation between human and robotic participant is essential at all levels. First, when defining domain knowledge, it is important to know which concepts must be followed by a human readable description. At execution time, humans are very good in understanding the situation at hand, even in the presence of incomplete information. With experience, humans are also very good in understanding

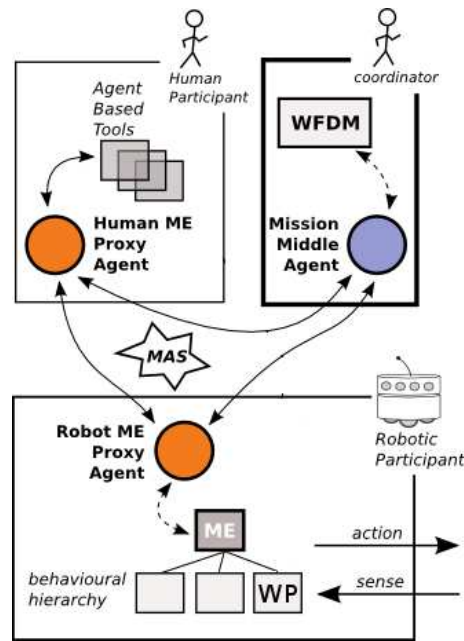


Figure 2: Multi-agent system for teamwork.

when something is not working properly, like suspecting that messages are being lost when information appears in an intermittent way. Unlike robots, humans are not fully dependent on system level mechanisms to handle these situations. Consequently, watchdog and handshaking mechanisms, among others, are important in tasks to be performed by robots.

Understanding what activities are to be performed by humans and robots allows the system, for instance, to use the network in a parsimonious way. Handshaking mechanisms, which introduce network overhead, can be relaxed when performed in messages flowing from robots to humans. Considering humans and robots in such asymmetrical way, allows the system to exploit each one's specificities on its behalf.

3.1 Coordinator

By using the WFDM tool, the domain expert formalises knowledge, the coordinator adapts, instantiates, and monitors the execution of the mission. The WFDM tool interacts with the coordinator's proxy in the system, i.e. the *mission middle agent*, in order to provide the coordinator with a list of available physical entities able to play the role of each mission's participant. The coordinator is responsible for the final selection. Afterwards, the part of the plan corresponding to each participant is sent to its *Mission Execution (ME) proxy agent*. Finally the team initiation is done by the *mission middle agent*, by informing all *ME proxy agents* of the event.

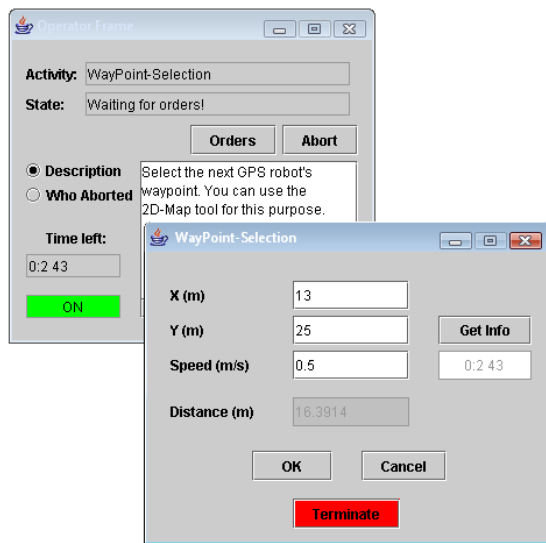


Figure 3: Human participant graphical interface. The frame in the rear illustrates the main front-end containing the activity's description. The frame in the front is dynamically adapted according to the current activity.

During mission execution, each *ME proxy agent* informs the *mission middle agent* of its execution state (e.g. which activity is currently being executed). This information is then presented through the WFDM tool to the coordinator, reflecting the status of the mission. This symbolic information augments the coordinator's *mission awareness*.

3.2 Participants

Each *ME proxy agent* is composed of two main components: (1) the *Multi-Agent System Interaction Mechanism* (MAS-IM), and the (2) *physical entity interface*. The MAS-IM is the component that enables the agent to interact with other agents in the multi-agent community, as well as to exploit its middleware services (e.g. yellow pages service). In this work, this module is built over the JADE platform. The *mission middle agent* also aggregates a MAS-IM module for the same purposes. The *physical entity interface* is the module abstracting the physical entity, i.e. its control system in the robot case (e.g. the *behavioural hierarchy* in Fig. 2), and a set of graphical interfaces (see Fig. 3) in the human case.

3.3 Plan Execution

Let us start explaining the plan execution with a motivating example. At a given moment, the *human ME proxy agent* knows that its current activity is *waypoint-selection*. In this situation, its *physi-*

cal entity interface adapts the graphical interface as in Fig. 3, so that the human can fill in the next waypoint for the robot. Each time the operator updates this field, its *operator ME proxy agent* sends an inter-agent message to the *robot ME proxy agent*, currently executing the activity *goto-waypoint*. This message exchange has been specified in the team plan by means of a data-flow link. Then, through its *physical entity interface*, the *robot ME proxy agent* updates the robot's control system according to the incoming message, consequently modulating the *goto WayPoint (WP) behaviour*. The *WP behaviour* is implemented by a set of perception-action rules able to drive the robot towards the given waypoint. In addition, the human is also provided with a message suggesting the use of a 2-D visualisation (Santos et al., 2007) so as to enhance its *situation awareness*, before selecting the waypoint. This example highlights the main role of the *human ME proxy agent*: to provide *awareness*.

In detail, the plan execution proceeds as follows. As mentioned, each *ME proxy agent* receives from the *mission middle agent* the part of the team plan corresponding to the participant it is representing. Then, it executes its part of the plan according to the following algorithm:

1. Obtain participant's start-activity.
2. While the current activity is not terminated, update its input parameters with the contents of incoming, from other *ME proxy agents*, data-flow messages. In addition, *ME proxy agents* send data-flow messages to others alike, whose contents are the current activity's output parameters values.

The aforementioned process of updating the activity's input parameters is done by sending a message to the participant's *physical entity interface*, which is able to interface directly with the entity's execution layer (e.g. robot's control system). In turn, the execution layer provides the *ME proxy agent* with the current values of the activity's output parameters, through the *physical entity interface*.

The activity's termination event, along with its code (e.g. *not-ok-aborted*, *not-ok-time-out*, or *ok*), is provided to the *ME proxy agent* through the *physical entity interface*.

3. Being the current activity, *C*, terminated, the last obtained values of its output parameters are sent to the *ME proxy agents* of those participants that have active incoming transitions from *C*. These transitions become active if their associated conditions on the termination code of *C* are met. These messages are buffered in the receiving *ME*

proxy agents allowing subsequent asynchronous consumption. For further reference, these messages will be called transition messages.

4. Wait until one of the subsequent participant's activities becomes active. This activation occurs if all necessary, or one sufficient of its transitions, is active too. This is assessed by verifying if any of the received transition messages refers to the necessary and sufficient transitions.
5. The parameters encompassed in received transition messages are used to update the activity's input parameters. If more than one message (e.g. sent by activities in different participants) feeds the same input parameter, only one is selected according to a pre-specified – in the plan – priority. The actual update of the activity's input parameters is carried out as in step 2, i.e. through the *physical entity interface*.
6. Return to step 2 until the end-activity is reached.

To allow the coordinator to follow the mission unfolding, messages stating activities and transitions activation/deactivation events are sent to the *mission middle agent*, which in turn updates the WFDM tool.

4 CASE STUDY

In order to illustrate the proposed architecture, one case study has been selected: scanning a terrain with a scent sensor to detect minefields. The case study is defined as a high-level task involving one robot, and two humans, viz. one robot operator plus one sensor operator. The goal is to determine if a given terrain is a minefield. When the mission starts, the robot is equipped with a sensor able to determine the probability of the terrain to be a minefield. After analysing the terrain, the sensor is returned to the sensor operator, which is located in a safe location away of the potential minefield. The robot operator, also remote to the operations site, helps the robot whenever needed. Fig. 1 depicts how the team plan looks like in the graphical interface of the WFDM tool.

In this case study, the robot first moves towards the operator handling the scent sensor so as to get the sensor (*GotoSensorOperator_1*). After reaching the operator, *GotoSensorOperator_1* terminates activating *PutSensor*, in which the sensor operator equips the robot with the sensor. Then, the robot operator parameterises a zig-zag behaviour (i.e. a set of parallel lanes to be followed in a sequential manner) using the graphical interface of activity *DefineZigZag*. Afterwards, the robot moves in the direction of the

defined zig-zag region (*GotoZigZagRegion*), while being modulated by the robot operator whenever necessary (*AssistGotoZigZagRegion*). The zig-zag specification is passed to the robot as a transition input parameter, whereas the *GotoZigZagRegion* modulating signal is passed through a data-flow link. As soon as the robot reaches the zig-zag region, the zig-zag behaviour is activated (*ZigZag*), which is also assisted by the robot operator (*AssistZigZag*). An example of assistance is “change to the next lane”. If the robot departs too much from the lane being followed, caused for instance by the presence of a large obstacle, then *ZigZag* terminates with an exception. In response, the robot passes to teleoperation mode (*BeingTeleOperated*) and the current robot operator's activity is terminated.

Then, the robot operator is called to teleoperate the robot (*TeleOperate*). In this case, the *TeleOperate* provides *BeingTeleOperated* with teleoperation commands as data-flow messages. This corresponds to the mission state illustrated in Fig. 1. As soon as the operator considers the robot is again in a convenient position to resume its autonomous zig-zag behaviour, *TeleOperate* terminates, which in turn requests *BeingTeleOperated* to terminate as well. This is an example of a human activity terminating a robot activity by means of data-flow. Being again in autonomous zig-zag behaviour, the robot eventually reaches the end point of the zig-zag region and *ZigZag* terminates. Then, the robot moves towards the sensor operator (*GotoSensorOperator_2*), leaving the sensor there (*RemoveSensor*), whose data is logged (*LogData*). All *Goto** activities are of the same type *GotoXY*.

4.1 Empirical Results

A set of simulated and field missions with the physical robot Ares for off-road environments (Santana et al., 2007; Santana et al., 2008), demonstrated the feasibility of the system. The information provided to the operator was enough for a proper awareness at each moment of the mission. The design of demining and surveillance missions, as complex as the one presented as case study, posed no major challenges to the user. However, for more complex tasks it was clear the need for workflow nesting capabilities. In terms of network load, the system showed to be sustainable, even for teleoperation cycles of 10Hz. However, wireless communication temporary failures resulted in the loss of messages, resulting in system's performance degradation, and sporadic crashes.

5 CONCLUSIONS AND FUTURE WORK

To our knowledge, this paper contributes with a pioneering step towards exploitation of knowledge based techniques in human-robot teamwork. The goal was to enable cooperative execution of stereotyped tasks, essential in demanding scenarios, where timely decision making is required. The cooperative workflow formalism, usually employed for business oriented human organisations, was selected.

Clear distinctions on the way humans and robots interact required the workflow formalism to be adapted. Some adaptations were suggested, with particular focus on *data-flow links*. These links enable the implementation of tightly coupled coordination. This ability is usually disregarded in works of both theoretical teamwork and cooperative workflow fields, which typically focus on high-level tasks with sporadic interactions. Although multi-robots literature is more concerned with tightly coupled coordination, it lacks a structural approach to cope with the human factor. This paper presented a multi-agent system that explicitly considers the human. First, the workflow formalism is usually employed by humans and consequently natural to them. Second, by considering different message exchanging protocols and system level activity parameters, both human and robot asymmetries are explicitly taken into account. Third, human readable information is formally attached to the ontology concepts used by the human participant. As future work we expect to make use of nested workflows. In addition, the abstraction of human-robot sub-teams as work-flow participants will also be subject of analysis. Dynamic invocation of team sub-plans will be pursued as a way of applying well known stereotyped problem solvers (i.e. mission templates) to the situation at hand. Robustness against communication channels degradation must be further studied. Handshaking and message aging policies must be analysed, separately, for the human-robot and robot-robot interaction cases. A thorough analysis of non-expert user friendliness is still missing.

ACKNOWLEDGEMENTS

We thank Paulo Santos and Carlos Cândido for proofreading. The work was partially supported by FCT/MCTES grant No. SFRH/BD/27305/2006.

REFERENCES

- Bellifemine, F., Poggi, A., and Rimassa, G. (1999). JADE – a FIPA-compliant agent framework. In *Proc. of PAAM'99*, pages 97–108, London.
- Drury, J., Keyes, B., and Yanco, H. (2007). LASSOing HRI: Analyzing Situation Awareness in Map-Centric and Video-Centric Interfaces. In *Proc. of the HRI'07*, pages 279–286, Arlington, Virginia, USA.
- Drury, J., Scholtz, J., and Yanco, H. (2003). Awareness in human-robot interactions. In *Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics*.
- Godart, C., Charoy, F., Perrin, O., and Skaf-Molli, H. (2000). Cooperative workflows to coordinate asynchronous cooperative applications in a simple way. In *Proc. of the 7th Int. Conf. on Parallel and Distributed Systems*, pages 409–416.
- Nourbakhsh, I., Sycara, K., Koes, M., Yong, M., Lewis, M., and Burion, S. (2005). Human-Robot Teaming for Search and Rescue. *Pervasive Computing, IEEE*, 4(1):72–78.
- Santana, P., Barata, J., and Correia, L. (2007). Sustainable robots for humanitarian demining. *Int. Journal of Advanced Robotics Systems (special issue on Robotics and Sensors for Humanitarian Demining)*, 4(2):207–218.
- Santana, P., Cândido, C., Santos, P., Almeida, L., Correia, L., and Barata, J. (2008). The ares robot: case study of an affordable service robot. In *Proc. of the European Robotics Symposium 2008 (EUROS'08)*, Prague.
- Santos, V., Cândido, C., Santana, P., Correia, L., and Barata, J. (2007). Developments on a system for human-robot teams. In *Proceedings of the 7th Conf. on Mobile Robots and Competitions*, Paderne, Portugal.
- Scerri, P., Pynadath, D., and Tambe, M. (2002). Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17:171–228.
- Schreiber, G., Akkermans, H., Anjewierden, A., deHoog, R., Shadbolt, N., VandeVelde, W., and Wielinga, B. (2000). *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press.
- Sierhuis, M., Clancey, W., Alena, R., Berrios, D., Shum, S., Dowding, J., Graham, J., Hoof, R., Kaskiris, C., Rupert, S., and Tyree, K. (2005). NASA's Mobile Agents Architecture: A Multi-Agent Workflow and Communication System for Planetary Exploration, i-SAIRAS 2005. *München, Germany*.
- Sycara, K. and Sukthankar, G. (2006). Literature review of teamwork models. Technical Report CMU-RI-TR-06-50, Robotics Institute, Carnegie Mellon University.
- Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence*, 7:83–124.

PROSPECTIVE ROBOTIC TACTILE SENSORS

Elastomer-Carbon Nanostructure Composites as Prospective Materials for Flexible Robotic Tactile Sensors

Maris Knite, Gatis Podins, Sanita Zike, Juris Zavickis

Institute of Technical Physics, Riga Technical University, Azenes str. 14/24, Riga, Latvia
knite@latnet.lv

Velta Tupureina

Institute of Polymer Materials, Riga Technical University, Azenes str. 14/24, Riga, Latvia
veltupur@ktf.rtu.lv

Keywords: Flexible, pressure sensor, polyisoprene, carbon black, CNT.

Abstract: Our recent achievements in the design, processing and studies of physical properties of elastomer – nano-structured carbon composites as prospective compressive strain sensor materials for robotic tactile elements are presented. Composites made of polyisoprene matrix and high-structured carbon black or multi-wall carbon nano-tube filler have been designed and manufactured to develop completely flexible conductive polymer nano-composites for tactile sensing elements. Electrical resistance of the composites as a function of mechanical strain and pressure is studied.

1 INTRODUCTION

Sensors of strain and pressure are important in many fields of science and engineering. One of the main confines of the existing conventional sensors is being discrete-point, fixed-directional and inflexible. For design of pressure sensors the most often used material is piezoelectric ceramics made separated from the material or structure being monitored. There is a demand for new flexible large-area sensors that can be embedded, for example, into the flexible skin material of robotic fingers and used for sensing multiple locations. Due to increasing application in the machine-building, especially in aerospace industry, polymer – carbon black composites are most actively studied materials (Manson, 1976; Sichel, 1982). Such composites are mostly used as the so-called inactive materials in electrical heating elements, resistors, and antistatic charge materials or shields of electromagnetic radiation (Wessling, 1986). Conductive rubber composites produced using carbon black, are still attracting attention as active materials due to the effectiveness in applications such as strain and pressure sensors, temperature sensors and selective gas sensors (Aneli, 1999; Zhang, 2000; Das, 2002;

Job, 2003; Knite, 2002; Knite, 2004; Dohta, 2000; Dharap, 2004). New interesting properties are expected in case the composite contains dispersed nano-size conducting particles. If the size of carbon particle and specific surface area of carbon black are between 60 to 200 nm and 16-24 m²/g, respectively (low-structure carbon nano-particles (LSNP)), the electrical resistance of natural rubber composites slowly decreases with applied pressure (Job, 2003). The effect is explained by the increasing number of conductive channels due to the increase of external pressure.

Resistance of polyisoprene – carbon nano-composites grows very rapidly and reversibly for both – tensile and compressive strain when high-structured carbon nano-particles (HSNP) (specific surface area 950 m²/g, mean diameter 25 nm) are used as the filler (Knite, 2002; Knite, 2004).

In this paper our recent success in the design, processing and studies of physical properties of polymer – nano-structured carbon (PNC) composites is presented with regard to prospective flexible compressive strain and pressure-sensing materials for soft robotic tactile elements.

Superior mechanical and electrical properties of carbon nano-tubes offer attractive possibilities for new sensors. So far most studies have been related

to mechanical deformation and change of nano-scale electrical properties. An attempt to use the strain sensing capability of single-wall carbon nano-tubes (SWCNT) on the nano-scale level in a macro-scale strain sensor was made by Dharap et al.. We present in this paper an attempt to use the multi-wall carbon nano-tubes (MWCNT) to devise a flexible composite for macro-scale pressure indicators (relative pressure difference sensors) or robotic tactile elements.

2 DESIGN PRINCIPLES OF THE STRUCTURE OF MATERIALS

On the basis of the results of other authors (Aneli, 1999; Zhang, 2000; Das, 2002; Job, 2003) we have developed the following principles designing the structure of materials to obtain most sensitive multifunctional elastomer-carbon nano-composites:

1) Polyisoprene (natural rubber) of the best elastic properties has to be chosen as the matrix material;

2) High-structured carbon nano-particles (HSNP) providing a fine branching structure and a large surface area (better adhesion to polymer chains compared to LSNP) should be taken as the filler. Because of a higher mobility of HSNP compared with LSNP the electro-conductive network in the elastomer matrix in this case is easily destroyed by very small tensile or compressive strain. We suppose this feature makes the elastomer-HSNP composite an option for more sensitive tactile elements in robots.

3) The highest sensitivity is expected in the percolation region of a relaxed polyisoprene composite. The smallest mechanical strain or swelling of the composite matrix remarkably and reversibly increases resistance of such a composite. From the thermodynamic point of view the sensing of strain is based on the shift of percolation threshold, for example, under tensile strain as shown in Figure 1 (Knite, 2002; Knite, 2004).

As seen from Fig. 1, the shift of the percolation threshold at less than 30% strain changes electric resistance of the composite with 10 mass parts of HSNP filler more than 10^4 times. The highest strain sensitivity is expected in the percolation region (9-10 mass parts of the HSNP filler) of a relaxed PNC composite. Thus, the maximum sensitivity of PNC composite materials to thermodynamic forces is supposed to occur near the percolation threshold of electric conductivity.

To obtain completely flexible tactile sensing elements of large area (relative to rigid piezoelectric sensors) a layer of the active PNC composite is fixed between two conductive rubber electrodes by means of a special conductive rubber glue.

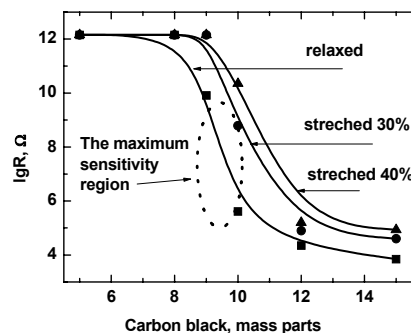


Figure 1: Shift of the percolation threshold in polyisoprene – high-structured carbon nano-particle composite under tensile strain (Knite, 2002; Knite, 2004).

3 PREPARATION OF SAMPLES AND ORGANISATION OF THE EXPERIMENT

The polyisoprene – nano-structured carbon black (PNCB) composite was made by rolling high-structured PRINTEX XE2 (DEGUSSA AG) nano-size carbon black and necessary additional ingredients – sulphur and zinc oxide – into a Thick Pale Crepe No9 Extra polyisoprene (MARDEC, Inc.) matrix and vulcanizing under 30 atm pressure at 150 °C for 15 min. The mean particle size of PRINTEX XE2 is 30 nm, DBP absorption – 380 ml/100 g, and the BET surface area – 950 m²/g.

The polyisoprene – carbon nanotube (PCNT) composites containing dispersed multi-wall carbon nanotubes (MWCNT) were prepared as follows. The size of MWCNT: OD = 60-100 nm, ID = 5-10 nm, length = 0.5-500 μm, BET surface area: 40-300 m²/g. To increase the nano-particles mobility and to obtain a better dispersion of the nano-particles within the matrix the matrix was treated with chloroform. The prepared matrix was allowed to swell for ~ 24 h. The MWCNT granules were carefully grinded with a small amount of solvent in a china pestle before adding to the polyisoprene matrix. Solution of the polyisoprene matrix and the concentrated product of nano-size carbon black was mixed with small glass beads in a blender at room temperature for 15 min. The product was poured into a little aluminum foil box and let to stand for ~ 24 h,

dried at 40 °C and vulcanized under high pressure at 160°C for 20 min.

Discs of 16 mm in diameter and 6 mm thick were cut from the vulcanized PNCB composite sheet. Conductive polyisoprene – HSCB (30 mass parts) composite electrodes were prepared and fastened to the disc with special conductive adhesive (BISON Kit + 10 mass parts of HSCB) as shown in Figure 2.

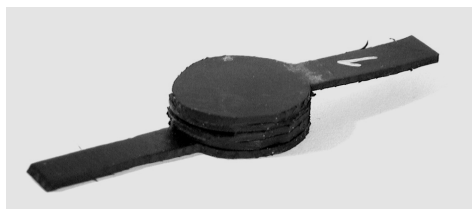


Figure 2: Picture of completely flexible strain sensing element made of PNCB composite with conductive rubber electrodes.

Aluminum electrodes were sputtered on opposite sides of the sensing element (20 × 11.5 × 2.4 mm) made of the PCNT composite as shown in Figure 3.

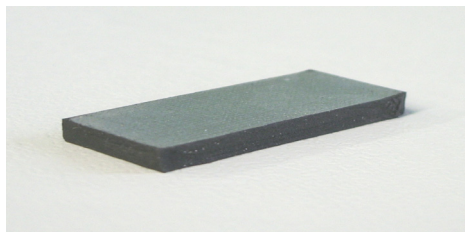


Figure 3: picture of a strain sensing element made of PCNT composite with sputtered Al electrodes.

Electrical resistance of samples was measured vs mechanical compressive strain and pressure on a modified Zwick/Roell Z2.5 universal testing machine, HQ stabilized power supply, and a KEITHLEY Model 6487 Picoammeter/Voltage Source all synchronized with an HBM Spider 8 data acquisition logger. Resistance R of the composites was examined with regard to compressive force F and the absolute mechanical deformation Δl in the direction of the force. Uniaxial pressure and relative strain were calculated respectively.

4 EXPERIMENTAL RESULTS AND DISCUSSION

The percolation thresholds of PNCB and PCNT composites were estimated in the first place. Of all the composites examined, the best results were

obtained with samples containing 14.5 mass parts of MWCNT and 10 mass parts HSCB, apparently belonging to the region slightly above the percolation threshold. Dependence of electrical resistance on uniaxial pressure first was examined on a PNCB composite disc without the flexible electrodes. Two brass sheets 0.3 mm thick and 16 mm in diameter were inserted between the disc and electrodes of the testing machine.

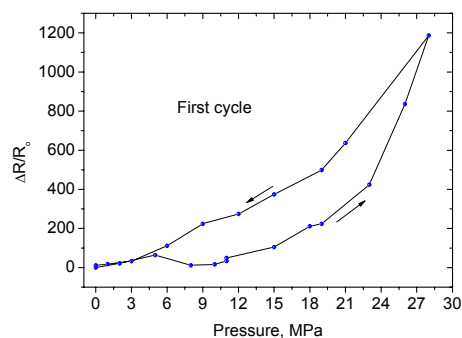


Figure 4: Electrical resistance (in relative units) of an element (without flexible electrodes) of PNCB composite containing 10 mass parts of HSCB as function of pressure. $T = 293$ K.

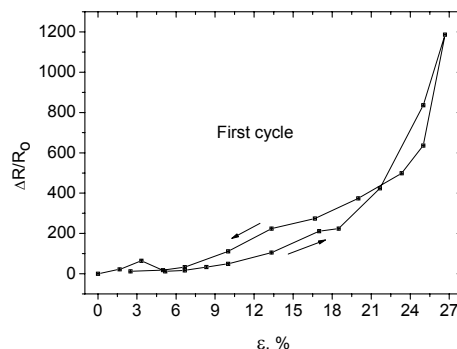


Figure 5: Electrical resistance (in relative units) of an element (without flexible electrodes) of PNCB composite containing 10 mass parts of HSCB as function of compressive strain ϵ . $T = 293$ K.

The piezoresistance effect in PNCB composite is reversible and positive ($(\Delta R)/R_0 > 0$) (Figure 4 and Figure 5).

As a next the measurements of the piezoresistance effect observed in an element of PNCB composite with flexible electrodes attached is illustrated in Figure 6 and Figure 7 showing that the piezoresistance effect decreases approximately 10 times but remains positive.

The positive effect can be explained by transverse slippage of nano-particles caused by

external pressure leading to destruction of the conductive channels.

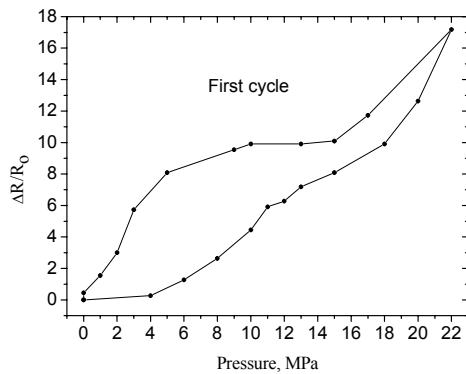


Figure 6: Electrical resistance (in relative units) of an element (with flexible electrodes) of PNCB composite containing 10 mass parts of HSCB as function of pressure. T = 293 K.

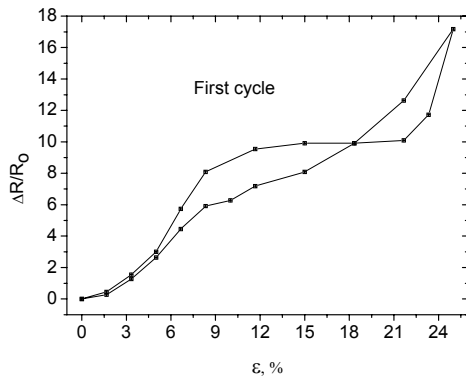


Figure 7: Electrical resistance (in relative units) of an element (with flexible electrodes) of PNCB composite containing 10 mass parts of HSCB as function of compressive strain ϵ . T = 293 K.

As seen from Figures 8, 9 and 10, the electrical resistance of the sensing element of PCNT composite decreases monotonously with uniaxial pressure and compressive strain. In this case the piezoresistance effect is considered as negative ($(\Delta R)/R_0 < 0$). Compared with a sensing element of the PNCB composite with flexible electrodes the piezoresistance effect – the absolute value of $(\Delta R)/R_0$ of a sensing element of the PCNT composite (Figure 7 and Figure 9) is more than 10 times smaller. Thus, the PNCB composite is more sensitive to mechanical action than the PCNT composite. The latter exhibits a more monotonous dependence of electrical resistance on compressive strain.

Moreover, only insignificant changes of disposition of the curve were observed during 20

cycles (Figure 10). We explain the negative piezoresistance effect by formation of new conductive channels of MWCNT under external pressure.

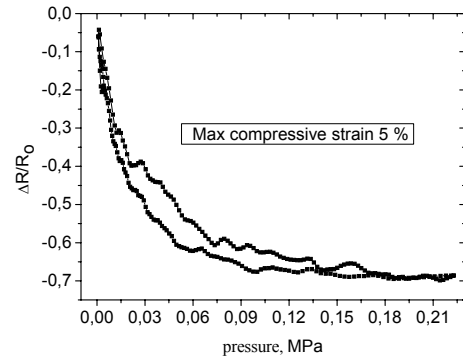


Figure 8: Electrical resistance (in relative units) of an element (with Al electrodes) of PCNT composite containing 14.5 mass parts of MWCNT as function of pressure. T = 293 K.

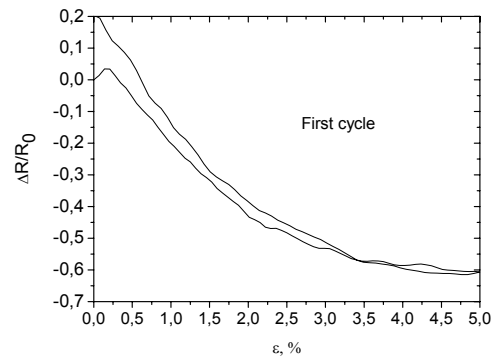


Figure 9: Electrical resistance (in relative units) of an element (with Al electrodes) of PCNT composite containing 14.5 mass parts of MWCNT as function of compressive strain ϵ . T = 293 K.

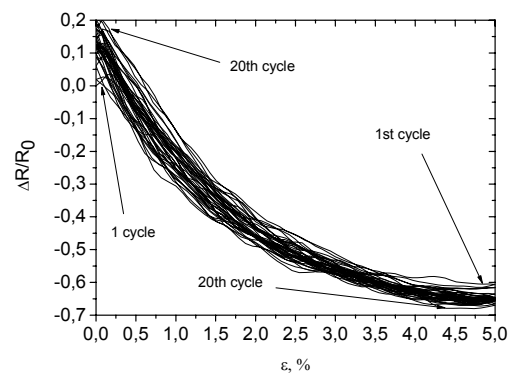


Figure 10: Electrical resistance (in relative units) of an element (with Al electrodes) of PCNT composite containing 14.5 mass parts of MWCNT as function of compressive strain ϵ . 20 loading cycles. T=293 K.

Consequently, the PNCB composite could be a prospective material for pressure-sensitive indication while the PCNT composite can be considered as a prospective material for pressure sensors.

5 CONCLUSIONS

Completely flexible sensing elements of polyisoprene – high-structured carbon black and polyisoprene – multi-wall carbon nanotube composites have been designed, prepared and examined. The first composite having a permanent drift of its mean electrical parameters is found to be a prospective material for indication of pressure change. The other composite has shown good pressure sensor properties being capable to withstand many small but completely stable and reversible piezoresistive cycles.

REFERENCES

- Manson, J.A., Sperling, L.H., 1976. *Polymer blends and their composites*. New York: Plenum Press.
- Sichel, E.K., editor, 1982. *Carbon black polymer composites*. New York: Decker.
- Wessling, B., 1986. *Elektrisch leitfähige Kunststoffe*. *Kunststoffe*, 76: 930
- Aneli, J.N., Zaikov, G.E., Khananashvili, I.M., 1999. *Appl J., Polymer Sci.*, 74: 601.
- Zhang, X.W., Pan, Y., Zheng, Q., Yi X. S., 2000. *Journal of Polymer Science B.*, 38: 2739.
- Das, N.C., Chaki, T.K., Khastgir, D., 2002. *Polymer International*, 51: 156.
- Job, A.E., Oliveira, F.A., Alves, N., Giacometti, J.A., Mattoso, L.H.C., 2003. Conductive composites of natural rubber and carbon black for pressure sensors. *Synthetic metals*, 135-136: 99-100
- Knite, M., Teteris, V., Polyakov, B., Erts, D., 2002. Electric and elastic properties of conductive polymeric nanocomposites on macro- and nanoscales. *Materials Science & Engineering C*, 19: 5-19.
- Knite, M., Teteris, V., Kiploka, A., Klemenoks, I., 2004. Reversible tenso-resistance and piezo-resistance effects in conductive polymer-carbon nanocomposites. *Advanced Engineering Materials*, 6: 742-746.
- Dohta, S., Ban, Y., Matsushita, H., 2000. Application of a flexible strain sensor to a pneumatic rubber hand. *Proc. of 6th Triennial International Symposium on Fluid Control, Measurement and Visualization*, Canada, Sherbrooke, 87.
- Dharap, P., Li, Z., Nagarjaiah, S., Barrera, E.V., 2004. Nanotube film based on single-wall carbon nanotubes for strain sensing, *Nanotechnology*, 15: 379-382.

DCT DOMAIN VIDEO WATERMARKING

Attack Estimation and Capacity Evaluation

O. Dumitru, M. Mitrea and F. Prêteux

Institut TELECOM / TELECOM & Management SudParis, ARTEMIS Departement, France
{octavian.dumitru, mihai.mitrea, francoise.preteux}@it-sudparis.eu

Keywords: DCT video watermarking, capacity, attack, *pdf* estimation, Gaussian mixtures.

Abstract: The first difficulty when trying to evaluate with accuracy the video watermarking capacity is the lack of a reliable statistical model for the malicious attacks. The present paper brings into evidence that the attack effects in the DCT domain are stationary and computes the corresponding *pdfs*. In this respect, an in-depth statistical approach is deployed by combining Gaussian mixture estimation with the probability confidence limits. Further on, these *pdfs* are involved in capacity computation. The experimental results are obtained on a corpus of 10 video sequences (about 25 minutes each), with heterogeneous content.

1 INTRODUCTION

For property right identification purposes, the watermarking techniques insert a *mark* into some original media (*e.g.* a video). If the mark insertion does not result in visual artefacts, the method features *transparency*. If a pirate cannot eliminate the mark without damaging the marked video, the method features *robustness* (Cox & others, 2002).

In practice, the better the robustness, the worse the transparency. In order to reach a balance between these two constraints, the mark is inserted into some spectral representations of the original data, *e.g.* in the *DCT* (Discrete Cosine Transform).

A crucial issue is to compute the watermarking capacity, *i.e.* the largest amount of information which can be inserted into a video, for prescribed transparency and robustness. The watermarking capacity is computed as the capacity of the noisy channel modelling the watermarking method, Figure 1. According to this model, the mark is sampled from the information source. The detection is impaired by the noise sources: the original video itself and the attacks. The side information watermarking exploits the fact that the original video is known at the insertion but unknown at the detection. As such a noise source should not decrease the channel capacity (Costa, 1983), the attacks remain the restricting factor and their intimate knowledge would grant accuracy in capacity evaluation. The present paper focuses on some real life attacks and models their effects in the

DCT domain. Note that attack modelling is not a trivial task. Actually, any mathematical approach should properly answer at least the following questions:

1. *Does a general statistical model for the considered attack effects, independent with respect to the video sequence, really exist?*
2. *When considering an individual video sequence, does a reliable model exist for any (intra)frame content and any (inter)frame dependency? Positive answers at these first questions mean a proof of stationarity concerning the attacks.*
3. *In case such a model exists, which is its pdf (probability density function)? Although the Gaussian law is generally considered, previous studies rejected this popular assumption.*

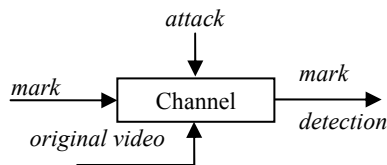


Figure 1: The watermarking model.

The paper has the following structure. After having defined a set of random variables corresponding to the attack effects, Section 2 presents the statistical investigation procedure. Section 3 describes the experimental results. The capacity evaluation is dealt with in Section 4 while Section 5 concludes the paper. The Appendix summarises some theoretical bases.

2 INVESTIGATION PROCEDURE

2.1 Attack Effect Representation

Be there an L frame colour video. Each frame is represented in the HSV space. The following steps are applied to each frame (Mitrea & others, 2006):

- Compute the DCT on the original V component.
- Decreasingly sort the coefficients and record the largest R values in a vector n_o ; record their corresponding locations in a vector l .
- Apply the DCT to the attacked V component and record the coefficients at the l locations into the new vector, denoted by n_a .
- Compute the vector: $difference = n_a - n_o$.

A set of L vectors of the same type as $difference$ (each of them with R components) is thus obtained.

Be $noise$ a vector with L components, containing the values corresponding to an arbitrarily chosen rank r in the set of $difference$ vectors: $noise = [n_1, n_2, \dots, n_L]$. Such a vector is sampled from a random variable modelling the attack effects in the r^{th} rank of the DCT hierarchy. To model the attacks means to obtain the pdf for the corresponding random variable (a model for each rank).

2.2 Pdf Estimation for Attacks

There are many pdf estimation tools based on *iid* (independent and identically distributed) data, but they do not apply here. The $noise$ vector is computed on successive (dependent) frames and the *a priori* lack of support for attack stationarity can raise suspicions about the data identical distribution. Hence, a general estimation procedure should be considered (Mitrea & others, 2007):

- *Eliminate the data dependency.* Sample the $noise = [n_1, n_2, \dots, n_L]$ vector with a D period. Shift the sampling origin and get D *iid* data sets $[x^i_1, x^i_2, \dots, x^i_N] = [n_i, n_{D+i}, \dots, n_{(N-1)D+i}]$, where $i = 1, 2, \dots, D$ and $N = L/D$.
- *Extract partial information from each iid data set.* Obtain $\hat{p}^i(x)$ ($i = 1, 2, \dots, D$) by Gaussian mixture estimation (Appendix).
- *Extract global information.* Apply the Gaussian mixture estimation to the $noise$ vector and obtain $\hat{p}_{av}(x)$ (an average model).
- *Define the model.* First, define a similarity measure between two $pdfs$, eq. (1):

$$m(u(x), v(x)) = \frac{\sum_{s=1}^S \left(\int_{I_s} (u(x) - v(x)) dx \right)^2}{\sum_{s=1}^S \left(\int_{I_s} v(x) dx \right)^2}, \quad (1)$$

where I_s , $s = 1, 2, \dots, S$ is a subdivision of the $[x_{\min}; x_{\max}]$ interval on which the u and v $pdfs$ take non-zero values. Secondly, define the attack model as $\hat{p}(\cdot)$ which is the *iid* estimate closest to the $\hat{p}_{av}(\cdot)$ in the $m(\cdot)$ sense:

$$\hat{p}(x) = \arg \min_i m(\hat{p}_{av}(x), \hat{p}^i(x)). \quad (2)$$

- *Evaluate the model accuracy.* Calculate the average similarity measure between each of the D $pdfs$ and the $\hat{p}(\cdot)$ model, eq. (3):

$$Error = \frac{1}{D} \sum_{i=1}^D m(\hat{p}^i(x), \hat{p}(x)). \quad (3)$$

If this procedure is successful when applied to an individual video (Section 3.1), then positive answers to the last two questions in the Introduction are obtained. A positive answer to the first question is obtained iff. the same model is obtained for different video sequences (Section 3.2).

3 EXPERIMENTAL RESULTS

The corpus contains 10 video sequences (64 Kbit/s), each of them of $L = 35000$ frames (about 25 minutes each). The content is heterogeneous, combining film, news, and home video excerpts.

The frame size is 192×80 pixels. The V component is normalised to the $[0, 1]$ interval. The DCT is individually applied to whole frames, and the largest $R = 360$ coefficients are investigated.

3.1 Model Computation

The model is computed for an arbitrarily chosen video sequence. The following parameters are considered: $D = 250$ frames (*i.e.* 10s); $K = 10$ $pdfs$ in the mixture; $N_{iter} = 200$ iterations in the EM algorithm; $S = 20$ evenly distributed intervals.

Table 1 presents the models for three ranks (1, 150, 300) and three attacks (Gaussian filtering, sharpening, and StirMark). In each case, the $\hat{p}(x)$ model is computed according to (2), its the parameters ($P(k)$, μ_k , σ_k) according to (A3) and

the corresponding errors to (3). Notice that each and every time, the *Error* values are lower than 0.04.

In order to illustrate the results in Table 1, Figure 2 depicts in continuous line the models for one rank ($r=300$) and the three attacks. For comparison, Figure 2 also represents (in dashed line) the Gaussian *pdf* with the same mean values and variances as the computed models.

The same results were obtained for each of the 10 video sequences in the corpus and for each investigated attack: the parameters were slightly different but the errors were lower than 0.05.

The models, for all 360 ranks and for other attacks (Frequency Model Laplacian Removal, median filtering, small rotations, JPEG compression) can be obtained by contacting the authors.

3.2 Model Validation

Up to now, the experimental results point to the existence of a model for the attack effects on a particular video sequence and estimate this model (*i.e.* elucidates the second & third questions in the Introduction).

Table 1: Statistical model for the watermarking attacks in the DCT hierarchy.

Attack	Rank	Model parameters										Error	
Gaussian filtering	$r=1$	$P(k)$	0.076	0.072	0.228	0.095	0.021	0.096	0.071	0.133	0.120	0.083	0.035
		$\mu(k)$	0.274	0.319	0.214	0.305	0.788	0.283	0.330	0.052	0.440	0.374	
		$\sigma(k)$	0.110	0.118	0.031	0.116	0.037	0.112	0.108	0.055	0.091	0.112	
	$r=150$	$P(k)$	0.019	0.025	0.190	0.213	0.091	0.075	0.271	0.025	0.031	0.056	0.013
		$\mu(k)$	0.224	0.195	0.031	0.009	0.072	0.121	0.063	0.101	0.201	0.131	
		$\sigma(k)$	0.130	0.135	0.011	0.021	0.083	0.010	0.010	0.110	0.134	0.046	
	$r=300$	$P(k)$	0.080	0.044	0.048	0.146	0.056	0.038	0.041	0.284	0.093	0.164	0.013
		$\mu(k)$	0.131	0.096	0.094	0.066	0.083	0.091	0.090	0.023	0.062	0.055	
		$\sigma(k)$	0.053	0.112	0.111	0.025	0.069	0.110	0.110	0.012	0.026	0.029	
Sharpening	$r=1$	$P(k)$	0.319	0.057	0.083	0.066	0.095	0.056	0.101	0.073	0.074	0.071	0.020
		$\mu(k)$	-1.478	-1.185	-2.325	-2.151	-2.341	-0.971	-2.424	-0.536	-2.347	-2.141	
		$\sigma(k)$	0.284	0.789	0.557	0.604	0.551	0.747	0.441	0.648	0.549	0.606	
	$r=150$	$P(k)$	0.102	0.029	0.091	0.033	0.171	0.033	0.140	0.081	0.155	0.159	0.021
		$\mu(k)$	-0.324	0.213	-0.124	-1.104	-0.080	-0.716	-0.063	-0.279	-0.078	-0.035	
		$\sigma(k)$	0.137	0.028	0.102	0.316	0.103	0.168	0.103	0.150	0.104	0.098	
	$r=300$	$P(k)$	0.054	0.151	0.283	0.105	0.078	0.067	0.073	0.098	0.023	0.063	0.021
		$\mu(k)$	-0.211	-0.105	-0.106	-0.114	-0.369	-0.030	-0.115	-0.120	-0.905	-0.203	
		$\sigma(k)$	0.180	0.096	0.095	0.156	0.133	0.133	0.156	0.158	0.155	0.178	
StirMark	$r=1$	$P(k)$	0.070	0.087	0.105	0.116	0.097	0.100	0.059	0.131	0.069	0.161	0.037
		$\mu(k)$	-0.480	0.075	-0.354	-0.289	-0.115	-0.389	-0.401	-0.348	-0.619	-0.150	
		$\sigma(k)$	0.368	0.397	0.271	0.263	0.436	0.270	0.404	0.269	0.357	0.215	
	$r=150$	$P(k)$	0.235	0.083	0.043	0.096	0.070	0.149	0.087	0.048	0.075	0.109	0.019
		$\mu(k)$	0.036	-0.084	0.206	0.103	0.422	0.006	0.017	-0.096	0.092	0.124	
		$\sigma(k)$	0.084	0.204	0.027	0.063	0.073	0.100	0.136	0.203	0.155	0.146	
	$r=300$	$P(k)$	0.085	0.075	0.046	0.078	0.137	0.036	0.010	0.046	0.300	0.093	0.018
		$\mu(k)$	0.004	0.191	0.197	-0.003	0.192	0.165	-0.005	0.212	0.047	0.015	
		$\sigma(k)$	0.144	0.193	0.192	0.141	0.074	0.195	0.140	0.189	0.036	0.148	

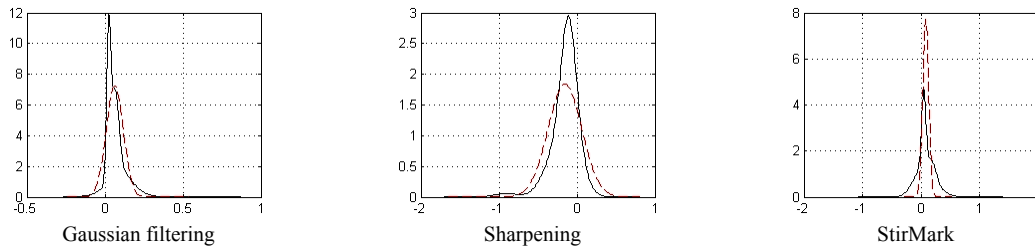


Figure 2: The attack models (continuous line) and the corresponding Gaussian laws (dashed line) for the rank $r=300$.

The model independence w.r.t. the video sequence and the estimation procedure is now to be investigated.

First, it should be précised whether the model computed on a particular video sequence can be representative for the whole corpus or not. In this respect, the investigation algorithm is resumed on the rest of 9 video sequences from the corpus and the corresponding models are computed. The errors between the reference model and these new models are evaluated according to three criteria: the similarity measure in eq. (1), the Kullback-Leibler divergence and the Hellinger distance (Appendix). For each criterion, and for three ranks, the minimal, maximal, and average errors are reported in Table 2. The numerical values obtained for the distance in eq. (1) ascertain a quite good accuracy (and generality) for the model provided in Table 1: the average errors are acceptably low, with one exception, namely the Gaussian filtering. The Kullback-Leibler divergence and the Hellinger distance lead to acceptably small values for all the attacks. In order to compute these three measures, the following instantiations were made in eqs. (1), (A4) and (A5): $v(x)$ is the reference *pdf* while $u(x)$ is, successively, each of the other 9 individual models computed on the corpus. The interval I is $I = [\mu_{mix} - 3 * \sigma_{mix}, \mu_{mix} + 3 * \sigma_{mix}]$, where μ_{mix} and σ_{mix} are the mixture mean and variance.

Secondly, the concordance between the maximum likelihood estimation (which is the basis for the EM Gaussian mixture algorithm) and the popular confidence limit estimation is checked. Note that the EM Gaussian mixture estimation results in a continuous *pdf* while the confidence limit estimation provides values for the probability that a random

variable takes values in a given interval, but not the *pdf* itself. Consequently, the interval where the Gaussian mixture *model* takes non-zero values is evenly divided into 10 sub-intervals. On the one hand, confidence limits for the probability that the noise effects would take values in these sub-intervals are derived. On the other hand, the integral of the Gaussian mixture model on the same sub-intervals are computed. The experiments bring into evidence that each and every time (*i.e.* for each type of investigated attack and for each rank) the integral on the EM Gaussian model belongs to the corresponding confidence limits.

This sub-section shows that an individual model (Table 1), computed on a particular video sequence, is valuable for all the video sequences involved in the experiments and, moreover, that it does not depend on the estimation procedure. This means a positive answer to the first question in Introduction.

4 CAPACITY COMPUTATION

As discussed in Introduction, any side-information watermarking technique can be modelled by a noisy channel, where the mark is a sample from the information source and the noise is represented by the attacks. In order to evaluate the capacity of such a channel, the eqs. (A7) and (A8) are considered. For the capacity limits in eq. (A7), the noise power

N is the variance of the *noise* vector, Section 2.2. The signal power P was derived from transparency constraints, so as to ensure a mark 30dB lower than the original (unmarked) coefficients.

Table 2: The errors (minimal, maximal, average) between the reference model and the 9 models obtained on different video sequences, for three ranks: $r = 1$, $r=150$, and $r = 300$.

Type	Attack	$r = 1$			$r = 150$			$r = 300$		
		Min	Max	Average	Min	Max	Average	Min	Max	Average
Error	Gaussian filtering	0.662	0.758	0.710	0.058	0.216	0.137	0.071	0.153	0.112
	Sharpening	0.109	0.148	0.128	0.028	0.065	0.046	0.055	0.087	0.071
	StirMark	0.077	0.093	0.085	0.065	0.109	0.087	0.084	0.131	0.108
D_{KL}	Gaussian filtering	0.131	0.204	0.167	0.011	0.016	0.014	0.029	0.030	0.029
	Sharpening	0.035	0.712	0.374	0.066	0.110	0.088	0.081	0.098	0.089
	StirMark	0.106	0.110	0.108	0.015	0.018	0.016	0.024	0.026	0.025
D_{HL}	Gaussian filtering	0.054	0.075	0.064	0.006	0.014	0.010	0.006	0.010	0.008
	Sharpening	0.213	0.255	0.234	0.009	0.019	0.014	0.015	0.015	0.015
	StirMark	0.025	0.029	0.027	0.002	0.003	0.002	0.004	0.004	0.004

The N_1 entropic power was also estimated on the n_o original coefficient vector. The bandwidth W was computed as half the frame rate.

When considering the capacity value in eq. (A8), the model provided by the present study (*i.e.* the *pdf* in Table 1) is considered as the noise *pdf* $p_N(n)$. The $p_X(x)$ function giving the capacity value is searched for by means of a numerical strategy. Actually, it is considered that $p_X(x)$ itself can be represented as a mixture of 5 Gaussian laws, thus restricting the searching to a space with 15 dimensions (5 weights, 5 means values and 5 variances). These 15 dimensions are not independent. First, the sum of weights should equal 1. Secondly, the mean of the mark (*i.e.* the mixture mean) is set to 0 (a generally accepted assumption in watermarking). Thirdly, the mixture variance was set so as to ensure a good transparency (*i.e.* 30dB lower than the host video).

The capacity values computed with the general formula and with Shannon limits are shown in Table 3. A general agreement between the two types of capacity estimation can be noticed, with some exceptions (for $r = 1$ of Gaussian filtering, StirMark). At the same time, the capacity estimation starting from the attack models is compulsory when a certain degree of precision is required: that capacity evaluation by limits can lead at relative errors of about 100% and larger!

Table 3: Capacity value and limits (lower and upper) for rank $r = 1, r = 150$ and $r=300$.

Rank	Attack	Gaussian filtering	Sharpening	StirMark
	Capacity			
$r = 1$	value	3.567	1.332	2.307
	limits	(3.632 ; 3.651)	(1.268 ; 1.725)	(2.394 ; 2.415)
$r = 150$	value	0.339	0.251	0.259
	limits	(0.037 ; 0.949)	(0.004 ; 0.569)	(0.005 ; 0.273)
$r = 300$	value	0.055	0.006	0.009
	limits	(0.018 ; 0.935)	(0.002 ; 0.621)	(0.002 ; 0.273)

5 CONCLUSIONS

The present paper brings into evidence that some real life watermarking attack effects are stationary in the DCT hierarchy and accurately estimates the corresponding probability density functions. Then, these models are involved in capacity evaluation.

From the applicative point of view, beyond watermarking itself (*i.e.* reaching the capacity limit in a practical application), these results are the starting point for a large variety of applications in the multimedia content processing, as smart indexing or in-band content enrichment, for instance.

Further work will be also devoted to considering the Blahut approach for watermarking capacity evaluation.

ACKNOWLEDGEMENTS

This work is partially supported by the HD3D-IIO project of the Cap Digital competitiveness cluster.

REFERENCES

- Archambeau, C., Lee, J., Verleysen, M., 2003. Convergence Problems of the EM Algorithm for Finite Gaussian Mixtures, *Proc. 11th European Symposium on Artificial Neural Networks*, Bruges, Belgium, pp. 99-106.
- Archambeau, C., Verleysen, M., 2003. Fully Nonparametric Probability Density Function Estimation with Finite Gaussian Mixture Models, *Proc. ICAPR*, Calcutta, India, pp. 81-84.
- Archambeau, C., Valle, M., Assenza, A., Verleysen, M., 2006. Assessment of Probability Density Estimation Methods: Parzen Window and Finite Gaussian Mixtures, *Proc. IEEE International Symposium on Circuits and Systems*, Kos, Greece.
- Basseville, M., 1996. Information: entropies, divergences et moyennes, *Internal report-INRIA*, N°1020.
- Costa, M., 1983. Writing on dirty paper, *IEEE Transactions on Information Theory*, Vol. IT-29, pp. 439-441.
- Cox, I., Miller, M., Bloom, J., 2002. *Digital Watermarking*, Morgan Kaufmann Publishers.
- Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society, Series B*, Vol. 39, No. 1, pp. 1-38.
- Dumitru, O., Duta, S., Mitrea, M., Prêteux, F., 2007. Gaussian Hypothesis for Video Watermarking Attacks: Drawbacks and Limitations, *EUROCON 2007*, Warsaw, Poland, pp. 849-855.
- Dumitru, O., Mitrea, M., Preteux, F., 2007. Accurate Watermarking Capacity Evaluation, *Proc. SPIE*, Vol. 6763, pp. 676303:1-12.
- Mitrea, M., Prêteux, F., Petrescu, M., 2006. Very Low Bitrate Video: A Statistical Analysis in the DCT Domain, *LNCS*, Vol. 3893, pp. 99-106.

- Mitrea, M., Dumitru, O., Prêteux F, Vlad, A., 2007. Zero Memory Information Sources Approximating to Video Watermarking Attacks, *LNCS*, Vol. 4705, pp. 409-423.
- Trailovic, L., Pao, L., 2002, Variance Estimation and Ranking of Gaussian Mixture Distribution in Target Tracking Applications, *Proc. of the 41st IEEE Conf. on Decision and Control*, Las Vegas – Nevada, USA, pp. 2195-2201.

APPENDIX

A1 Pdf Estimation Tools

Be there $[x_1, x_2, \dots, x_N]$ a set of N experimental data complying with the *iid* model. Suppose that these data are sampled from a random variable X whose *pdf* $p(x)$ is unknown and should be estimated. A $\hat{p}(x)$ Gaussian mixture is a linear combination of Gaussian laws and can approximate any continuous $p(x)$ *pdf* (Archambeau & other, 2003):

$$\hat{p}(x) = \sum_{k=1}^K P(k) p_k(x), \quad (A1)$$

$$\text{where } p_k(x) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right).$$

The number of mixtures K is pre-established by the experimenter and $P(k)$, μ_k , σ_k are $3K$ parameters to be estimated by the EM (expectation maximisation) algorithm (Dempster & other, 1977), based on a maximum likelihood criterion:

- the E step:

$$p^{(i)}(k/x_n) = \frac{p_k^{(i)}(x_n) P^{(i)}(k)}{\hat{p}^{(i)}(x_n)}, \quad (A2)$$

- the M step:

$$\mu_k^{(i+1)} = \frac{\sum_{n=1}^N p^{(i)}(k/x_n) x_n}{\sum_{n=1}^N p^{(i)}(k/x_n)},$$

$$(\sigma_k^2)^{(i+1)} = \frac{\sum_{n=1}^N p^{(i)}(k/x_n) (x_n - \mu_k^{(i+1)})^2}{\sum_{n=1}^N p^{(i)}(k/x_n)} \quad (A3)$$

$$P^{(i+1)}(k) = \frac{1}{N} \cdot \sum_{n=1}^N p^{(i)}(k/x_n),$$

where the (i) upper index denotes the current iteration; the total number of iterations is also subject to the experimenters choice.

The relationship among the parameters of the individual Gaussian laws and the mixture parameters is given in (Trailovic, Pao, 2002).

Alongside with the similarity measure defined in eq. (1), two popular methods for *pdf* comparison are involved in the experiments (Basseville, 1996):

- Kullback-Leibler divergence:

$$D_{KL}(u, v) = \int_I u(x) \log_2 \frac{u(x)}{v(x)} dx; \quad (A4)$$

- Hellinger distance:

$$D_{HL}(u, v) = \frac{1}{2} \int_I \left(\sqrt{u(x)} - \sqrt{v(x)} \right)^2 dx. \quad (A5)$$

A2 Capacity Evaluation Basis

The capacity of a continuous channel, whose input and output information sources are denoted by X and Y is given by the Shannon's formula (A6):

$$C = \max_{p_X(x)} \int_{x_1}^{x_2} \int_{y_1}^{y_2} f_{XY}(x, y) dx dy, \quad (A6)$$

$$f_{XY}(x, y) = p_{XY}(x, y) \log_2 \frac{p_{XY}(x, y)}{p_X(x) p_Y(y)},$$

where $p_X(x)$ and $p_Y(y)$ stand for the input and output *pdfs*, while $p_{XY}(x, y)$ is the joint *pdf* of X and Y . The x_1, x_2 and y_1, y_2 are the limits of the intervals on which the input and output *pdfs* have non-zero values. In the case of a non-Gaussian noise, Shannon derives some upper and lower limits, eq. (A7):

$$W \log_2 \frac{P + N_1}{N_1} \leq C \leq W \log_2 \frac{P + N}{N_1}, \quad (A7)$$

where W is the channel bandwidth, P is the signal power, N is the noise power, and N_1 is the noise entropy power (*i.e.* the power of a white-type noise which has the same bandwidth and entropy as the considered noise).

When assuming the noise is additive and independent, eq. (A6) becomes:

$$C = \max_{p_X(x)} \int_{x_1}^{x_2} \int_{x_1+n_1}^{x_2+n_2} g_{XY}(x, y) dy dx, \quad (A8)$$

$$g_{XY}(x, y) = p_X(x) p_N(y-x) \log_2 \frac{p_N(y-x)}{\int_{n_1}^{n_2} p_N(n) p_X(y-n) dn},$$

where the noise limits are $n_1 = y_1 - x_1$ and $n_2 = y_2 - x_2$ (Dumitru, & others, 2007).

A QUADRATIC PROGRAMMING APPROACH TO THE MINIMUM ENERGY PROBLEM OF A MOBILE ROBOT

Alain Segundo Potts*, José Jaime da Cruz* and Reinaldo Bernardi**

**Department of Telecommunications and Control, São Paulo University, São Paulo, Brazil*

***Genius Institute of Technology, São Paulo, Dept. of Telecommunications and Control, São Paulo University, Brazil
alain_2do@yahoo.com, jaime@lac.usp.br, rbernardi@genius.org.br*

Keywords: Mobile robots, quadratic programming, minimum energy, optimal control.

Abstract: As a consequence of physical constraints and of dynamical nonlinearities, optimal control problems involving mobile robots are generally difficult ones. Many algorithms have been developed to solve such problems, the more common being related to trajectory planning, minimum-time control or any specific performance index. Nevertheless optimal control problems associated to mobile robots have not been reported. Minimum energy problems subject to both equality and inequality constraints are generally intricate ones to be solved using classical methods. In this paper we present an algorithm to solve it using a Quadratic Programming approach. In order to illustrate the application of the algorithm, one practical problem was solved.

1 INTRODUCTION

1.1 Preliminaries

Mobile robotics is an important research area and for its study many researchers have dedicated a lot of time to it. There are many problems in mobile robots that are not present in industrial robots. Problems with posture maintenance, localization, equilibrium and energy consumption are common both at design and operation times. Limbed robots can be considered an important engineering conquest due to the fact that they have larger mobility, flexibility and freedom of movements than any other automatic machine (Dudek, 2000). Research in this area requires strong knowledge of mechanics, electronics, computation, and eventually biomechanics. Limbed robots are capable of walking and climbing and have been developed around the world. (Armada et al., 2003; Virk, 2005). Some of them have been used to inspect bridges (Abderrahim et al., 1999) and pipelines (Galves; Santos; Pfeiffer, 2001).

Presently the literature reports just two robots with the ability of tree climbing. The first one was developed at the Waseda University, Japan, and the second one is the RiSE robots (Robots in Scansorial Enviroments) (Saundersa et al., 2006) developed at the United States. The RiSE robot is a member of a

new class of climbing robots whose design is based on animals. For all of them, the energy consumption is a big problem. Generally the battery is their heaviest part since a considerable amount of energy is necessary to drive the legs' motors.

Typically a leg has the form of a serial mechanism with a highly nonlinear dynamics. This is one of the reasons that make to find an optimal control law a difficult problem. A second reason is the presence of both equality and inequality constraints imposed on the system – e.g., the actuator of each joint is subject to saturation (Spont et al., 1989).

Many authors have worked in the optimization of robots operation in the context of trajectory planning. (Lin et al., 1983; Garg et al., 2002; Luo et al., 2004). The problems considered in general aimed to minimize the time or some quadratic performance index. Nevertheless in the majority of them both the Coriolis and Centrifugal terms were omitted.

Optimal trajectory control systems usually can be built by solving two associated sub-problems: i) the optimal trajectory planning (OTP); ii) the trajectory tracking control (TTC). Since there are many complex constraints conditions concerning robot kinematics and dynamics, the corresponding

algorithms for solving the OTP problem have been improved in recent years (Luo et al., 2004). Even though good results have been obtained with the OTP problem for both manipulators and industrial robots, this is not the case for mobile robots.

A minimum energy formulation may be an interesting approach for mobile robots, particularly in applications where the battery weight is a critical issue. Minimum energy problems may be difficult to solve by classical methods since they involve both the nonlinear dynamics of the robot and a set of constraints.

In this paper a Quadratic Programming approach to the minimum energy problem of a mobile robot is proposed. The method is based on the discretization of the problem. Numerical tests were performed for Kamanbaré¹, a robot currently under development at the Automation and Control Laboratory (LAC), University of São Paulo.

1.2 The Robotic Platform

Kamambaré is a biomimetic robotic platform, i.e., a robotic platform inspired in nature, with the purpose of climbing trees for environmental research applications (see Fig. 1). More specifically the platform locomotion is inspired in the form lizards climb trees. The main characteristics sought in the definition of the Kamanbaré platform were: locomotion in irregular environments (unpredictability of the branch complexity that compose a tree), surmounting obstacles (nodes and small twigs), tree climbing and descending without risking stability, and keeping low structural weight (mechanics + electronics + batteries). (Bernardi et al., 2006)

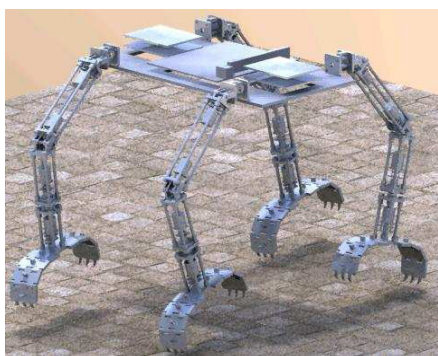


Figure 1: Robot Kamanbaré.

¹ *Kamanbaré* is the word in the Tupi indian language for chameleon.

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). (Bernardi et al., 2006)

Each leg has three rigid links connected by two rotational joints with one degree-of-freedom (d.o.f.) each. The first link is connected to the platform by a two d.o.f. rotational joint (Fig. 2).



Figure 2: Limb of the Kamanbaré Platform.

All joints are controlled by DC motors. The control problem for mobile robot is the problem of determining the time history of joint required to cause the end-effectors (the gripper) executed a commanded motion.

There are many control techniques and methodologies that can be applied to the control of limbed robots. The particular control method chosen as well as the manner in which it is implemented can have a significant impact on the performance of the robot and consequently on the range of its possible applications. Optimal control of energy can be an interesting policy when the robot must operate autonomously for a long time. The control problem for the tree climbing robot considered here is to determine the time history of each limb joint required to cause the leg to move from an initial angle to a final one in such a way that the energy loss in the motors is minimal.

2 FORMULATION AND SOLUTION OF THE OPTIMAL CONTROL PROBLEM

2.1 The State Space Model

This work consider the control strategy named independent joint control. In this type of control each axis of the limb is controlled as a single input /single output system. Any coupling effects due to the motion of the others links is either ignored or treated as a disturbance. (Spont et al., 1989)

The space state model that describes the dynamics of the DC motor located at the i -th joint ($1 \leq i \leq n$) of the leg can be expressed by the equations:

$$\dot{x} = Ax + Bu \tag{1}$$

$$y = Cx + Du + E\tau \tag{2}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -b/J \end{bmatrix}, \tag{3}$$

$$B = \begin{bmatrix} 0 \\ K_m/J \end{bmatrix}, \tag{4}$$

$$C = [0 \quad -K_b/R_a], \tag{5}$$

$$D = 1/R_a, \tag{6}$$

$$E = r/(R_a K_m), \tag{7}$$

and

$$x = \begin{bmatrix} \theta_m \\ \dot{\theta}_m \end{bmatrix}, \tag{8}$$

$$u = V - \frac{r}{K_m} \tau. \tag{9}$$

J is the moment of inertia of the rotor, b is the viscous damping coefficient of the mechanical system, V is the armature voltage (control variable), R_a is the armature resistance, $Y = i_a$ is the armature current, K_m is the torque coefficient of the motor, K_b is the counter-electromotive coefficient, τ is the load torque, $r < 1$ is the gear reduction factor, θ_m is the angular position of the rotor and $\dot{\theta}_m$ is the angular speed of the rotor. Notice that the input variable u depends on both the control

variable V and the load torque τ , which also depends on V through θ_m and $\dot{\theta}_m$.

The load torque τ , which depends on the robot dynamics, is given by : (Spong, 1989)

$$\tau = M(q)\ddot{q} + B(q)[\dot{q}_i, \dot{q}_z] + C(q)[\dot{q}^2] + G(q) \tag{10}$$

where $M(q)$ is the leg inertia matrix, $B(q)$ is the Coriolis torque matrix, $C(q)$ is the centrifugal torque matrix, $G(q)$ is the gravitational torque vector,

$$[\dot{q}_i, \dot{q}_z] = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_{n-1}, \dot{q}_n]^T,$$

$$\forall i \neq z : 1 < i < n, 1 < z < n$$

$$[\dot{q}^2] = [\dot{q}_1^2 \quad \dot{q}_2^2 \quad \dots \quad \dot{q}_n^2]^T$$

and

$$q = [q_1 \quad q_2 \quad \dots \quad q_n]^T,$$

where q_i , $1 \leq i \leq n$, is the generalized coordinate of joint i and n is the number of joints of the leg.

Some dynamical effects like friction were not included in (10) although they may be significant for some limbs. In addition, a more detailed model of the leg dynamics could include various sources of flexibility, deflection of the links under load and vibrations (Borrow et al., 2004). Nevertheless, this model is sufficiently accurate for our purposes since these effects are not significant for the leg under consideration.

One of the characteristics of the independent joint model is that τ is multiplied by the gear reduction r . Thus effect of the gear ratio is to reducing the coupling nonlinearities presents in dynamics of the limbs.

The solution to equation (1) is given by:

$$x(t) = e^{At} x(0) + \int_0^t e^{A(t-\zeta)} Bu(\zeta) d\zeta \tag{11}$$

2.2 The Optimal Control Problem

In this section it is assumed that $\tau(t)$ is known for all t in the interval $[0 \ t_f]$. The performance index adopted is the Joule loss in the armature resistance of each motor during the motion:

$$\min \varepsilon = \int_0^{t_f} P dt \quad (12)$$

where t_f is the time required to move the joint from the initial to the final position, and P is the power dissipated:

$$P = R_a i_a^2 = R_a Y^2 = R_a [x' C' + u' D' + \tau' E'] [Cx + Du + E\tau] \quad (13)$$

where Y was defined in (2).

The optimal control problem is subject to the following constraints:

$$-\frac{K_b}{R_a} x_2(t) + \frac{1}{R_a} u(t) + \frac{r}{R_a K_m} \tau(t) \leq I_m, \quad (14)$$

$$-I_m \leq -\frac{K_b}{R_a} x_2(t) + \frac{1}{R_a} u(t) + \frac{r}{R_a K_m} \tau(t), \quad (15)$$

$$-V_m \leq u(t) \leq V_m, \quad (16)$$

$$x_1(t_f) = q_f / r, \quad (17)$$

$$x_2(t_f) = 0, \quad (18)$$

where I_m and V_m are, respectively, the maximum armature current and maximum armature voltage of the motor and $t \in [0 \ t_f]$. With no loss of generality we take $x_1(0) = 0$. Considering the motor initially at rest, $x_2(0) = 0$.

This type of problem is hard to solve and generally involve a great computational effort (Kirk, 1998). Mobile robots require a quick solution and to solve it in real-time is practically impossible. These are among the reasons for which we decided to look for another kind of solution.

2.3 Discretization

Let us to define the time-step as:

$$\Delta t = \frac{t_f}{N} = t_k - t_{k-1} \quad (1 < k \leq N) \quad (19)$$

where N can be chosen sufficiently large to discretize t_f . Then it is assumed that $u(t)$ is a stepwise constant function and u_k is used to denote

the value of $u(t)$ for all t in the k -th time interval $[t_{k-1}, t_k)$.

Taking into account equations (11), (12), (13) and (19), the functional ε can be rewritten as:

$$\min \varepsilon = \sum_{k=1}^N P_k \Delta t = \sum_{k=1}^N R_a i_a^2 \Delta t \quad (20)$$

and the solution of system (1) as:

$$x(t_k) = e^{A t_k} \Gamma I_k U \quad (21)$$

where,

$$\Gamma = \int_0^{t_1} e^{-A\zeta} B d\zeta + \int_0^{t_2} e^{-A\zeta} B d\zeta + \dots + \int_0^{t_{N-1}} e^{-A\zeta} B d\zeta, \quad (22)$$

$$U = [u_1 \ u_2 \ \dots \ u_N]^T \quad (23)$$

and

$$I_k = \begin{bmatrix} e_1 & 0 & 0 & 0 \\ 0 & e_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e_n \end{bmatrix} \quad (24)$$

Where $e_i = 1$ for $1 \leq i \leq k$ and $e_i = 0$ for $k < i \leq N$.

Matrix Γ can be calculated offline since it depends only on the motor parameters.

From equations (20) and (21) it is possible rewrite (20) as:

$$\min \varepsilon(U) = \frac{1}{2} U^T Q U + \tau K^T U + \sigma \quad (25)$$

where matrices Q and K depend only on the motor parameters and σ is a constant. The constraints for the problem are:

$$-I_m \leq -\frac{K_b}{R_a} x_{2_k} + \frac{1}{R_a} u_k + \frac{r}{R_a K_m} \tau_k \leq I_m \quad (26)$$

$$-V_m \leq u_k \leq V_m \quad (27)$$

$$x_{1_N} = q_f / r \quad (28)$$

$$x_{2_N} = 0 \quad (29)$$

where constraints (26) and (27) apply for all k , $1 \leq k \leq N$.

Since σ is a constant, it is not relevant for the optimization.

The problem given by equations (25)-(29) has thus the form of a Quadratic Programming problem which is certainly the kind of nonlinear programming problem closest to linear programming from analytical and computational point of view. Solution for this kind of problem can be efficiently found by numerical methods. (Avriel, 1976) (Winston, 1995).

2.4 The Algorithm

An iterative algorithm to solve the minimum energy problem based in equations (13) and (20) is proposed in this section.

To start the algorithm it is assumed that $\tau_k =: \tau_k^0 \equiv 0$ for all k , $1 \leq k \leq N$. Then Quadratic Programming problem of minimizing the function (20) subject to constraints (26) to (29) is solved. Denote by U_k^{0*} the optimal solution for this problem. Using equation (17) and recalling equation (1), both the motor angular position $\theta_m^0(k)$ and angular speed $\dot{\theta}_m^0(k)$ can be evaluated for all k , $1 \leq k \leq N$.

The second step of the algorithm begins by using the leg dynamical equations (10) to evaluate a new torque time history τ_k^1 for all k , $1 \leq k \leq N$. The new Quadratic Programming problem is then solved and U_k^{1*} is obtained. $\theta_m^1(k)$ and $\dot{\theta}_m^1(k)$ are evaluated for all k , $1 \leq k \leq N$.

The process is repeated until $\|\tau_k^{j+1} - \tau_k^j\| < \varepsilon_\tau$ for a given accuracy ε_τ and two consecutive steps j and $j+1$. When convergence is attained the optimal vector of armature voltages V^{j+1*} can be evaluated using equation (9).

The algorithm is expected to converge since the gear reduction ratio r is generally small and the effect of the torque on the motor dynamics is correspondingly small too.

The algorithm may thus be summarized as:

Algorithm

```

 $\tau_k^0 = 0$ 
 $j = -1$ 
repeat
 $j = j + 1$ 

$$\begin{cases} \min \varepsilon(U) = \frac{1}{2} U^T Q U + \tau_k^j K^T U \\ \text{subject to constraints: equations (26) to (29)} \end{cases}$$


$$\dot{x}_k^j = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{J} \end{bmatrix} x_k^j + \begin{bmatrix} 0 \\ \frac{K_m}{J} \end{bmatrix} U_k^{j*}$$


$$[q \quad \dot{q} \quad \ddot{q}]_k^j = r [x \quad \dot{x} \quad \ddot{x}]_k^j$$


$$\tau_k^{j+1} = M(q)\ddot{q} + B(q)[\dot{q}\dot{q}] + C(q)[\dot{q}^2] + G(q)$$

until  $(\max(\tau_k^{j+1} - \tau_k^j) \leq \varepsilon_\tau)$ 

```

3 SIMULATION RESULTS

This section presents the results obtained from the application of the algorithm above to a leg similar to that of the Kamanbaré platform.

The algorithm code was written in MatLab. The following data were used: $t_f = 2s$, $N = 10$, $\varepsilon_\tau = 10^{-11}$ and

$$\begin{bmatrix} \theta_{m_{0N}} & \theta_{m_{1N}} & \theta_{m_{2N}} \end{bmatrix}^T = \begin{bmatrix} \pi & 0 & -\frac{\pi}{4} \end{bmatrix}^T$$

Algorithm convergence occurred in 15 iterations. Table 1 shows the algorithm steps until the optimal solution is reached. The overall processing time was quite small.

Figure 3 show the optimal solution U_k^{15*} for each motor.

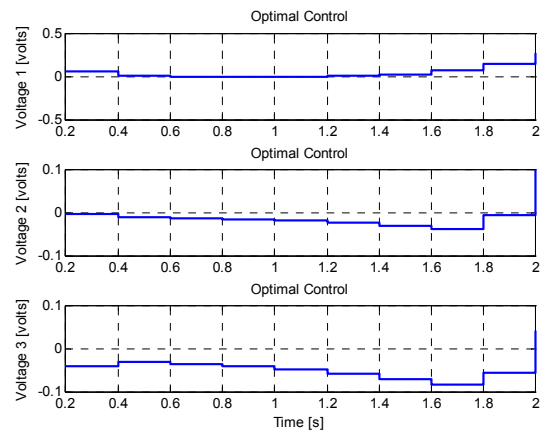


Figure 3: Optimal Solution.

Table 1: Minimum Energy Consumption.

Iteration (j)	Energy Consumption $\varepsilon = \sum_{k=1}^N P_k \Delta t$ [watts]		
	Joint 1 (Motor 1)	Joint 2 (Motor 2)	Joint 3 (Motor 3)
1	0.22583525809518	0	0.01411470363095
2	0.54691612008194	0.00249201214070	0.02563789026998
3	0.60424685648279	0.00151916407557	0.02084265484899
4	0.60913169232854	0.00162074115964	0.02201351589222
5	0.60878562782495	0.00159940182719	0.02201765824694
6	0.60878328339302	0.00160200730314	0.02202687314770
7	0.60877614492969	0.00160176202487	0.02202629388348
8	0.60877639908587	0.00160179288124	0.02202637161877
9	0.60877635993567	0.00160178879636	0.02202635576755
10	0.60877636906069	0.00160178928402	0.02202635725193
11	0.60877636847905	0.00160178921712	0.02202635705179
12	0.60877636859543	0.00160178922523	0.02202635707836
13	0.60877636858174	0.00160178922422	0.02202635707548
14	0.60877636858322	0.00160178922435	0.02202635707588
15	0.60877636858300	0.00160178922433	0.02202635707583

Figures 4 and 5 show kinematics of each joint; the angular position and the speed and acceleration achieved for the optimal control $U_k^{15^*}$.

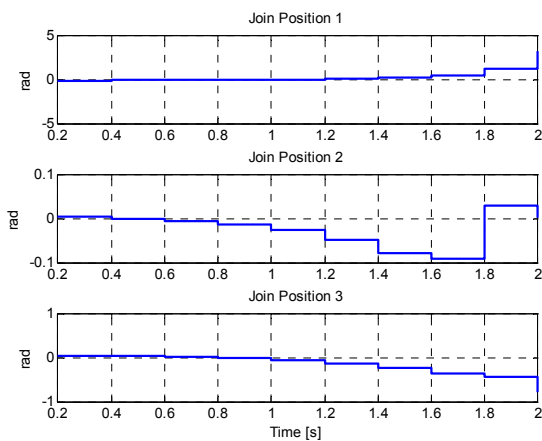


Figure 4: Joints Positions.

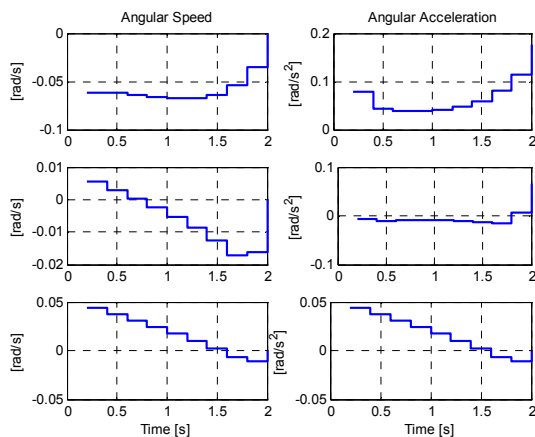


Figure 5: Speed's Joints and Acceleration's Joints.

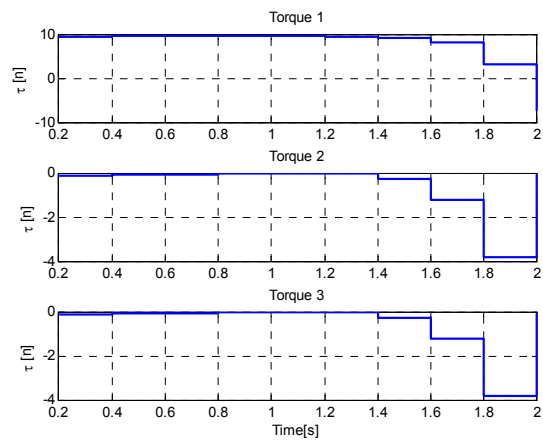


Figure 6: Torques for each motor.

In this particularly case the motor's 2 and 3 have the same torque.

4 CONCLUSIONS

This paper discussed the formulation and solution of an important problem related to mobile robotics: the minimum energy loss problem.

An optimal control problem was formulated to represent this case. After discretization in time the optimal control problem was rewritten in the form of a Quadratic Programming problem whose solution could be obtained efficiently. The algorithm was tested for a leg similar to that of the Kamanbaré platform.

Although the whole problem is nonlinear and quite complex the algorithm converged quickly in all the

tests performed by now. It is thus expected that the it can be used to operate in real-time.

REFERENCES

- Abderrahim, M., Balaguer, C., Gimenez, A., Pastor, J. M., Padrón, V. M., 1999. *ROMA: A Climbing Robot for Inspection Operations*. In, Proceedings of the IEEE International Conference on Robotics and Automation, Vol 3, pp 10-15.
- Armada, M., Santos, P. G. de, Jiménez, M. A., Prieto, M. 2003. *Application of CLAWAR Machines*. In, The International Journal of Robotics Research. Vol. 22, n°3-4, pp. 251-264.
- Avriel, M, 1976. *Nonlinear Programming. Analysis and Methods*. Prentice-Hall. New Jersey, 1st edition.
- Bernardi, R., Cruz, J.J. 2006. *KAMANBARÉ. A tree-climbing biomimetic robotic platform for environmental research*. In, International Conference on Informatics in Control, Automation and Robotics.
- Bobrow, J.E., Park, F.C., Sideris, A. 2004. *Recent Advances on the Algorithmic Optimization of Robot Motion*. Departament of Mechanical and Aerospace Engineering, University of California.
- Dudek, G., Jenkin, M., 2000. *Computational Principles of mobile Robotics*. Cambridge Press. 1st edition.
- Galvez, J. A., Santos, P. G. de, Pfeiffer, F., 2001. *Intrinsic Tactile Sensing for the Optimization of Force Distribution in a Pipe Crawling Robot*. In, IEEE/ASME Transactions on Mechatronics, Vol. 6, n°1.
- Garg, D. P., Kumar, M., 2002. *Optimization Thecniques applied to multiple manipulators for path planning and torque minimization*. In Eng. Applications Art, Int., Vol 15, pp 241-252.
- Kirk, D., 1998. *Optimal Control Theory an Introduction*. Dover Publication, New York, 2nd edition.
- Lin, Ch., Chang P., Luh, J.Y.S. 1983 *Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots*. In IEEE Transactions on Automatic Control, Vol AC-28, No 12, December.
- Luk, B., Cooke, D., Galt, S., Collie, A., Chen, S. 2005. *Intelligent legged climbing service robot for remote Maintenance applications in hazardous environments*. In Robotics and Autonomous Systems. Accepted June 2005.
- Luo, X., Fan, X., Zhang, H., Chen T., 2004. *Integrated Optimization of Trajectories Planning for Robots Manipulators Based on Intensified Evolutionary Programming*. Proceedings of the 2004 IEEE, International Conference on Robotics and Biomimetics.
- Saundersa, A., Goldmanb, D., Fullb,R., Buehlera, M. 2006. *The RiSE Climbing Robot: Body and Leg Design*. In, Proceeding of SPIE, Vol. 6230, 623017.
- Spong, M., Vidyasagar, M., 1989. *Robot Dynamics and Control*. John Wiley & Sons, Canada, 1st edition.
- Virk, G. S. 2005. *The CLAWAR Project: Developments in The Oldest Robotics Thematic Network*. IEEE Robotics & Automation Magazine, pp. 14-20.
- Winston, W. 1995. *Introduction to Mathematical Programming. Applications and Algorithms*. Duxbury Press. California, 2nd edition.

PARAMETERIZATION AND INITIALIZATION OF BEARING-ONLY INFORMATION

A Discussion

R. Aragues and C. Sagues

DIIS - I3A, University of Zaragoza, María de Luna, 50018 Zaragoza, Spain

raragues@unizar.es, csagues@unizar.es

Keywords: Bearing-only, Feature parameterization: cartesian and inverse-depth, Delayed/Undelayed initialization.

Abstract: In this paper we discuss feature parameterization and initialization for bearing-only data obtained from vision sensors. The interest of this work refers to the comparison of the bearing-only data representation and initialization techniques. The behavior of the algorithm is analyzed for different robot motions and depth of the features. The results are evaluated in terms of the sensitivity to step size and performance to ill conditioned situations. The problem studied refers to robots moving on the plane, sensing the environment and extracting bearing-only information from uncalibrated cameras to recover the position of the landmarks and its own localization.

1 INTRODUCTION

The manipulation of bearing information is an important issue in robotics. Bearing-only data is the kind of information provided by cameras through the projection of landmarks which are in the scene. In order to recover the position of these landmarks in the world, multiple observations taken from different positions must be combined.

Compared with information extracted from other sensors such as lasers, bearing information is complicated to use. However, the multiple benefits of using cameras have motivated the interest in the researchers. These benefits include the property that cameras are able to sense quite distant features so that the sensing is not restricted to a limited range.

This sensing of the environment in the form of bearing information may be used for many applications such as the computation of the landmark localization in the environment or the calculation of the own robot pose mostly known as SLAM *Simultaneous Localization and Mapping*.

Algorithms which use bearing information must deal with the problem of creating representations for features by the combination of bearing data. The problem of feature parameterization and feature initialization are of big importance here.

With regard to the feature parameterization, the classical approach has been the use of a *cartesian* parameterization (Bailey, 2003), (Kwok and Dis-

sanayake, 2004), (Costa et al., 2004), (Klippenstein et al., 2007). Some approaches prefer a *depth* parameterization, where features are stored as a starting point of the ray where the feature lays, the inclination of the ray and the depth (Davison, 2003). An *inverse-depth* parameterization is an alternative, similar to the *depth* parameterization but using the inverse of the depth instead (Montiel et al., 2006). Some approaches use no explicit feature parameterization and instead represent landmarks as constraints between three robot poses (Trawny and Roumeliotis, 2006).

With regard to the feature initialization, *Undelayed* techniques immediately introduce features in the map so that they can be used to improve the robot estimation (Montiel et al., 2006), (Trawny and Roumeliotis, 2006), (Costa et al., 2004), (Kwok and Dissanayake, 2004) while *Delayed* techniques defer the introduction into the map until the features are near-Gaussian (Bailey, 2003), (Klippenstein et al., 2007). *Delayed* techniques often create temporal representations for landmarks which are maintained in separate filters and evolve with the incorporation of new observations of these landmarks until they are finally introduced into the map (Davison, 2003).

The problem of depth computation for landmarks is afforded in two separate ways. Some approaches create depth representation from only one bearing assuming an approximate value for it. These techniques are able to cover depths from the position where the landmark was observed until infinity or until a max-

imum depth within the workspace (Kwok and Disanayake, 2004), (Davison, 2003), (Montiel et al., 2006). The other approach to depth computation is the combination of observations taken from different robot poses, where triangulation techniques are used to recover the depth (Bailey, 2003), (Klippenstein et al., 2007).

The interest of this work refers to the comparison of the bearing-only data representations and initialization techniques, analyzed for different robot motions relative to depth of the landmarks in the scene. Two feature parameterizations are studied. The first is an standard *cartesian* parameterization, where features are described by their (x, y) position. The alternative representation is an adaptation of the *inverse-depth* (Montiel et al., 2006) to the 2D situation. Besides, both *Undelayed* and *Delayed* strategies for feature initialization are used and their performance is compared in different scenarios.

The problem studied in this paper refers to robots moving on the plane, sensing the environment and extracting bearing-only information from uncalibrated images to recover the position of the landmarks and its own localization. As a result of this investigation, some theoretical solutions are proposed, and their validity is supported by an exhaustive experimentation using simulated data. Some preliminary experiments have been carried out using real data from omnidirectional images.

2 BACKGROUND

The problem studied in this paper is related to the use of bearing-only information for the SLAM problem using EKF. The robot moves on the plane and elements in the map are represented by their 2D coordinates. Robot observes landmarks within a field of view of 360° due to the use of omnidirectional cameras and obtains bearing-only measurements. Odometry is used to predict robot motion in every step. The EKF *Extended Kalman Filter* is a widely used technique in these problems and a lot of information can be found in the literature. The data association problem is not discussed in this paper. An innovation test is used to select the observations which will be used in the filter update. This test computes an individual compatibility for all observation-prediction pairs and then obtains the greatest set of jointly compatible pairs using the JCBB algorithm (Neira and Tardós, 2001). Although traditionally this algorithm is used to solve the data association problem, we use it in order to avoid the filter divergence in the presence of poorly initialized features or high innovations.

Along this paper, next notation will be used:

$\mathbf{x} = (\mathbf{x}_r, \mathbf{x}_1 \dots \mathbf{x}_n)$: the state vector containing current robot pose (\mathbf{x}_r) and the positions of landmarks ($\mathbf{x}_1 \dots \mathbf{x}_n$)

\mathbf{P} : the covariance matrix.

$\mathbf{x}_{r_j} = (x_{r_j}, y_{r_j}, \theta_{r_j}) \in \mathbb{R}^3$, $\theta_{r_j} \in [-\pi, \pi]$, for $j = 1..k$: j -th robot pose. When there is no confusion, the subscript j is omitted.

$\mathbf{x}_i = (x_i, y_i) \in \mathbb{R}^2$, for $i = 1..n$: Position of the i -th feature in the map, for *cartesian* parameterization, or $\mathbf{x}_i = (x_i, y_i, \theta_i, \rho_i) \in \mathbb{R}^4$, $\theta_i \in [-\pi, \pi]$, for $i = 1..n$: when referring to *inverse-depth* parameterization.

z_{ji} : measurement taken from robot pose j to feature i . When only one robot pose is used, z_i refers to the observation of feature i .

3 FEATURE PARAMETERIZATION

Cartesian parameterizations represent features by their (x, y) coordinates. This parameterization is very intuitive since the feature position within the map can be easily obtained. The initialization of features in this *cartesian* parameterization is problematic due to the nonlinearity of the triangulation techniques used to recover its position based on the observations taken from different robots poses. It can be easily shown that bearings generate bigger uncertainty as landmark position goes away from the camera. The observation model for a feature $\mathbf{x}_i = (x_i, y_i)$ observed from a robot pose $\mathbf{x}_r = (x_r, y_r, \theta_r)$ is (Bailey, 2003):

$$z_i = h(\mathbf{x}_r, \mathbf{x}_i) = \arctan\left(\frac{y_i - y_r}{x_i - x_r}\right) - \theta_r \quad (1)$$

Inverse-depth parameterizations represent a feature \mathbf{x}_i as a ray starting at (x_i, y_i) , the position where the feature was firstly observed, with a global bearing θ_i and a depth of $\frac{1}{\rho_i}$. Every feature is stored in the state vector using these four parameters $(x_i, y_i, \theta_i, \rho_i)$. The *cartesian* coordinates of the landmark could be calculated as:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} + \frac{1}{\rho_i} \mathbf{m}_i \quad (2)$$

where $\mathbf{m}_i = [\cos(\theta_i) \sin(\theta_i)]^T$.

The observation model with *inverse-depth* for a feature $\mathbf{x}_i = (x_i, y_i, \theta_i, \rho_i)$ observed from a robot pose $\mathbf{x}_r = (x_r, y_r, \theta_r)$ is:

$$\mathbf{h} = \mathbf{atan2}(h_y^{xy}, h_x^{xy}) \quad (3)$$

where (h_y^{xy}, h_x^{xy}) are the coordinates of the feature in the robot reference:

$$\mathbf{h}^{xy} = \begin{pmatrix} h_x^{xy} \\ h_y^{xy} \end{pmatrix} = \mathbf{R}_r \left(\begin{pmatrix} x_i \\ y_i \end{pmatrix} + \frac{1}{\rho_i} \mathbf{m}_i - \begin{pmatrix} x_r \\ y_r \end{pmatrix} \right) \quad (4)$$

$$\text{with } \mathbf{R}_r = \begin{bmatrix} \cos \theta_r & \sin \theta_r \\ -\sin \theta_r & \cos \theta_r \end{bmatrix}.$$

This observation model remains valid if next equation is used instead of equation 4 provided that $\rho_i > 0$:

$$\mathbf{h}^{xy} = \begin{pmatrix} h_x^{xy} \\ h_y^{xy} \end{pmatrix} = \mathbf{R}_r \left(\rho_i \left(\begin{pmatrix} x_i \\ y_i \end{pmatrix} - \begin{pmatrix} x_r \\ y_r \end{pmatrix} \right) + \mathbf{m}_i \right) \quad (5)$$

As advantage with respect to the *cartesian* parameterization, the observation model for the *inverse depth* is near linear. Additionally, landmarks at infinity ($\rho_i = 0$) or uncertainties that extend to infinity can be represented. The main drawback of the *inverse-depth* is that features are over-parameterized, and therefore the Covariance matrix size is greater.

4 FEATURE INITIALIZATION

The feature initialization in SLAM consists in the creation of a representation of the landmark's position and its introduction into the stochastic map through its *mean* and its *covariance matrix*. The feature initialization problem of bearing-only is due to the fact that features are only partially observable.

As told, a measurement only gives information about the direction towards the landmark and two or more observations must be combined in order to recover the depth of the landmark. However, there are some situations where the depth cannot be recovered. Next we give a formal description of these situations. *Theorem 1.-* Let us name \mathbf{x}_{r_1} a robot position and \mathbf{x}_{r_2} a second position translated but not rotated with respect to \mathbf{x}_{r_1} . Let us name z_{1i} the observation of a feature \mathbf{x}_i taken from \mathbf{x}_{r_1} and z_{2i} the observation of the same feature taken from \mathbf{x}_{r_2} . Let us name d_p the translation from \mathbf{x}_{r_1} to \mathbf{x}_{r_2} on a perpendicular direction to z_{1i} and d_l the translation on a parallel direction to z_{1i} . Without loss of generality, let d_l be equal to zero. The landmark depth (distance between \mathbf{x}_{r_1} and the landmark) can be totally determined from $\alpha = z_{1i} - z_{2i}$ as

$$\text{depth} = d_p / \tan \alpha \quad (6)$$

Corollary 1.1.- This is an undetermined problem (0/0) when simultaneously $d_p = 0$ and $\alpha = 0 + k\pi$ for $k \in \mathbb{Z}$.

Corollary 1.2.- This problem remains undetermined independently of the magnitude of d_l .

Corollary 1.3.- The landmark is at infinity if simultaneously $\alpha = 0 + k\pi$ for $k \in \mathbb{Z}$ and d_p is different of zero.

Theorem 2.- Let us name \mathbf{x}_{r_1} a robot position and \mathbf{x}_{r_2} a second position rotated but not translated with respect to \mathbf{x}_{r_1} . Let us name z_{1i} the observation of a feature \mathbf{x}_i taken from \mathbf{x}_{r_1} and z_{2i} the observation of the same feature taken from \mathbf{x}_{r_2} . Robot rotation (θ_{r_2}) can be absolutely determined from $\theta_{r_2} = z_{1i} - z_{2i}$.

Corollary 2.1.- Given a pure rotation motion, feature depth cannot be recovered.

Corollary 2.2.- Given a translation and rotation motion with landmarks of infinite depth, the robot rotation can be computed from $z_{1i} - z_{2i}$ for any $d_p < \infty$ and robot translation cannot be recovered.

Based on these theorems, ill-conditioned situations are identified:

Proposition 1.- Depth of features aligned with robot trajectory cannot be recovered. This situations is formalized in Corollaries 1.1 and 1.2.

Proposition 2.- Depth cannot be recovered with pure rotation motions as shown in Corollary 2.1.

Proposition 3.- Landmarks at infinity give robot orientation, but no translation information can be obtained from them. This is based on Corollary 2.2.

Feature estimates calculated when the depth computation problem is ill-conditioned present high covariances and great estimation errors which may cause linealization problems. Once a feature has been wrongly initialized, new observations taken from robot poses not aligned with the feature will not be able to correct its position. If a *cartesian* parameterization is used, an additional problem is that features with infinite depth cannot be represented and their initialization must be deferred. This situation is formalized in Corollary 1.3.

4.1 Undelayed Initialization

The undelayed initialization consists in the introduction of landmarks into the system the first time the landmark is observed. This technique presents many benefits since the information attached to a landmark can be used earlier and it allows the use of landmarks which may never been initialized if a delayed strategy is used. Since the first time a landmark is observed only bearing information is available, undelayed techniques must deal with the problem of creating a representation for the depth and its associated uncertainty.

If an *inverse-depth* parameterization is used, landmarks are introduced using a fixed initial depth and an uncertainty representation is created which covers all

depths from some d_{min} to infinity. This initial depth must be adjusted depending on the workspace.

Since *cartesian* parameterization requires low covariances, an *undelayed* initialization is only possible if multiple hypothesis in depth are created (Kwok and Dissanayake, 2004), (Kwok et al., 2007), (Sola et al., 2005). All these approaches present a high complexity and size of the map. Due to this complexity, approaches using *undelayed* initialization together with *cartesian* parameterization are no longer analyzed in this paper.

4.2 Delayed with Two Observations

This delayed technique consists in the combination of the first two observations of a landmark to recover its position using a triangulation algorithm. This is a not purely delayed technique, since there are no conditions which must be satisfied by the observations in order for the landmark to be initialized, and all landmarks are introduced in the map provided that they are observed from at least two robots poses. The main benefit of this initialization strategy is that the solution is independent on the workspace. However, triangulation algorithms used to recover the landmark position are highly non-linear and, depending on the arrangement of robot poses and features, the problem may be *ill-conditioned*.

If a *cartesian* parameterization is used, the recovered feature position must be near-Gaussian and covariances must be small. For this reason, additional tests are used to check that features satisfy these conditions. If features are parameterized using *inverse-depth*, this strategy may suppose a benefit in the sense that it is independent on the size of the scene. Therefore higher covariances in the estimates are admissible and recovered features are near-Gaussian even for low parallaxes.

4.3 Delayed until Condition

In a pure delayed initialization technique, observations of landmarks are accumulated and its initialization is deferred until a condition of Gaussianity is satisfied; then observations are used to create a representation for the feature (Bailey, 2003), (Klippenstein et al., 2007).

If a delayed initialization is used, some landmarks may never been initialized. Since the information provided by landmarks cannot be used until the landmark is initialized, a delayed technique decreases the amount of information available to improve robot the pose. Many delayed techniques present a high computational cost to calculate the condition, and have

their own problems and limitations. The main benefit is that the representation for the landmark is more accurate and reliable than the obtained by an undelayed strategy.

5 DISCUSSION

As told, the aim of this work is the comparison of *cartesian* and *inverse-depth* parameterizations combined with *delayed* and *undelayed* initialization techniques. These have been selected because are the most commonly used, being also simple and of low computational complexity.

5.1 Inverse-Depth Undelayed

This technique is an adaptation to the 2D situation of the technique described in (Montiel et al., 2006). A feature \mathbf{x}_i is introduced into the map using a single observation. The current robot pose $\mathbf{x}_r = (x_r, y_r, \theta_r)$ is used together with the observation z_i and an initial depth ρ_0 parameterized in *inverse-depth* to get the feature representation \mathbf{x}_i . This depth is worked out using a minimal distance d_{min} which must be selected depending on the workspace:

$$\rho_{min} = \frac{1}{d_{min}}; \rho_0 = \frac{\rho_{min}}{2}; \sigma_\rho = \frac{\rho_{min}}{4} \quad (7)$$

where ρ_{min} is the inverse of depth, ρ_0 is the initial inverse-depth, which is the middle value of the interval $[0, \rho_{min}]$, and σ_ρ is the standard deviation used to initialize ρ_0 (95% of ρ is in the interval $[\rho_0 - 2\sigma_\rho, \rho_0 + 2\sigma_\rho] = [0, \rho_{min}]$.) The initial value of the feature is calculated as:

$$\mathbf{x}_i = \mathbf{g}(\mathbf{x}_r, z_i, \rho_0) = (x_r, y_r, \theta_r + z_i, \rho_0) \quad (8)$$

5.2 Inverse-Depth Delayed with Two Observations

As a proposal, an *inverse-depth* parameterization (Montiel et al., 2006) is combined with a delayed initialization technique where the second observation is used to calculate the initial depth for the feature. The position for the feature \mathbf{x}_i which has been observed from \mathbf{x}_{r_1} and \mathbf{x}_{r_2} producing measurements z_{1i} and z_{2i} is calculated as follows:

$$\begin{aligned} \mathbf{x}_i &= \mathbf{g}(\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, z_{1i}, z_{2i}) = (x_{r_2}, y_{r_2}, \theta_{r_2} + z_{2i}, \rho_0) \\ \rho_0 &= \frac{s_2 * c_1 - c_2 * s_1}{(y_{r_1} - y_{r_2}) * c_1 - (x_{r_1} - x_{r_2}) * s_1} \end{aligned} \quad (9)$$

where $c_j = \cos(\theta_j + z_{ji})$ and $s_j = \sin(\theta_j + z_{ji})$, for $j = 1, 2$.

An additional test is used in order to detect situations where inverse-depth cannot be recovered and intersections take place in the opposite direction of the observation. In these situations, the initialization is deferred.

5.3 Cartesian Delayed with Two Observations

Given the first two observations z_{1i}, z_{2i} of a landmark \mathbf{x}_i taken from robot poses $\mathbf{x}_{r_1}, \mathbf{x}_{r_2}$, the landmark position $\mathbf{x}_i = (x_i, y_i)$ is calculated as follows (Bailey, 2003):

$$\begin{aligned} x_i &= \mathbf{g}_1(\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, z_{1i}, z_{2i}) = \frac{x_{r_1} s_1 c_2 - x_{r_2} s_2 c_1 - (y_{r_1} - y_{r_2}) c_1 c_2}{s_1 c_2 - s_2 c_1} \\ y_i &= \mathbf{g}_2(\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, z_{1i}, z_{2i}) = \frac{y_{r_2} s_1 c_2 - y_{r_1} s_2 c_1 + (x_{r_1} - x_{r_2}) s_1 s_2}{s_1 c_2 - s_2 c_1} \end{aligned} \quad (10)$$

where $c_j = \cos(\theta_j + z_{ji})$ and $s_j = \sin(\theta_j + z_{ji})$, for $j = 1, 2$.

Similarly a test is used to check that features can be recovered and intersections of bearings are not in the opposite direction of the observations.

5.4 Cartesian/Inverse-Depth Delayed until Finite Depth

A *delayed* technique is proposed where feature initialization is deferred until finite uncertainty in depth can be estimated.

This is achieved by a simple test which compares two observation rays and checks if they are parallel. This situation is characterized by Corollaries 1.1 and 1.3. When observation rays are parallel, the uncertainty in depth of the recovered landmark extends to infinity and the initialization is deferred. This test is especially useful when a *cartesian* parameterization is used, since infinite depths cannot be modeled.

Let $\mathbf{x}_{r_j} = (x_{r_j}, y_{r_j}, \theta_{r_j})$, for $j = 1, 2$ be the two robot poses where observations z_{ji} , for $j = 1, 2$ to a landmark \mathbf{x}_i were taken. Global bearings α_{ji} , for $j = 1, 2$ to the landmark are calculated as:

$$\alpha_{ji} = \theta_{r_j} + z_{ji} \quad (11)$$

If we name $S_{\alpha_{ji}}$ the linearized propagated covariance for bearing α_{ji} then the Chi-squared test for Finite Depth is expressed as:

$$\frac{(\alpha_{1i} - \alpha_{2i})^2}{S_{\alpha_{1i}} + S_{\alpha_{2i}}} > \chi_{0.99, 1d.o.f}^2 \quad (12)$$

5.5 Cartesian/Inverse-Depth Delayed until Feature Not Aligned with Robot Poses

As stated in Proposition 1, the initialization of features aligned with the robot trajectory is problematic when working with bearing-only data. When a feature is observed from two robot poses which are in line with the feature, it is not possible to make a right depth initialization. Corollary 1.1. gives a formal explanation of this situation: feature is aligned with robot trajectory when the observation rays are parallel and the robot translation takes place in a direction which is parallel to the observation.

Let $\mathbf{x}_{r_j} = (x_{r_j}, y_{r_j}, \theta_{r_j})$, for $j = 1, 2$ be the two robot poses where observations to a landmark \mathbf{x}_i were taken. From here α_{1i}, α_{2i} , for $j = 1, 2$ can be computed with equation 11. Let $S_{\alpha_{ji}}$, for $j = 1, 2$ be their linearized propagated covariances. Observation rays are parallel when:

$$\frac{(\alpha_{1i} - \alpha_{2i})^2}{S_{\alpha_{1i}} + S_{\alpha_{2i}}} \leq \chi_{0.99, 1d.o.f}^2 \quad (13)$$

The robot trajectory from \mathbf{x}_{r_1} to \mathbf{x}_{r_2} has a global inclination which can be calculated as:

$$\theta_t = \arctan\left(\frac{y_{r_2} - y_{r_1}}{x_{r_2} - x_{r_1}}\right) \quad (14)$$

Let S_{θ_t} be the linearized propagated covariance for bearing θ_t . The trajectory is parallel to the observation rays when:

$$\frac{(\theta_t - \alpha_{ji})^2}{S_{\theta_t} + S_{\alpha_{ji}}} \leq \chi_{0.99, 1d.o.f}^2 \quad (15)$$

for $j = 1, 2$.

The initialization of features is deferred until a pair of observations is available where the feature is not aligned with the trajectory. This delayed technique is less restrictive than the explained in section 5.4 and is specially useful for an *inverse-depth* parameterization since it allows the initialization and the use of features of infinite depth.

6 EXPERIMENTS

In order to analyze the performance of the different parameterizations and initialization techniques, some experiments have been designed so that the performance and robustness of the algorithms can be analyzed.

The experimentation and analysis of results is carried out using a simulator which presents many benefits. First of all, exactly the same experiment can be solved by several algorithms so that results are fully comparable. Besides, ground truth information is available to compare with the obtained results.

Some preliminary experiments have been carried out using omnidirectional images which can be seen in Figure 1. The matches have been obtained using SURF descriptors (Murillo et al., 2007).

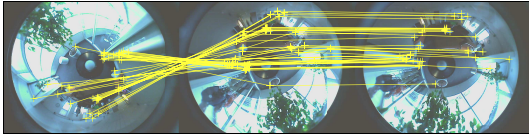


Figure 1: Omnidirectional image: feature extraction and matching.

In the simulated experiments, an observation noise with an standard deviation of 0.125 degrees is used. Features are placed on the walls of a squared room.

An initialization to the system is introduced from three robot poses and the first 5 observed landmarks. It is based on SFM techniques with the Trifocal Tensor (Sagués et al., 2006). The data association problem is not discussed in this paper and data association is supposed to be perfect.

Algorithms have been tested in different scenarios and under different conditions of *visibility*, *trajectory* and *step sizes*. The *Visibility* affects to the number of visible landmarks. Two possibilities are evaluated: *Total*, where all features are visible from all robot poses and *Section*, where the workspace is divided into four sections; In every step robot observes the features within its section and a few from the neighborhood in order to connect the sections. When the visibility is *Total*, no loop closing takes place and distant features are used.

As stated in section 4 the *Robot Trajectory* has a big influence on depth computation in such a way that if landmark is on the direction of robot translation, depth computation is an undetermined problem. Two trajectories have been evaluated. The first is an *Squared* trajectory composed by several pure translation motions and four 90° pure rotations. In this trajectory some features are aligned with the robot movement for many steps. The odometry noise is introduced as a function of the step size (st) and it can be seen in columns *Pure translation* and *Pure rotation* of Table 1. The second trajectory is *Circular*: Robot describes a circumference when moving along the environment which supposes mixed rotations and translations. No feature in the map is observed in line with the trajectory. The standard deviations of the odome-

try noise are shown in column *Mixed motion* of Table 1,

Table 1: Odometry noise relative to the step size (st).

Standard deviation	Pure translation	Pure rotation	Mixed motion
x_r	$0.01 * st$	$0.03 * st$	$0.03 * st$
y_r	$0.01 * st$	$0.03 * st$	$0.03 * st$
θ_r	2°	2.5°	2.5°

The *Step Size* determines the distance (in meters) between two consecutive robot poses. This is the parameter which affects the most the behavior of algorithms. Step sizes of 0.125 m, 0.250 m, 0.5 m and 1 m are tested.

6.1 Analyzed Information

The variables used in order to analyze the performance of an algorithm are listed below.

Final Divergence. Percent of results where the final robot pose diverges from its estimation. The condition which is tested for each component (x_r, y_r, θ_r) independently can be written as

$$\frac{(a - \hat{a})^2}{P} > \chi_{0.99, 1d.o.f.}^2 \quad (16)$$

a being (x_r, y_r, θ_r) the ground-truth, \hat{a} the estimated value for variable a and P its estimated covariance.

Map Consistency. Percent of features in the final map whose estimation is *consistent* with the ground truth. A feature is considered *consistent* if the estimation error in its x_i or y_i coordinate satisfy:

$$\frac{|a - \hat{a}|}{\sqrt{P \chi_{0.99, 1d.o.f.}^2}} \leq 1.5 \quad (17)$$

where the variable a represents the x_i or y_i coordinates.

Trajectory Divergence. Percent of steps in the trajectory where the estimation of the robot pose (x_r, y_r, θ_r) diverges.

Feature Initialization Step. Average of the number of steps needed to initialize a feature, calculated as the difference between the step when a feature is first observed and the one when the feature is introduced into the map.

Feature Usage. Average of the feature used per step calculated as the percentage of features used in the

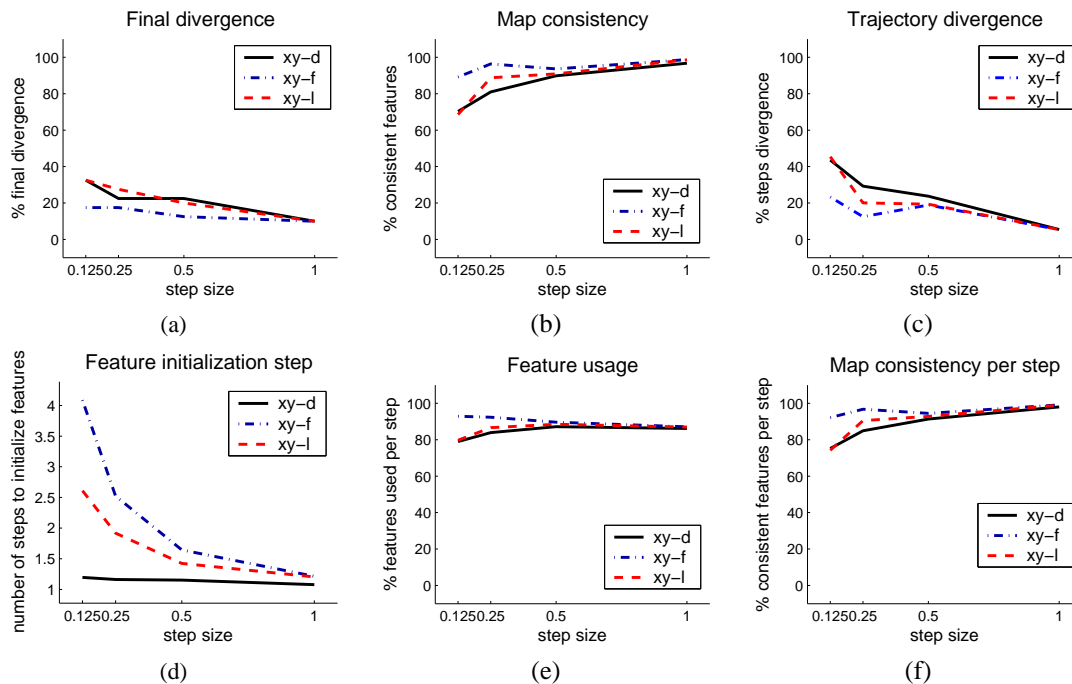


Figure 2: Cartesian delayed techniques comparison. Analysis of the results for different step sizes (x -axis). The algorithms used are *cartesian delayed*. **xy-d**: with two observations. **xy-f**: until finite depth. **xy-l**: until feature not aligned with robot poses.

filter update versus the features observed.

Map Consistency per Step. Average of the percent of consistent features in the map in every step.

Additionally, information related to the precision and error of the final robot pose, the trajectory and the final map has been also studied.

6.2 Results

A total of 160 experiments have been designed, and all of them have been solved using the available algorithms discussed in section 5. For the Inverse-depth undelayed, a minimal depth $d_{min} = 0.5m$ is used.

The results are analyzed in three different blocks. In the first we compare the *cartesian delayed* algorithms. In the second, we compare all *inverse-depth delayed* approaches and in the third block, a global comparison is carried out where the best of the *cartesian delayed* algorithms and the *inverse-depth delayed* algorithms are compared to the *inverse-depth undelayed* algorithm.

6.2.1 Cartesian Delayed Comparison

The results obtained by the *cartesian delayed* algorithms can be found in Figure 2. The cartesian delayed until finite depth (**xy-f**) algorithm performs bet-

ter than the delayed with two observations (**xy-d**) and the delayed until features not aligned (**xy-l**) methods: the final divergence (Figure 2.a) and trajectory divergence (Figure 2.c) are the lowest for all step sizes, the map consistency (Figure 2.b, Figure 2.f) are the highest, and the number of features used to update (Figure 2.e) is higher than the used by the other cartesian algorithms for all step sizes even though this algorithm needs more steps to initialize a feature (Figure 2.d).

6.2.2 Inverse-depth Delayed Comparison

From the study of the results obtained by the inverse-depth delayed algorithms, we can observe that all algorithms performed in a very similar way (Figure 3). The final divergence (Figure 3.a), map consistency (Figure 3.b), trajectory divergence (Figure 3.c), feature usage (Figure 3.e) and map consistency per step (Figure 3.f) results are similar for all inverse-depth delayed algorithms. Only the feature initialization step (Figure 3.d) differs, due to the use of the different delayed strategies.

An especial study is carried out in order to compare the capability of the inverse-depth algorithms to deal with features which are observed during many steps aligned with the trajectory. The most critical situation is when the robot moves following an squared trajectory and only observes landmarks within its sec-

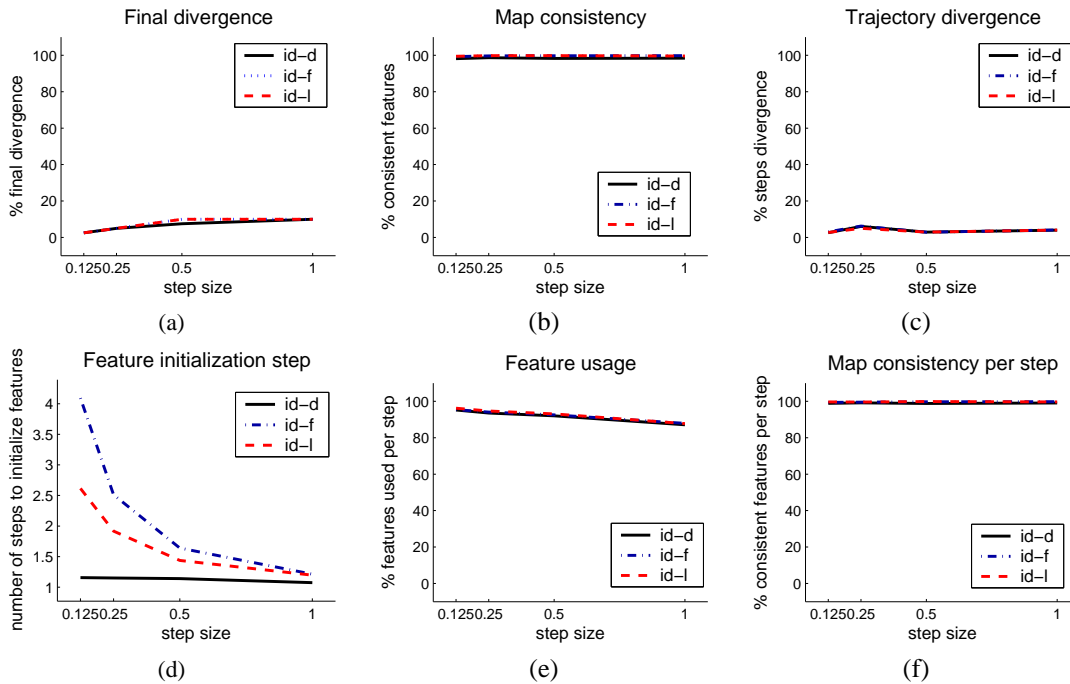


Figure 3: Inverse-depth delayed comparison. Analysis of the results for different step sizes (x -axis). The algorithms are: id-d: with two observations. id-f: until finite depth. id-l: until feature not aligned with robot poses.

tion. In this situation the problematic features are F12, F23, and F34 (Figure 4). In this figure, the ground-truth robot trajectory and landmark positions are displayed in red, while the estimates and uncertainties calculated by the algorithms are drawn in blue. As can be observed, both the trajectory and the landmark positions have been correctly estimated in all cases. However, features F12, F23 and F34 present high uncertainty (Figure 4.a) when the algorithm used is the *inverse-depth with two observations* (**id-d**).

Paying attention to the problematic features (F12, F23, F34) in Figure 4 we can observe the results of an earlier initialization of features which are in line with the trajectory. Even though their initial estimate and covariance correctly represent the feature position, posterior observations are not able to correct its position due to the huge innovation.

The *Inverse-depth delayed until finite depth* (**id-f**) and *Inverse-depth delayed until feature not aligned with robot poses* (**id-l**) performed in a similar way. However, the second is preferred because of its capability to initialize and use features of infinite depth.

6.2.3 Global Comparison

As can be observed in Figure 5, the behavior of the inverse-depth undelayed algorithm (**id-u**) is seriously affected by the step size. For the smallest step size (0.125m), almost all experiments converged in the last

robot pose (Figure 5.a) while for the other step sizes, many experiments diverged. The number of consistent features in the final map (Figure 5.b) is lower than for the other algorithms. This behavior is also observed for the number of consistent features per step (Figure 5.f).

The cartesian delayed until finite depth algorithm (**xy-f**), its behavior is not so much affected by the step size but we can observe a better performance when the step size increases: the final divergence (Figure 5.a) is slightly higher for smaller step sizes. The number of consistent features in the final map (Figure 5.b) and along the steps (Figure 5.f) slightly decreases for smaller step sizes. The feature usage (Figure 5.e) remains high for all step sizes.

The inverse-depth delayed until features not aligned algorithm (**id-l**) produced the best results, exhibiting an stable behavior for all step sizes: almost all experiments converged (Figure 5.a) and also along the trajectory (Figure 5.c). Almost all features are consistent in the final map (Figure 5.b) and along the steps (Figure 5.f), and the feature usage is the highest (Figure 5.e).

An interesting information about the features usage can be extracted from Figure 5.d and Figure 5.e: it can be observed that when an undelayed strategy is selected, the percent of features used to update the map in every step (Figure 5.e) is much lower than the

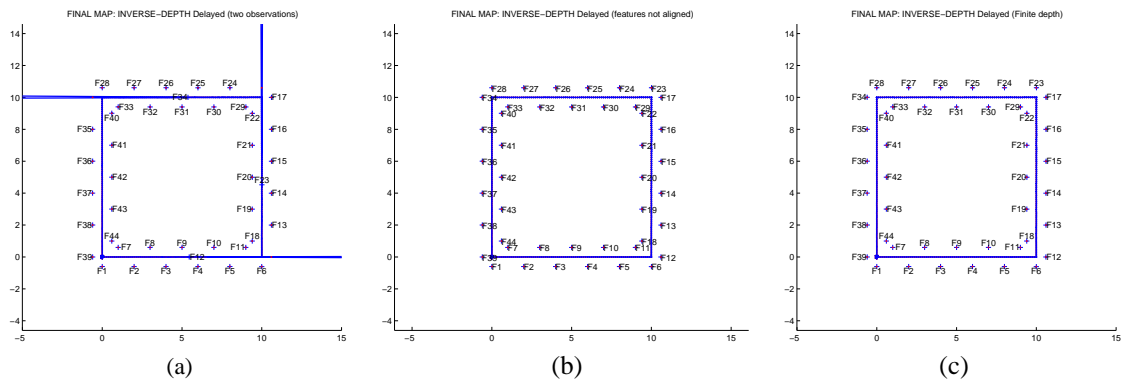


Figure 4: (a) Inverse-depth delayed with two observations. (b) Inverse-depth delayed until feature not aligned with robot poses. (c) Inverse-depth delayed until finite depth.

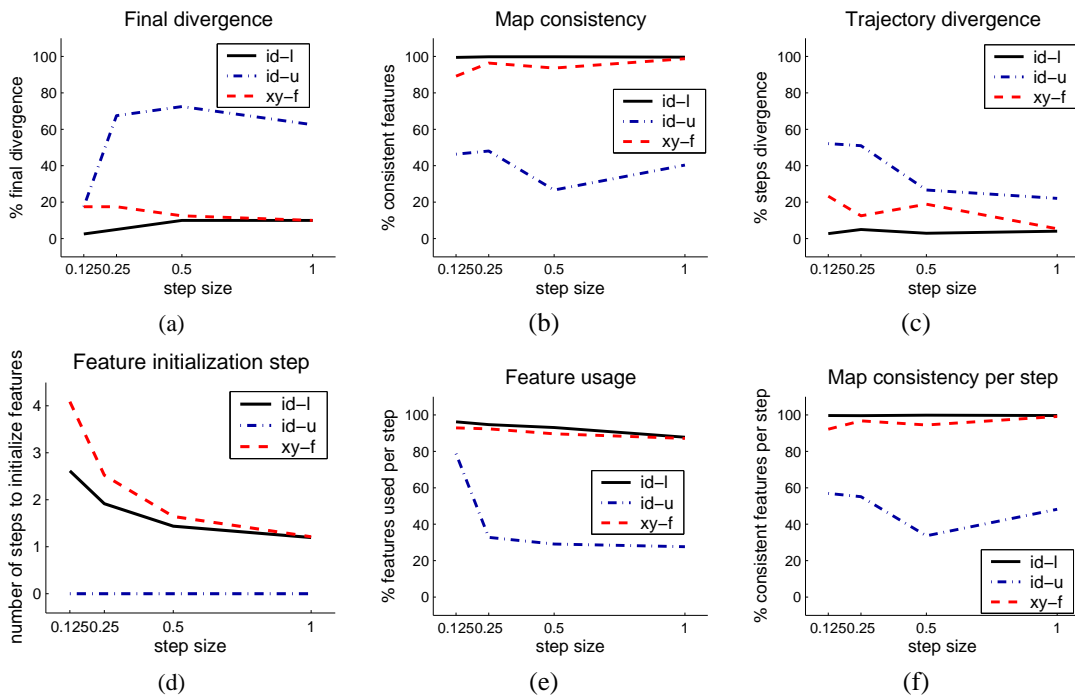


Figure 5: Global comparison. Analysis of the results for different step sizes (x -axis). The algorithms used are: id-u: *inverse depth undelayed*, $d_{min} = 0.5m$. xy-f: *cartesian delayed until finite depth*. id-l: *inverse depth delayed until feature not aligned with robot poses*.

used by the delayed algorithms even though features initialization requires a lower number of steps (Figure 5.d.) Therefore, delayed techniques provide important benefits due to the fact that the initial estimates introduced into the map are better with lower covariance.

7 CONCLUSIONS

In this paper we have discussed feature parameterization and initialization using bearing-only measurements. Both considerably affect the results of the algorithms. However, this paper shows that even with a perfect feature parameterization, if the initialization problem is ill-conditioned the results are inconsistent. As conclusion we can state that in general situations the delayed inverse depth until features not aligned performs competitively.

An interesting result of this study is the related

to the cartesian parameterization when it is combined with a finite depth test. It was expected that cartesian algorithm based in triangulation techniques were to suffer a great degradation of their performance for small step sizes. However, results show that the algorithm delayed until finite depth with cartesian parameterization is not very sensitive to the step size and exhibits very competitive results, which makes it an appropriate algorithm for indoors. Other interesting conclusion is that introducing features earlier in the EKF does not mean that more/better information will be available to update the state.

In this paper we have also stated ill-conditioned situations: a pure rotation motion and features aligned with the trajectory. None of them can be managed in any case. Some ideas have been presented to detect these situations which will allow the algorithms to decide which data can be used in each step.

ACKNOWLEDGEMENTS

This work was supported by projects MEC DPI2006-07928 and IST-1-045062-URUS-STP.

REFERENCES

- Bailey, T. (2003). Constrained initialisation for bearing-only slam. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 1996–1971, Taipei, Taiwan.
- Costa, A., Kantor, G., and Choset, H. (2004). Bearing-only landmark initialization with unknown data association. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1164–1770.
- Davison, A. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Proc. Ninth IEEE Intl. Conf. on Computer Vision*.
- Klippenstein, J., Zhang, H., and Wang, X. (2007). Feature initialization for bearing-only visual slam using triangulation and the unscented transform. In *Proc. IEEE Intl Conf. on Mechatronics and Automation*, pages 157–164, Harbin, China.
- Kwok, N. and Dissanayake, G. (2004). An efficient multiple hypothesis filter for bearing-only slam. In *Proc. IEEE/RSJ Intl. Conf. on Robotics and Systems*, volume 1, pages 736–741, Sendai, Japan.
- Kwok, N. M., Ha, Q. P., Huang, S., Dissanayake, G., and Fang, G. (2007). Mobile robot localization and mapping using a gaussian sum filter. *Intl. Journal of Control, Automation, and Systems*, 5(3):251–268.
- Montiel, J. M. M., Civera, J., and Davison, J. (2006). Unified inverse depth parametrization for monocular slam. In *Robotics Science and Systems, RSS*, Philadelphia, Pennsylvania.
- Murillo, A. C., Guerrero, J. J., and Sagüés, C. (2007). Surf features for efficient robot localization with omnidirectional images. In *IEEE/RSJ Int. Conf. on Robotics and Automation*, pages 3901–3907.
- Neira, J. and Tardós, J. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890897.
- Sagüés, C., Murillo, A. C., Guerrero, J. J., Goedemé, T., Tuytelaars, T., and Gool, L. V. (2006). Localization with omnidirectional images using the 1d radial trifocal tensor. In *Proc of the IEEE Int. Conf. on Robotics and Automation*, pages 551–556.
- Sola, J., Monin, A., Devy, M., and Lemaire, T. (2005). Undelayed initialization in bearing only slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*.
- Trawny, N. and Roumeliotis, S. (2006). A unified framework for nearby and distant landmarks in bearing-only slam. In *Proc. IEEE Intl. Conf. on Robotics and Automation*.

EFFICIENT PLANNING OF AUTONOMOUS ROBOTS USING HIERARCHICAL DECOMPOSITION

Matthias Rungger, Olaf Stursberg

Institute of Automatic Control Engineering, Technische Universität München, 80333 Munich, Germany
{matthias.rungger, stursberg}@tum.de

Bernd Spanfelner, Christian Leuxner, Wassiou Sitou

Software and System Engineering, Technische Universität München, 80333 Munich, Germany
{leuxner, sitou, spanfelner}@in.tum.de

Keywords: Hierarchical Planning, Autonomous Robots, Context Adaptation, Hybrid Models, Predictive Control.

Abstract: This paper considers the behavior planning of robots deployed to act autonomously in highly dynamic environments. For such environments and complex tasks, model-based planning requires relatively complex world models to capture all relevant dependencies. The efficient generation of decisions, such that realtime requirements are met, has to be based on suitable means to handle complexity. This paper proposes a hierarchical architecture to vertically decompose the decision space. The layers of the architecture comprise methods for adaptation, action planning, and control, where each method operates on appropriately detailed models of the robot and its environment. The approach is illustrated for the example of robotic motion planning.

1 INTRODUCTION

The spectrum of applications for future robots will continuously grow, and an increasing percentage will be employed in dynamic environments. Examples are service robots which assist elderly or disabled persons and rescue robots which operate in hostile areas that are devastated by earthquakes or similar incidents. A characteristic of such applications is that an autonomous planning of actions must consider changing environment conditions and include reliable and immediate decisions on which available resources should be employed to fulfill a momentary task. In particular, if the environment includes dynamic objects with non-deterministic or partly unpredictable behavior, the representation of the constraints for a planning problem becomes complex and has a significant impact on the realtime-computability of action plans. Another difficulty encountered in this case is that environment models which are purely identified based on data series measured over a short period of time for a specific situation are merely suitable to reflect the behavior of the environment sufficiently well.

In order to cope with the issues of model complexity and quality of prediction within action planning, this paper proposes a planning architecture which combines multi-layer decision making with the use of

a knowledge base for storing learned goal-attaining action strategies. Of course, the idea of hierarchical planning architectures is not new in general, and corresponding approaches can be found, e.g., in (Nau et al., 1998), (Galindo et al., 2007), (Barto and Mahadevan, 2003). However, the novel contribution of this paper is that a three-tier scheme is suggested which separates the planning task into steps of configuring the system structure, of planning a sequence of actions, and refining these actions into locally optimal control trajectories. The configuration step allocates the set of resources (sensors, actuators, or algorithms for planning and control) that seem most appropriate to accomplish a given task. By employing reinforcement learning, the action planning on a medium layer produces a sequence of discrete actions that qualitatively accomplish the task. The third layer refines an action plan by generating control trajectories which correspond to the action plan and establish a quantitative setting for the actuators over time. The overall concept setting is explained in the following section.

2 HIERARCHICAL PLANNING APPROACH

The proposed hierarchy vertically decomposes a complex planning task for reduction of complexity wherever possible, and consists of the layers shown in Fig.1: The adaptation-layer perceives and evaluates the current situation based on measured data and decides on the configuration, i.e. the resources selected to solve a current task. The planning layer considers the allocated resources and existing constraints for calculating an action plan that accomplishes the task. In the opposite direction, information about the existence of a feasible plan for the chosen configuration is transmitted from the planning to the adaptation layer. The control layer refines the action sequences received from the planning layer by computing a control trajectory for each action. The control trajectory is then passed to the corresponding actuators of the system. If no feasible trajectory is found for an action (or a part of the action sequence), this information is passed by to the planning layer and triggers replanning. Thus, the three layers continuously interact during the online execution to compute a task accomplishing strategy. To enable this computation, the system must have the capability to predict the behavior of itself and its environment up to a point of time in which the task is accomplished or cannot be accomplished anymore. These predictions are obtained from evaluating the models shown in Fig. 1 for a choice of configurations, action sequences, and control strategies.

The hierarchy implements a nested feedback loop also in the following sense: as soon as a current situation changes (e.g. because either a task changes or a relevant change of the environment is detected) the previously generated configuration, action plan, or control trajectory is re-evaluated and possibly modified. The reactivity to the behavior of the environment requires continuous update of the models based on measurement signals from the sensors of the autonomous system. On each layer, appropriate identi-

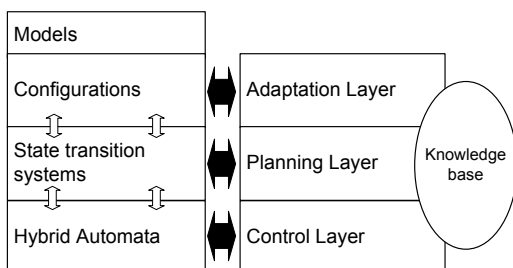


Figure 1: The hierarchical architecture.

fication techniques are included to reparametrize the models to the momentarily perceived situation.

In summary, the main objective of the three-tier architecture is (a) to decompose the decision making into three qualitatively distinct categories, (b) to select components, algorithms, and behavior in a top-down manner to reduce the search effort in the decision space, and to (c) include a knowledge-base by which already 'experienced' (or learned) behavior is used as heuristics for efficiently deciding which strategy is goal-attaining. For this scheme, the claim is not to outperform single state-of-the-art algorithms on a particular layer, but to provide an architecture for proper integration of different algorithms to solve a broad variety of complex planning tasks.

3 ADAPTATION LAYER

The term *adaptation* is here understood as the capability of an autonomous system to react to changing tasks or varying aspects of the context in which the system is embedded. The term 'context' refers to the state of the system environment, as e.g. the proximity of obstacles to a moving robot (the 'system'). The adaptation on the uppermost layer of the hierarchy means here to deduce from the current context and a task to be accomplished a suitable *configuration*, which is a subset of available components, i.e. available hardware devices (e.g. actuators) or software algorithms for action planning and control.

Components. On the adaptation layer, the system and its environment are described in terms of communicating units called *components*. Formally, a system is defined as a pair (C, CH) with C as a set of components and CH as a set of channels. The components communicate through directed channels, which are defined by $ch = C \times C \times Id$ with Id as a set of unique names. A system is completely determined by a network of components connected via channels, and the components send messages along channels and thereby express their behavior. Different ways to describe such behavior exist, like e.g. process algebras or relations on inputs and outputs like FOCUS (Broy and Stoelen, 2001) or COLA (Haberl et al., 2008). The *behavior* of a component is here specified by a sequence of data $msg \in MSG$ which is received and sent over the ports I and O of the component.

Adaptation Mechanism. To formalize the adaptation of a component-based model, the notion of *Mode Switch Diagrams* (MSD) is introduced. An MSD is

a tuple $(M, \text{map}, \delta, m_0)$ defining a transition system with a set of modes M , a transition function δ , a function $\text{map} : m \rightarrow C$ that relates modes to components, and m_0 is the initial mode. The transitions are defined by $\delta : M \times P \rightarrow M$ with predicates P depending on the inputs $i \in I$ of the components assigned to a mode. The function δ encodes the transition of the MSD between two modes m_i and m_{i+1} , and it represents the *adaptation*. An example of an adaptation is shown in Fig. 2. An activated mode m_{i+1} determines the components of the system which are selected until a new context change triggers another transition. The mode constitute the frame for the action planning carried out on the medium layer.

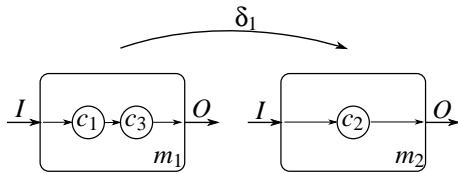


Figure 2: Transition between two modes m_1 and m_2 of an MSD example.

4 PLANNING LAYER

The planning layer comprises algorithms that search for a sequence of actions which transfers the initial state of the system or environment into a desired goal state. The specific algorithm to be used for a given task, the available set of actions, and the relevant model is determined by the mode information received from the adaptation layer. The model used on the planning layer is specified as a state transition system $\Sigma = (S, A, \delta)$ consisting of a set of states $S = \{s_1, s_2, \dots, s_n\}$, a set of actions $A = \{a_1, a_2, \dots, a_m\}$, and a transition function $\delta : S \times A \rightarrow 2^S$. The planning task is mapped into a goal state s_G (or a set of goal states $S_G \subseteq S$, respectively) which has to be reached from a current state $s_0 \in S$. A *plan* is a sequence of actions that evokes a state path ending in s_G (or an $s \in S_G$).

The planning techniques on the medium layer follow the principle of *reinforcement learning*. In order to formulate the planning problem as optimization, an utility function $r : S \times S \rightarrow \mathbb{R}$ is introduced, which defines the reward of taking a transition of Σ . The goal is encoded implicitly by assigning a high reward to a state sequence via which a goal state is reached, and by producing a negative reward for undesired behavior of Σ . Reinforcement learning represents learning from interaction, i.e. the system learns a strategy (as a state-to-action mapping) based on the reward. At time t , the system observes the state $s \in S$, and for a cur-

rent policy $\pi = P(a|s)$ an action $a \in A$ is chosen. The system moves from state s to s' according to the transition function, and it receives the reward r . The goal is to learn a policy π which maximizes the cumulative discounted future reward. By introducing the action-value function $Q(s, a)$, a measure is available which estimates the profit of taking an action a in state s . The action-value function under policy π is given by the Bellman equation:

$$Q^\pi(s, a) = [R(s'|s, a) + \gamma \sum_{a'} \pi(s', a') Q^\pi(s', a')],$$

where s' and a' denote state and action at the next time instant. γ is the discount factor and is given by $0 < \gamma < 1$. $R(s'|s, a)$ denotes the reward for taking action a in state s , resulting in the new state s' . If the reward function as well as the system dynamics are known, the optimal policy

$$Q^* = \max_{\pi} Q^\pi(s, a)$$

can be calculated explicitly, resulting in a system of $|S| \cdot |A|$ equations. Since this is often computationally intractable (in addition to some dynamics may not be known before-hand) approximations to the optimal action-value function are used.

One possible algorithm to estimate $Q^*(s, a)$ is called SARSA (Takadama and Fujita, 2004), which learns the current state-value function $Q(s, a)$ by updating the function with the observed reward, while interacting with the environment:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q^\pi(s', a') - Q(s, a)],$$

where α is the learning rate. SARSA is a so called *on-policy* RL algorithm, which updates the action-value function while following a policy π . The policy is ϵ -*greedy*, where the greedy action $a^* = \arg \max_a Q(s, a)$ is selected most of the time. Once in a while, with probability ϵ , an action is chosen randomly. SARSA is used here as RL technique in order to reduce the “risk” by avoiding negative rewards while paying with a loss of optimality (Takadama and Fujita, 2004).

5 CONTROL LAYER

The control layer ensures the correct execution of the action sequence derived on the planning layer. It establishes a connection of the discrete actions received from the medium layer to the real world by applying continuous signals to the system’s actuators. The combination of discrete actions with continuous dynamics motivates the use of *hybrid automata* (Henzinger, 1996) to model the behavior on the control layer. Hybrid automata (HA) do not only establish

an adequate interface to the higher layers, but also provide the necessary expressivity required to model, e.g. robot-object-interaction. For each component selected by the adaptation layer, one hybrid automaton is introduced, and the various automata can interact via synchronization or shared variables.

Using a variant of HA with inputs according to (Stursberg, 2006), a hybrid automaton modeling the system is given by $HA = (X, U, Z, inv, \Theta, g, f)$, with X as the continuous state space, U the input space, Z the set of discrete locations, inv the assignment of invariance sets for the continuous variables of the discrete locations, and Θ the set of discrete transitions. A mapping $g : \Theta \rightarrow 2^z$ associates a guard set with each transition. The discrete-time continuous dynamics f defining the system dynamics is $x(t_{j+k}) = f(z(t_k), x(t_k), u(t_k))$. At any time t_k , the pair of continuous and discrete state forms the current hybrid state $s(t_k) = (z(t_k), x(t_k))$. Hybrid automata for modeling relevant components of the environment introduced without input sets U (since not directly controllable).

Model Predictive Control. In order to generate control trajectories for the HA, the principle of *model predictive control* (MPC) is used (Morari et al., 1989). Considering the control problem not only as a motion planning problem like typically done in the robotic domain, e.g. (LaValle, 2006), but as an MPC problem has the following advantages: (1) an optimal solution for a given cost function and time horizon is computed, (2) model-based predictions for the behavior of the system and the environment lead to more reliable and robust results, and (3) a set of differential and dynamic constraints can be considered relatively easy. The MPC scheme solves, at any discrete point of time t_h , an optimization problem over a finite prediction horizon to obtain a sequence of optimal control inputs to the system. The optimization problem considers the dynamics of the system and the following additional constraints: a sequence of *forbidden regions* $\phi_{F,k} = \{F_k, F_{k+1}, \dots, F_{k+p}\}$, and a sequence of *goal regions* $\phi_{G,k} = \{G_k, G_{k+1}, \dots, G_{k+p}\}$ are specified over the prediction horizon $p \cdot (t_k - t_{k-1})$. Here, F_k denotes a state region that the system must not enter, and G_k is the state set into which the system should be driven. The constrained optimization problem can be formulated as:

$$\begin{aligned} \min_{\phi_{u,k}} \quad & J(\phi_{s,k}, \phi_{e,k}, \phi_{u,k}, p, \phi_{G,k}) \\ \text{s.t.} \quad & s(t_j) \notin F_j \quad \forall j \in \{k, \dots, k+p\} \\ & u_{min} \leq u(t_j) \leq u_{max} \\ & \phi_{s,k} \in \Phi_s \text{ and } \phi_{e,k} \in \Phi_e \end{aligned}$$

where $\phi_{s,k}$ and $\phi_{e,k}$ are the predicted state trajectories of the system and the environment, which must be contained in the sets of feasible runs Φ_s and Φ_e . No state $s(t_j)$ contained in $\phi_{s,k}$ must be in a forbidden region $F(t_j)$. u_{min} and u_{max} are the limitations for the control inputs $u(t_j)$ and thus for the control trajectory $\phi_{u,k}$. A possible solution technique for the above optimization problem is the following sequential one: (1) an optimizer selects a trajectory $\phi_{u,k}$, (2) the models of system and environment are simulated for this choice leading (possibly) to feasible $\phi_{s,k}$ and $\phi_{e,k}$, (3) the cost function J is evaluated for these trajectories, and (4) the results reveals if $\phi_{u,k}$ should be further modified for improvement of the costs or if the optimization has sufficiently converged.

Knowledge Base and Learning. In order to improve the computational efficiency, the MPC scheme is enhanced by a knowledge-base, in which assignments of action sequences to situations is stored. The objective of the learning unit with a knowledge base is to reduce the computational effort by replacing or efficiently initializing the optimization. The situations in the knowledge base are formulated as tuple $\delta = (\phi_{s,k}, \phi_{e,k}, \phi_{G,k}, \phi_{F,k}, \phi_{u,k}, J)$, where $s(t_k)$ and $e(t_k)$ again denote the current state of the system and the environment, and $\phi_{u,k}$, $\phi_{G,k}$, and $\phi_{F,k}$ are the sequences of control inputs, the goal, and forbidden sets, respectively. For a given situation at the current time t_k , by using *similarity* comparison¹, the learning unit infers a proper control strategy $\phi_{u,k}^L$ if it exists. Otherwise, the optimization is carried out and the result is stored in the knowledge-base.

6 APPLICATION TO A KITCHEN SCENARIO

The presented hierarchical architecture is applied to a service robot in a kitchen scenario. The task of the robot is to lay a table (see Fig 3), i.e. the robot is expected to drive back and forth between a table and a kitchenette for positioning plates and cutlery on the table. The scenario obviously formulates a very challenging planning task, as the robot has to decide which object to take, how to move to the desired place at the table, and how to avoid collision with humans moving in the same space – this is the motivation for employing a decomposition-based planning approach. To simplify the upcoming presentation, the

¹Similarity is here defined by small distances of the quantities specifying a situation in the underlying hybrid state space.

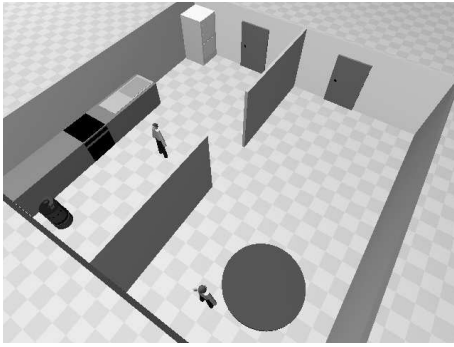


Figure 3: Setting of the kitchen scenario.

further discussion is here restricted here to the part of robot motion planning only.

Adaptation. On the uppermost layer, the set of components C comprises the relevant physical components of the setting and the choice of algorithms being available on the planning and control layers. Thus, the kitchen K , the table T , the kitchenette KN , the refrigerator R , the robot dynamics $B1$, the quantized robot dynamics $B2$, the persons P_1 and P_2 , the reinforcement learning algorithm alg_1 , the MPC algorithm alg_2 , the learning algorithm alg_3 , and the knowledge base kb are elements of the set of components for the motion scenario: $C = \{K, T, KN, R, B1, B2, P_1, P_2, alg_1, alg_2, alg_3, kb\}$. Modes are defined for the component structure like described above, i.e. they represent a particular subset of C according to $map : m \rightarrow C$. Note that modes may only affect parts of the system and that nesting of modes is possible (leading to alternative choices).

For the different steps in the motion planning problem, the adaptation chooses appropriate algorithms by switching to a corresponding mode. For the motion of the robot from one point to another within the kitchen, only one mode m_1 has to be defined. It encodes the path planning by reinforcement learning for the planning layer and the calculation of the corresponding input trajectories to the dynamic system (including learning about situations) for the control layer. For this example, adaptation (in the sense of a switch between modes) occur only if the goal changes due to the current one being achieved or becoming unreachable.

Planning. The states $s \in S$ of the state transition system Σ for this particular example represent different positions of the robot within the kitchen. The actions $a \in A$ on the planning layer denote constant inputs u_i , $i \in \{1, \dots, n\}$ (applied for a fixed time T) for the HA modeling the robot motion on the control layer. A discrete state of Σ represents the motion

for time T , and it correspond to the trajectory obtained from integrating the resulting dynamics $f_z(x, u_i)$ specified on the control layer. Five constant inputs $u_1 = (1, 0)^T$, $u_{2,3} = (0, \pm\pi/4)^T$, $u_{4,5} = (1, \pm\pi/4)^T$ encode pure translation, pure rotation, and curve transition respectively. The state set S of Σ represents a discretization of the floor space of the kitchen (modeled by two continuous variables x_1, x_2 on the lower layer). The discretization follows implicitly from the solution of the continuous dynamics on the control layer.

Choosing a reward of $r = -1$ for every taken action, the SARSA algorithm results in a sequence with a minimum number of actions. If the robot hits a static obstacle a reward of $r = -10$ is used to update the action-value function $Q(s, a)$. The outcome of the algorithm is depicted by the bold dashed line in Fig. 5, resulting from the concatenation of the action primitives. Σ consists in about 1000 states in this particular setup and the RL algorithm converges after approx. 80 runs (see Fig. 4), leading to a trajectory consisting about 20 actions. For a faster convergence, the action-value function $Q(s, a)$ is suitably initialized by a solution that leads to the goal position for the case that no dynamic obstacles are present.

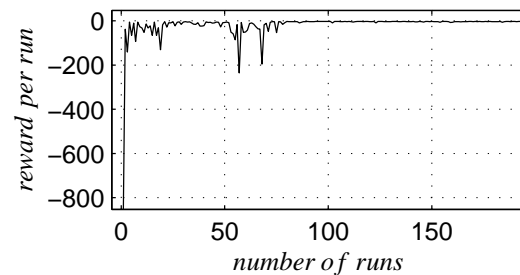


Figure 4: Reward improvement over repetitions of the motion.

Control. The HA used to model the robot that is regarded in the simulation scenario is defined with two locations z_1 and z_2 . Location z_1 defines the kitchen domain, and location z_2 represents that the robot enters a circular region around one of the two moving persons being present in the kitchen; in z_2 the speed of the robot is reduced. Thus, the dynamics in the two locations are $f_1 = x(t_k) + (\sin(\varphi)u_1(t_k), \cos(\varphi)u_1(t_k), u_2(t_k))^T \cdot \Delta t$ and $f_2 = x(t_k) + 0.5(\sin(\varphi)u_1(t_k), \cos(\varphi)u_1(t_k), u_2(t_k))^T \cdot \Delta t$ with a time-step Δt , a position vector x and a heading angle φ . The objective function for the MPC algorithm specifies the deviation from the trajectory y obtained from the planning layer: $J = \sum_{p=1}^H \|x(t_{k+p}) - y(t_{k+p})\|_2^2$ with the prediction horizon H . For the computation, a step size of $\Delta t = t_{k+1} - t_k = 0.1$ seconds and horizon of $H = 10$ is chosen. The forbidden regions are defined

as circles around the persons and the goal set follows from the trajectory y .

The solution of the optimization problem ϕ_u , is stored in the knowledge base together with the observed situation. The similarity measure of situations, needed to extract a previously calculated control input, is defined in terms of the Euclidean distances of the current state of the system and position of the persons.

Fig. 5 shows by the dotted line as outcome of the MPC algorithm the motion of the service robot from the kitchenette (1) to the table (2) and further to the refrigerator (3). The underlying map represents the kitchen in 2D and is used for the localization of the robot. For three time stamps t_1, t_2 , and t_3 the positions of the robot and the two persons are marked by a circle, and by a square / triangle respectively. The dash-dotted lines mark the motion of the two persons.

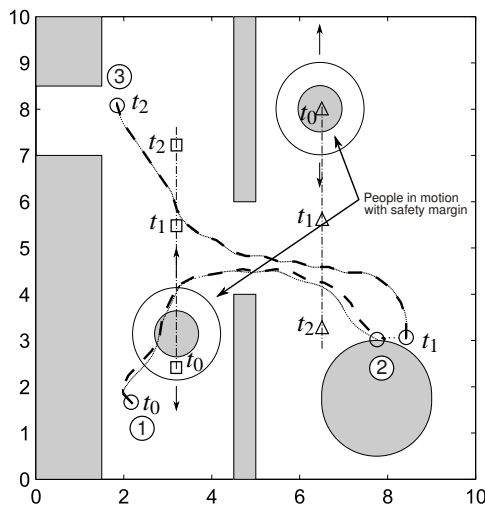


Figure 5: Trajectories of the robot moving between kitchenette (1), table (2), and refrigerator (3).

For each varying goal positions, the knowledge base is filled with information on goal-attaining strategies for future use. Employing the knowledge base reduces the calculation time essentially. The service scenario is simulated in the *Player/Stage/Gazebo* (PSG) environment.

7 CONCLUSIONS AND OUTLOOK

The aim of the presented concept is to provide an architecture by which the combination of control methods applied to hybrid systems, techniques from artificial intelligence, and vertical decomposition of tasks is realized. Simultaneously, the approach intends to

cover the whole perception-reasoning-action loop of an autonomous system that has to make decisions for goal-oriented behavior in new situations. While the considered example is quite simple, it illustrates the mechanism of task decomposition and improves efficiency of computing suitable control trajectories by the use of machine learning techniques.

Current work is focussed on extending the planning hierarchy to different learning and control algorithms.

ACKNOWLEDGEMENTS

Partial financial support of this investigation by the DFG within the Cluster of Excellence *Cognition for Technical Systems* is gratefully acknowledged.

REFERENCES

- Barto, A. G. and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(1-2):41–77.
- Broy, M. and Stoelen, K. (2001). *Specification and Development of Interactive Systems: Focus on Streams, Interfaces, and Refinement*. Springer.
- Galindo, C., Fernandez-Madrigal, J., and Jesus, A. G. (2007). *Multiple abstraction hierarchies for mobile robot operation in large environments*. Studies in Computational Intelligence, Springer.
- Haberl, W., Tautschnig, M., and Baumgarten, U. (2008). Running COLA on Embedded Systems. In *Proceedings of The International MultiConference of Engineers and Computer Scientists 2008*.
- Henzinger, T. (1996). The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS '96)*, pages 278–292.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press.
- Morari, M., Garcia, C., and Prett, D. M. (1989). Model predictive control: Theory and practice-A survey. *Automatica*, 25(3):335–348.
- Nau, D. S., Smith, S. J. J., and Erol, K. (1998). Control strategies in HTN planning: Theory versus practice. In *AAAI/IAAI*, pages 1127–1133.
- Stursberg, O. (2006). Supervisory control of hybrid systems based on model abstraction and refinement. *Journal on Nonlinear Analysis*, 65(6):1168–1187.
- Takadama, K. and Fujita, H. (2004). Lessons learned from comparison between q-learning and sarsa agents in bargaining game. In *Conference of North American Association for Computational Social and Organizational Science*, pages 159–172.

LICENSE PLATE NUMBER RECOGNITION

New Heuristics and a Comparative Study of Classifiers

César García-Osorio, José-Francisco Díez-Pastor, Juan J. Rodríguez and Jesús Maudes
Higher Polytechnic School, University of Burgos, Avda. Cantabria, Burgos, Spain
cgosorio@ubu.es, jdp0014@alu.ubu.es, jjrodriguez@ubu.es, jmaudes@ubu.es

Keywords: Computer Vision, Digital Image Processing, Optical Character Recognition, Smoothing Spatial Filter.

Abstract: We describe an artificial vision system used to recognize the Spanish car license plate numbers in raster images. The algorithm is designed to be independent of the distance from the car to the camera, the size of the plate number, the inclination and the light conditions. In the preprocessing steps, the algorithm takes a raster image as input and gives an ordered list of license plate areas candidates. Only in very rare occasions the second and third candidates are needed; most of the cases the first candidate is the correct one. During the preprocessing a new filter has been used. This filter is only applied to low saturation areas, getting better results this way. In order to choose the best classifier for the classifying stage, a comparative study has been performed using the data mining tool Weka.

1 INTRODUCTION

License plate number recognition or license plate recognition (LPR) is one of the most practical application of image processing and pattern recognition techniques. Its most common use is the control of parking areas (Sirithinaphong and Chamnongthai, 1999) and traffic monitoring (Setchell, 1997). In these contexts the distance to the camera, position, inclination and other variables that affect the image are quite well controlled and hence it is possible to take advantage of these restricted working conditions (Emiris and Koulouriotis, 2001). The application presented in this paper is designed to work in not so restricted and structured environments. We want to locate the license plate and recognize its number independently of its size, orientation, position or lightening condition. The only restriction we assume is that the background of the license plate number is white and the characters color is black. This is the common style for the Spanish license plates which we want to recognize.

The process of license plate number recognition involve three main steps: i) license plate localization, ii) character segmentation, and iii) character recognition. We have structured the rest of the paper in accordance with these steps. However, we will give an introduction to the Spanish licence plate number system first.



1.1 Spanish License Plates Formats

The increase in the number of cars has put to the limits the license plate number systems. So, plate number systems have needed to be adapted as the number of cars approached the maximum number these systems could register. In Spain, initially, the plate number were composed by one or two letters to code the province followed by up to six digits. This system lasted till 1971 when the plate numbers in Madrid (Urios-Mondéjar, 2007), the capital of Spain, were approaching the number 999999. In the new system there were four digits and a letter in a first moment, and later was necessary to use two letters. With the joining of Spain to the European union a new change happened.

In the year 2000, and close again to the coding limit of the system, a new change is made. In the new system a plate number is composed by four digits followed by three letters. Besides, the province code disappeared and an European Union logo with blue background and a 'E' in white¹ appeared in the left side of the plate number. As well, whenever a plate number need to be substituted because it is deteriorated, the new plate number will come with the European Union logo, even if the plate number is in the old format with the province code.

¹ 'E' for "*España*", "Spain" in Spanish

Table 1: Common Spanish plate number formats

With province code and no letter	V· 298176
With province code and letters	V·1257·HJ
With province code, letters and European Union logo	 BU 7893 T
Current plate number code	 0802 FXX

In Table 1 it is possible to see all these formats². These and others facts about Spanish plate numbers can be found in the webpage <http://www.geocities.com/amoros29/>.

2 LICENSE PLATE LOCALIZATION

2.1 Image Preprocessing

The first stage of the process is to locate the licence plate area in the input photo. In order to do so, we need to emphasize regions of high spatial frequency that correspond to edges. Therefore, we first use the classic Sobel filter. As the input to this step of the process is an RGB color image, and we know that in the license plate the background is white and the characters are black, we will apply the horizontal Sobel filter to the areas of low saturation. The saturation is obtained from RGB color image using the formula

$$S = \begin{cases} 0 & \text{if } \max\{R,G,B\}=0, \\ \frac{\max\{R,G,B\}-\min\{R,G,B\}}{\max\{R,G,B\}} & \text{otherwise.} \end{cases} \quad (1)$$

All the achromatic colors (grey scale colors) have the same saturation and lower than any chromatic color. Note that the white could be actually gray depending on illumination conditions. The black and white pixels of the license plate areas will have low saturation. So, we discard all the pixels with high saturation and apply the Sobel filter only to the pixels with low saturation. Experimentally, we found that the best results are obtained when applying the Sobel filter to pixels with saturation lower than 0.35. In Figure 1 we can see the results of using this heuristic.

²In the Spanish plate number system there are two rows plate numbers too, as well as other special plate numbers, that has not been taken into account in the applications explained in this paper.



Figure 1: Top left: Original RGB image. Bottom left: After applying traditional Sobel filter. Top right: the image without the low saturation areas. Bottom right: After applying the conditional Sobel filter.

Another heuristic we use here is to discard the 30% top part of the image and the 10% of the bottom part. This areas never have a license plate number (Rodríguez, 1999).

Next in the process is the binarization. We obtain a black and white image from the result of the previous transformation. The binarization threshold is chosen to get a black and white image with only 2% of the pixels being white, this way only those pixels with high intensity variation are considered as edges in the license plate. The edges due to background, or without that high intensity variation are discarded.

After all the previous preprocessing, the license plate will appear as a sequence of white and black transitions with a lot of white pixels concentrated in the license plate area. In other parts of the image the concentration of white pixels is lower. As the last step of this stage we use an horizontal soften filter. This filter can be thought as a kind of mean or average filter. The idea is to count the number of white pixel in a rectangular area with center at each pixel. Empirically, we found the best results were obtained with a size 3x61. If the number of white pixels is greater than 50 we leave that pixel white. That is, $M(x,y) = 1$ (1=white), if $\sum_{i=-30}^{30} \sum_{j=-1}^1 M(x+i,y+j) > 50$.

When we apply this filter to a line with a lot of low concentrated edge pixels, all those pixels are eliminated. On the contrary, when we apply this filter to a line with less edge pixels but high concentrated, the line will became white.

One could think that the effect of applying this filter is the same as the dilation/erosion of mathematical morphology (Serra, 1982), but this filter seems to be less sensitive to noise. In Figure 3 we can see the results of dilation/erosion with different window sizes.

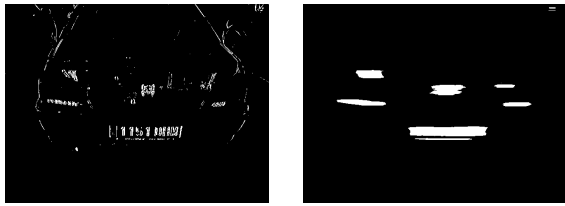


Figure 2: Left: Image after binarization. Right: Image after applying the horizontal soften filter.

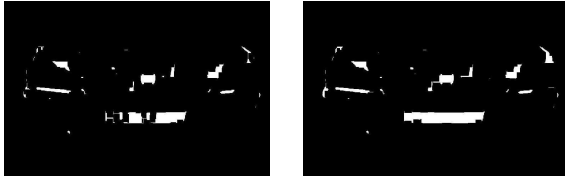


Figure 3: Left: Image after dilation/erosion with a window 15 pixels width. Right: Image after dilation/erosion with a window 20 pixels width.

2.2 Selection of Plate Candidate

After applying the last step of the previous stage, we have a black image with one or more white areas candidate to be a license plate. However, only one of those areas corresponds to the real license plate, while the others are consequence of noise or other objects in the background. Now, we need an algorithm to determine which one of these white areas is the real plate.

We have used several heuristics. First, we only keep the ten biggest areas, this way we eliminate most of the areas due to noise. We then order those ten plate candidates according to the following criteria

- ratio of width to heights greater than two, since a license plate is wider than taller.
- distance to the center of the image, the areas far from the center are unlikely to be plates.

For two areas with similar characteristics we choose the one with lower positions, we do this because we have found that some times the area due to the edges in the car radiator will give end as a false positive plate.

Note, that we keep all the ten areas not discarded. Later, if in the following stages we realize that the selection of plate area was incorrect we recover the following plate candidate from the ordered list. In the experiment this only happened in few occasions.

3 CHARACTER SEGMENTATION

After the previous stage, we obtained a license plate image. We need to do further image processing before



Figure 4: Top left: Original RGB image. Bottom left: After applying traditional Sobel filter. Top right: After binarization. Bottom right: After rotation.

starting to locate the characters in the plate. We have not tried any new heuristic with these steps. We apply the classic Sobel filter (Gonzalez and Woods, 2002), gradient binarization (Rodríguez, 1999) and Hough transform (Gonzalez and Woods, 2002) to correct the plate inclination if needed (see Figure 4).

3.1 Character Localization

Now with the preprocessed plate we use a novel method to locate the characters in the plate. That is one of the most delicate steps of the process.

First, we try to adjust further the license plate and we eliminate from the border of the image all the rows and columns with only white or only black pixels.

Second, we apply a coarse method to discard those blobs of black pixels which are obviously consequence of noise and are not really characters. This process must be conservative, because we do not want to discard correct characters. We have follow several heuristics:

- We discard all the areas whose width is greater than 1/8 of the width of the area we have identified as a license plate candidate. As the license plate will have a minimum of seven characters, an area with bigger width will be for sure not a character.
- We discard as well the areas which are taller than 0.7 the height of the plate.
- We use the color information to eliminate the areas with at least 50% of high saturation pixels. This areas correspond to the blue European Union logo. This have the effect of eliminate other chromatic areas in case our location of the license plate would have not adjust close enough to the plate.
- If the width of the widest area is lower than 1/20 of the width of the area we have identify as the license plate, we have found a case of false positive. We need to start the process with the next candidate from the list of plate candidates.

Next, we apply a more refined strategy. A character will be a blob of connected black pixels (or several, if the characters parts have been disconnected as a consequence of the previous preprocessing) fully inside of a rectangular area of dimensions calculate as a



Figure 5: License plate before and after eliminating the black areas that does not correspond with characters.

percentage of the dimensions of the license plate area. We align in turn the top left corner of this rectangle with the top left corner of all the blobs, starting with the one in the very top left. Be β_1 the blob we have used to align the rectangle. It will not be the case that β_1 does not fully fix inside the rectangle, because all that big blobs have been discarded with the previous coarse method. Now, if there is piece of another blob β_2 that is partially inside the rectangle, but not fully inside, we can discard β_1 . It will not be a character or part of a character, β_1 is probably noise. If there are not others blobs partially inside the rectangle, and there is another blob β_2 fully inside the rectangle, we can consider both β_1 and β_2 as parts of the same character. This method is able to group pieces of characters that have been disconnected after applying the filters, and at the same time discards any remaining noise.

In Figure 6 we can see an example of the method. The coarse method was not able to eliminate all the noise. The plate border is broken and was not big enough as to be discarded. Besides the “U” appears broken in two pieces. In the proposed method the rectangle will be aligned with the top left pixel of the first piece, and as the other piece is fully inside the rectangle both pieces are identified as part of the same character. On the contrary, the small blob due to the number plate screw is not consider part of the character because when we align the rectangle with its top left pixel, the six is only partially inside the rectangle. The same happens in the right side, when we align the rectangle with a border, and the “Z” is only partially inside the rectangle.

We have found this method slightly better than the traditional horizontal projection method for character segmentation and less sensible to broken characters. Still, in same rare circunstantes we could group some scattered points and identify them as parts of characters. But this cases could be easily managed using as criteria the number of black pixels.

3.2 Character Normalization

Before using a classifier to identify the characters we need to normalize them. There are three reasons for normalization:

- We can reduce the number or attributes and that will make the learning of the classifier faster.

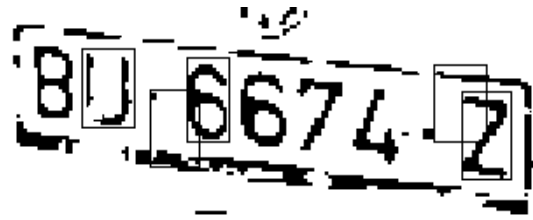


Figure 6: Character segmentation.

- We will have the same number of attributes for all the characters because they will have the same dimensions.
- We will improve the efficiency of the classifier as the input is more similar to the others instances in its class, and more different from the instances in other classes.

The normalization steps are:

- Center the character.
- Resize it to 16×16 keeping its ratio width/height (this is specially important for characters like “I”).

4 CHARACTER RECOGNITION

In this last section, we analyze and compare which set of characteristics and classifiers give the best classification results. To do this study we use Weka (Witten and Frank, 2002), a data mining tool written in Java <http://www.cs.waikato.ac.nz/ml/weka/>. We evaluate the performance of seven different classifiers and four different sets of characteristics using 2070 characters from about 700 licence plates.

We use 10×10 -fold cross validation, in each of the experiments 90% of the instances were used for training and 10% for validation, and this was repeated another nine times, with different 10% of instances each.

An important characteristic we could extract from the character image is the number of holes in the characters. That increase the classification accuracy for characters like “P” and “F” or “B”, “8”, “6” and “9”. The sets of characteristics use in this study were:

- In this data set we represent each of the 2070 characters using the whole $16 \times 16 = 256$ pixels of the character image together with the number of holes: 257 attributes in total. This is basically the character without any characteristic extraction process.
- We use the horizontal and vertical projections of the character. This gives 16 characteristics from the horizontal projections, 16 characteristics from

the vertical projections, plus the number of holes: 33 characteristics.

- c. In this data set each character is represented by its main four Kirsch compass filters, horizontal, vertical, and both diagonals. The results are down sampled to a 4×4 image, and the original image is down samples as well. This gives $5 \times 16 = 80$ characteristics, plus the additional one for the number of holes: 81 characteristics.
- d. We use a overlapped down sample of the original image. Each 4×4 block of pixels gives a value, then this mask is moved two pixels right, and so till the right side, then the mask is lower two pixels and the process repeated from the left. We obtain 49 attributes, plus the number of holes: 50 characteristics.

Regarding the classifiers, we tried the following:

1. functions.MultilayerPerceptron, with parameters: GUI=false, autoBuild=true, Debug=false, Decay=false, HidenLayers=(number of attributes + number of classes)/2, LearningRate=0.3, momentum=0.2, nominalToBinaryFilter=true, normalizeAtributes=true, normalizeNumericClass=true, RandomSeed=0, trainingTime=500, validationSetSize=0 and validationThreshold=20. Neural Networks has been used as classifiers in (Draghici, 1997; Nijhuis et al., 1995)
2. bayes.NaiveBayesUpdateable, with options: debug=false, useKernelStimator=false and useSupervisedDiscretization=false.
3. functions.SMO, Platt's sequential minimal optimization algorithms for training Support Vector Machines (Platt, 1999; Zheng and He, 2006; Chengwen et al., 2006), with options: buildLogisticModels=false, C=1.0, checksTurnedOff=false, Epsilon=1.0E-12, filterType=Normalize training data, Kernel=PolyKernel, RamdomSeed=1 and toleranceParameter=0.0010.
4. lazy.IBk, with no distance weigthing and options: KNN=1, crossValidate=false, Debug=false, MeanSquared=false, NearestNeighbourSearchAlgorithm=LinearNN, WindowsSize=0.
5. meta.AdaBoostM1, with options: classifier=J48, Debug=false, NumIterations=10, useResampling=false, WeightThreshold=100.
6. meta.Bagging, with options: classifier=J48, Debug=false, NumIterations=10, bagSizePercent=100, calcOutOfBag=false, Seed=1.
7. trees.J48, with options: binarySplits=false, ConfidenceFactor=0.25, debug=false, MinNumObj=2, numFolds=3, reducedErrorPruning=false,

Table 2: Results comparison

	a.	b.	c.	d.
1	99.10	97.88	98.15	99.03
2	<u>96.42</u>	<u>89.44</u>	<u>89.86</u>	<u>93.65</u>
3	99.19	<u>96.85</u>	98.01	99.09
4	<u>97.86</u>	<u>96.38</u>	<u>96.45</u>	<u>97.74</u>
5	<u>97.56</u>	<u>95.39</u>	<u>96.23</u>	<u>96.81</u>
6	<u>95.46</u>	<u>92.92</u>	<u>93.87</u>	<u>95.94</u>
7	<u>94.04</u>	<u>89.17</u>	<u>91.04</u>	<u>93.66</u>

SavaInstanceData=false, Seed=1, subtreeRaising=true, Umpruned=false, UseLaplaze=false.

The results of the experiments are shown in Table 2. The underline values indicate a significant worse accuracy compares with the multi layer perceptron that has been used as the base of the comparison. We can see that all the classifiers have an accuracy higher than 90% for all the data sets, what shows the benefits of all the preprocess and normalization work. Note, thought, that the best results are obtained when the character is used without any characteristic extraction (a). The worse set of characteristics are the horizontal and vertical projections (b).

Regarding the classifiers, as we could expect, AdaBoost and Bagging improve the performance of the J48 that they are using as base learner, being AdaBoost better than Bagging. The best classifiers is the SMO using directly the binary matrix that represent the character. However, finally we decide to use the multi layer perceptron, it is only a bit slower but is faster and its memory requirements are more than three times lower.

5 CONCLUSIONS

We have described an artificial vision system used to recognize the Spanish cars license plate numbers in raster images. We combine the use classic image processing techniques with some new ideas, such as the soften filter and the character segmentation method.

In the study of classifiers we confirmed the benefits of the preprocessing stages achieving accuracies above 90% for different sets of characteristics. For two of the classifiers we increase the accuracy to 99%.

In future version, we expect to take into account the two row Spanish license plates, and the special license plates with different combination of background and foreground colors.

As well, as one of the reviewers suggested, we want to prepare a more detailed study of the use of

the soften filter, specially regarding its robustness to noise in comparison with other filter in the literature.

ACKNOWLEDGEMENTS

This work has been possible thanks to the project BU004B06 from the “Consejería de Educación de la Junta de Castilla y León”, Spain.

REFERENCES

- Chengwen, H., Yannan, Z., Jiaxin, W., and Zehong, Y. (2006). An improved method for the character recognition based on svm. In *24th IASTED International Conference on Artificial Intelligence and Applications*, pages 457–461, Anaheim, CA, USA. ACTA Press.
- Draghici, S. (1997). A neural network based artificial vision system for license plate recognition. In *International Journal of Neural Systems*, volume 8, pages 113–126.
- Emiris, D. E. and Koulouriotis, D. E. (2001). Automated optic recognition of alphanumeric content in car license plates in a semi-structured environment. In *International Conference on Image Processing, ICIP*, volume 3, pages 50–53.
- Gonzalez and Woods (2002). *Digital Image Processing*. Addison-Wesley, Reading, MA, USA.
- Nijhuis, J., Brugge, M., Helmholt, K., Pluim, J., Spaanenburg, L., Venema, R., and Westenberg, M. (1995). Car license plate recognition with neural networks and fuzzy logic. In *IEEE International Conference on Neural Networks*, volume 5, pages 2232–2236. ACTA Press.
- Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. In Scholkopf, B., Burges, C., and Smola, A., editors, *Advances in kernel methods: support vector learning*, pages 185–208. MIT Press.
- Rodríguez, F. M. (1999). *Contribución al reconocimiento de caracteres en imágenes complejas*. PhD thesis, Escuela Técnica Superior de Ingenieros de Telecomunicaciones, Campus Universitario, 36310, Vigo, Galicia, Spain.
- Serra, J. (1982). *Image analysis and mathematical morphology*. Academic Press, London, 2nd edition.
- Setchell, C. J. (1997). *Applications of Computer Vision to Road-traffic Monitoring*. PhD thesis, Faculty of Engineering, Department of Electrical and Electronic Engineering.
- Sirithinaphong, T. and Chamnongthai, K. (1999). The recognition of car license plate for automatic parking system. In *Fifth International Symposium on Signal Processing and Its Applications, ISSPA*, volume 1, pages 455–457.
- Urios-Mondéjar, D. (2007). Sitio de las matrículas españolas. <http://www.geocities.com/amoros29/>. (last visited: 2-Dec-2007).
- Witten, I. H. and Frank, E. (2002). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman, 500 Sansome Street, Suite 400, San Francisco, CA 94111, 2nd edition.
- Zheng, L. and He, X. (2006). Number plate recognition based on support vector machines. In *IEEE International Conference on Video and Signal Based Surveillance (AVSS'06)*, page 13, Washington, DC, USA. IEEE Computer Society.

OBJECT EXPLORATION WITH A HUMANOID ROBOT

Using Tactile and Kinesthetic Feedback

Nicolas Gorges, Stefan Gaa and Heinz Wörn

Institute for Process Control and Robotics, University of Karlsruhe, Engler-Bunte-Ring 8, Karlsruhe, Germany
{gorges, gaa, woern}@ira.uka.de

Keywords: Tactile Sensing, Object Exploration, Robot Control, and Humanoid Robot.

Abstract: This work deals with the reactive and autonomous exploration of objects with a humanoid robot using only tactile and kinesthetic sensor feedback. To coordinate the flow of the exploration, a novel hierarchical exploration system is introduced. The lowest level extracts contacts points and elementary features based on the direct contact with the object. It furthermore provides elementary movement primitives. The intermediate level consists of different controlling behaviors to generate exploration movements according to the sensor feedback. This level enables the robot to explore an object pointwisely or continuously. The highest level evaluates the process of the exploration and determinates the reactive behavior of the underlying components. The evaluation scenario comprises the exploration of edges, which are arbitrarily located in space. The evaluation platform consists of a robot arm, a force-torque sensor, and a tactile sensor matrix. The proposed approach is evaluated and the different reactive behaviors as well as the used sensor modalities are compared.

1 INTRODUCTION

Service robots build a new block of research area. Their range of application highly differ from conventional industrial robot, as they are rather deployed in a domestic environment and are usually not intended for repetitive tasks with need for high precision or high forces. Typical tasks for a humanoid robot are all kinds of fetch and carry or manipulation tasks with a broad variety of objects. Not all of these object can be assumed to be known appriori. The robot might encounter unknown or partially unknown objects. Therefore, the robot has to use its sensors, like visual or haptic sensors, to explore an unknown object. The visual exploration is adequate to determine the location and the rough shape of an object but it is limited due to the ambiguity of visual data and the need for textured objects as well as good light conditions.

The haptic exploration describes the active palpation of objects. The relevant parts of the haptic exploration are the tactile perception (surface sensibility) and the kinesthetic perception (depth sensibility). A haptic exploration procedure of an object requires the direct interaction of the robot with the object. On the one hand, this delivers accurate 3D information about the object but, on the other hand, requires a reactive

control strategy to bring the sensors into the right position.

Previous work in (Klatzky et al., 1987) (Lederman and Klatzky, 1987) identified the following exploration procedures from observing human exploration behavior: lateral motion, pressure, enclosure, and contour following.

These procedures allows to determine the texture, hardness, shape and size of an object by haptic exploration. The exploration procedure of this work primarily focuses on the procedure of contour following. Here, the tactile perception is represented by a tactile sensor matrix whereas the kinesthetic perception is given by a force-torque sensor. Robotic exploration procedures so far include approaches with grippers (Schmidt et al., 2006) and with tactile sensor matrices (Chen et al., 1995), (Heidemann and Schoepfer, 2004).

A major problem for such exploration tasks is the detection of a contact with the object to be explored. The limitation is usually given by the sensibility and the spatial resolution of the sensor. Therefore, many approaches using tactile sensors assume that the sensor is already located directly at the region of interest.

In order to avoid this restriction, this work introduces a novel approach by coupling the information of a force-torque sensor in the wrist and a tactile sen-

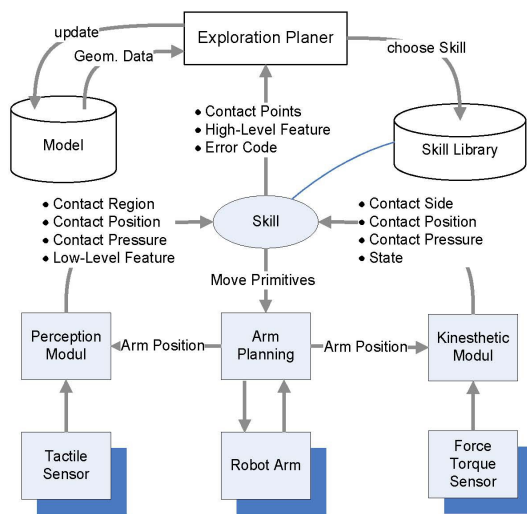


Figure 1: A system overview.

tor matrix representing the pose of an opened hand. This enables the robot on the one hand to detect direct contacts using the tactile sensor matrix, which are very accurate but less sensitive, and on the other hand to detect indirect contacts using the force-torque sensor and a model of the hand, which make it less accurate but very sensitive.

The remainder of this paper is structured as follows: After this introduction, the system overview of this approach is discussed in section 2 followed by the perception level in section 3. Our results are presented in section 4 before conclusions are given in section 5.

2 SYSTEM OVERVIEW

In order to consecutively determine the shape of an object, a concept consisting of three components is introduced, as shown in figure 1. Here, the highest level is represented by the exploration planner. A reactive control layer complies with the skill library, from which consecutively a skill is chosen and executed. The haptic layer builds the lowest block of the framework and is realized by three components: tactile perception, arm planning and kinesthetic perception. The planning layer and the reactive control layer are explained in the following. The haptic layer is presented separately in section 3.

2.1 Exploration Planner

The exploration planner makes decisions based on the geometric shape of an object which can be determined by a successive palpation sequence. Based on the estimate of the object's shape, it determines the align-

ment of the robot towards the object. The planner executes a sequence of skills whereas the skills provides the planner with contact features during their execution. The planner evaluates the alignment towards the object and chooses a different skill if necessary.

2.2 Exploration Skills

In this work, the exploration skills represent strategies to explore structures of an object. They are elementary operations for the exploration procedure and are executed by the superior global exploration component. A sequence of simple exploration skills leads to a complex exploration behavior.

Their key tasks are the coordination of the arm control and the creation of a defined exploration behaviors according to the preprocessed data. A skill has to fulfill two objectives. First of all, it has to bring the robot arm into the right position which is realized by a sequence of states. Secondly, it has to provide the planning level with an amount of contact points. To fulfill these tasks, a skill can access the underlying haptic components. The planning level supervises the execution of a skill and interprets the features provided by a skill in a global context.

A single skill is represented by a state machine which determines the flow of action. Such a skill is able e.g. to follow a structure or to rotate around a structure according to its task definition. The most states imply a coordination of sensor data and movements but also include instructions from the upper level. In general, the skills can be separated into two groups:

- Skills with discrete movements
- Skills with continuous movements

These two types of skills are explained in more detail in the following.

2.2.1 Skills with Discrete Movements

Skills with discrete movements are characterized by a palpation sequence as the skill departs from the object after establishing a contact before moving the arm to next position of interest. This is achieved by movement and control primitives which are the building blocks of the total movement. Movement primitives enable the robot to move with respect to a local or a global coordinate system. Control primitives also embed sensor data during a movement, e.g. to buildup a certain amount of pressure. Here, the communication between a skill and the planner is distinctive. The benefit is that structures can be explored systematically whereas the drawback is that they can only perform an alignment according to point contacts. They need

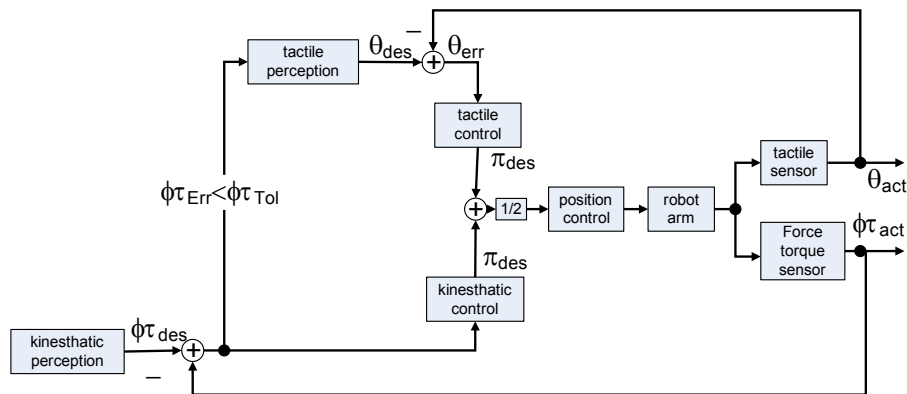


Figure 2: A control loop for skills with continuous movements

at least two contact points to perform an alignment and also depend on the measuring accuracy of these contact points.

The process of tracking an edge, for example, can be summarized as a sequence of the control and movement primitives “move up”, “move to the side”, “move down”, and “detect contact”. A sequence of this skill results into a simple but effective exploration procedure.

2.2.2 Skills with Continuous Movements

The exploration behavior of skills with continuous movements is given by a complex control loop, e.g. zero-force control or a tactile control. At first, these kind of skills establish a contact with the object surface. Then, using a superior control loop, they try to keep a steady contact towards the surface. Figure 2 shows the basic structure of such a superior control loop. It shows a cascaded system with the force control preceding the tactile control. It is characterized by a coarse control based on the measured forces and torques of the force-torque-sensor and by a fine control based on the measured pressure profile.

The idea is to use the force-torque sensor to keep the applied pressure $\phi\tau_{des}$ stable. On the one hand, the applied pressure should not exceed a given limit as it could damage the robot or the object. On the other hand, the pressure should not be too little as the tactile sensor matrix would not be able to measure an adequate pressure profile anymore and the sensor pad could even loose contact to the object surface. The tactile control loop is activated as soon as the applied pressure is within an acceptable interval, expressed by $\phi\tau_{err} < \phi\tau_{tol}$. The tactile control loop is defined by the deviation θ_{err} of the center of the tactile image from center of the sensor pad. The position corrections τ_{des} are given to the position control of the robot. This

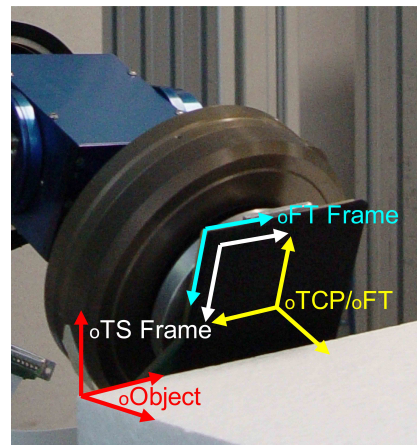


Figure 3: Setup of the evaluation platform and the involved coordinate systems.

enables the robot arm to align to the object and follow its surface.

3 PERCEPTION

The perception level is part of the lowest level. It preprocesses the data provided by the tactile sensor and the force-torque sensor. The perception in this work is therefore split into the tactile and kinesthetic perception. In the course of the paper, the tactile sensor matrix refers to the tactile perception whereas the force-torque sensor represents the kinesthetic perception. Figure 3 show the setup of the evaluation platform. It shows the tactile sensor and the force-torque sensor mounted to the robot arm. Furthermore, it illustrates the different coordinate systems which are involved in the perception layer.

The tactile sensor identifies very accurately the position of contacts caused by a direct touch. But

a small change in the orientation of the tactile sensor matrix causes a significant change in the resulting tactile image. Imagine an imprint of an edge on the tactile sensor, a small twist along the perpendicular image axis of the edge, makes the edge disappear until only a single point contact is determined. After a certain angle between the tactile sensor and the surface, no contacts can be detected as the sensor only turns normal forces into a pressure profile. This angle is called the **critical angle**.

The force-moment sensor, on the other hand, is not able to determine contacts in the inner regions of the sensor pad's surface. It can only detect the direction from the center point of the sensor to the border of the surface.

There is a simple decision rule for the determination of a contact, combining the tactile and the force-moment-sensor:

1. If the tactile sensor detects a contact, the contact must be inside the surface of the sensor pad. This is called a **TS-contact** and is the desired type of contact.
2. If the tactile sensor does not detect a contact but the kinesthetic perception does, the contact must be at the border of the sensor pad. We call this a **FTS-contact**.

It is obvious that the tactile sensor can only take a pressure profile if the robot apply enough force towards the direction of the object surface. Therefore, the more sensitive force-torque sensor is used to regulate the pressure applied to the object.

3.1 Tactile Perception

3.1.1 Working Principle

As already investigated and published by Weiss et al. (Weiss and Woern, 2005), the working principle of the tactile sensors depends on an interface effect between the metal electrodes and the structured conductive polymer covering the sensing electrodes. The resistance between the common electrode and a sensor cell electrode is a function of the applied load and time. This technique leads to very accurate pictures of the applied pressure profile and minimizes crosstalk between the sensor cells as well.

As each sensor cell represents a measured voltage, the voltage image has to be transferred to a pressure image. This characteristic curve of the tactile sensor can be obtained by calibration.

3.1.2 Tactile Feature Extraction

Identifying the characteristic features of an image using moments is a well known paradigm in image processing. The data of the tactile sensor matrix corresponds to a two-dimensional planar image. We analyze this image using moments up to the 2^{nd} order (Hu, 1962). The two-dimensional $(p+q)^{th}$ order moment $m_{p,q}$ of an image is defined as the following double sum over all image pixels (x,y) and their values $f(x,y)$:

$$m_{p,q} = \sum_x \sum_y x^p y^q f(x,y) \quad p, q \geq 0 \quad (1)$$

The moment $m_{0,0}$ constitutes the resulting force exerted on the sensor. The center of gravity $\underline{x}_c = (x_c, y_c)^T$ of this force can be computed to

$$x_c = \frac{m_{1,0}}{m_{0,0}} \quad (2)$$

$$y_c = \frac{m_{0,1}}{m_{0,0}} \quad (3)$$

Using the center of gravity, we can verify that the object surface is aligned to the center of the sensor pad. It also allows to calculate the higher order moments with respect to the center of gravity, the so-called *central moments* $\mu_{p,q}$:

$$\mu_{p,q} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q f(x,y) \quad p, q \geq 0 \quad (4)$$

The 2^{nd} order central moments

$$\mu_{2,0} = \sum_x \sum_y (x - x_c)^2 f(x,y) \quad (5)$$

$$\mu_{0,2} = \sum_x \sum_y (y - y_c)^2 f(x,y) \quad (6)$$

$$\mu_{1,1} = \sum_x \sum_y (x - x_c)(y - y_c) f(x,y) \quad (7)$$

approximate the image by an ellipse and represent its principal axes. The eccentricity of a contact is described by the relation of the eigenvalues λ_1 and λ_2 . If both eigenvalues have a similar value, then the contact area has a round shape and the eccentricity is close to zero. For these contacts it is not possible to calculate the orientation.

Touching an edge results in an oblong ellipse with an eccentricity ϵ close to 1 when using

$$\epsilon = \frac{(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2}{(\mu_{2,0} + \mu_{0,2})^2} \quad \epsilon \in [0, 1] \quad (8)$$

A corner point results to an eccentricity close to zero. To control the orientation of the sensor pad with respect to the object surface, we are interested in the

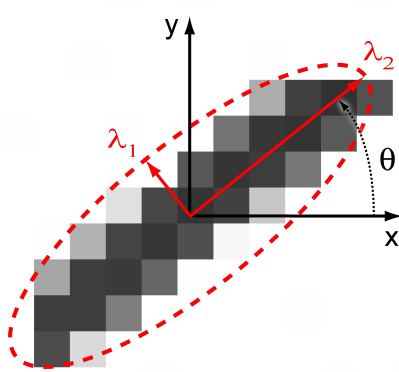


Figure 4: The angle θ between the principal axes of the tactile image and the sensor coordinate system.

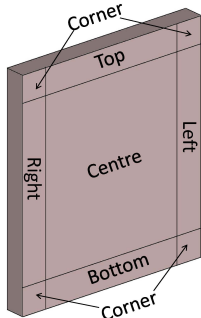


Figure 5: The 9 regions of the sensor pad.

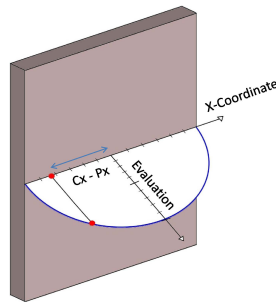


Figure 6: Evaluation of a contact point.

angle θ between the principal axes and the sensor coordinate system (cf. Fig. 4) which can be readily computed by

$$\theta = \frac{1}{2} \arctan \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}. \quad (9)$$

When tracking an edge, the desired angle θ is zero and can thus be directly used as the system deviation input to the controller to control one orientation DOF. The angle θ will be also referred to as the quality measure q_1 in the following.

To evaluate a contact point regarding to the orientation along the y-axis, we compute a quality measure from the distance $|y_c - y_p|$ of the contact point (x_c, y_c) to the center of the sensor pad called (x_p, y_p) . As the correlation between this distance and the angle is not linear, we weight the distance by a circular function. This circular function f with the radius r is given by

$$f_r(x) = \sqrt{r^2 - x^2} \quad (10)$$

This results into the function

$$q_2 = \sqrt{y_p^2 - |y_c - y_p|^2} \quad (11)$$

Figure 6 illustrates the computation of this quality measure which is used for the alignment of the sensor pad towards the object surface.

Furthermore, we divide the pad into 9 regions: 4 corner regions, 4 border regions and one interior region, as shown in figure 5. Each region is checked, if it accommodates a contact or not. This computation results into a 9-dimensional binary feature vector which can be used for a simple classification like the determination of corner contacts or contact side, depending on the present context.

3.2 Kinesthetic Perception

The force-torque sensor that we use is an FTC 50-40 from SCHUNK. It has 6DOFs, with a range of 150N for the forces, 4Nm for the torques M_x and M_y , and 8Nm for M_z . The accuracy is 5%. The data is sampled every 1ms and transmitted via CAN bus with a baudrate of 500kbit/s.

Since the data is quite noisy, the preprocessing of the sensor data includes a median filter with window size 7 to remove outliers. Since the tactile sensor pad is mounted on the top of the sensor, we must deduct its weight from the sensor values.

The compensation of the torques and forces is only possible in the global robot coordinate system and not in the local system. Local compensated forces are computed via back-transformation of global compensated forces into local coordinates. The required transformation steps are $f_l \rightarrow f_g \rightarrow f_{g,c} \rightarrow f_{l,c}$, where l and g mark local coordinated and global respectively. The marker c describes compensated values.

It is possible to determine contacts without the tactile sensor by relating the measured torques (m_x, m_y, m_z) with a model of the used sensorpad. The sensorpad can be described as a simple rectangle with the length (l_x, l_y) . The angle α from the center of the pad to the contact point can be calculated by $\text{atan}(m_x/m_y)$. The angle furthermore describes, if a contact is safe or not. A corner contact is declared as unsafe, as is it not accurate enough to identify the correct contact side.

4 RESULTS

At first, the measuring accuracy of the contact points is investigated using the tactile and the force-torque sensor. Then, the implementation of a skill with discrete movements for edge tracking is shown. This skill is used to evaluate the exploration with each sensor in a stand-alone application and with a combination of both sensors. Finally, a skill with continuous movements is shown and evaluated.

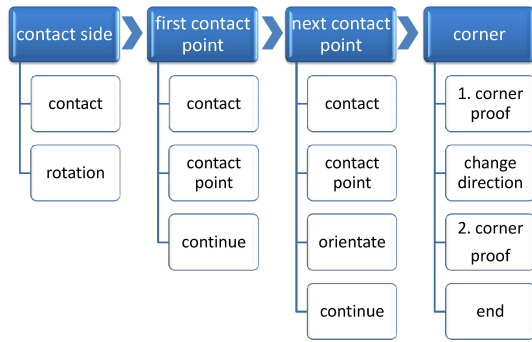


Figure 7: Implementation of a discrete skill.

4.1 Measuring Accuracy

To obtain the measuring accuracy of the tactile and the force-torque sensor, reference and measured points are generated consecutively along a line with a displacement of 1.0 cm in relation to the center of the tactile sensor pad. The measured contact points are determined by the interpolation of the contact area in the resulting tactile image. This simple experiment has shown that the used tactile sensor matrix with a cell distance of 0.6 cm has a mean square error of 0.2 cm. Furthermore, the accuracy is not correlated to the location of the contact point. A similar experiment has been done for testing the force-torque sensor. Here, the reference points were taken only at the boundary of the sensor pad. The measured mean square error is 0.6 cm.

The tactile sensor is superior to the force-torque sensor regarding the measuring accuracy. The tactile has one significant disadvantage - it exist a critical angle which restricts the operational area. Applying the maximum force of 8N towards a planar surface, the critical angle is 18 degrees. If the sensor pad is aligned with a larger angle towards the object, a contact cannot be detected anymore.

4.2 Discrete Skill for Edge Tracking

4.2.1 Implementation

The objective is a skill which pointwisely tracks an edge. Figure 7 shows a coarse view of the implementation of such a skill consisting of 4 main states and several sub-states. The skill tracks the contour of an edge and changes the direction of exploration as soon as the first corner has been detected. The first state involves the determination of the first contact and the contact side. Therefore, the robot arm moves into a specified direction until it detects a contact and departs again. According to this first contact, the orientation of the sensor pad is aligned so that the long side

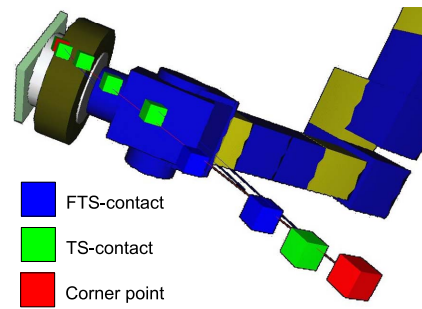


Figure 8: Visualisation of the exploration procedure.

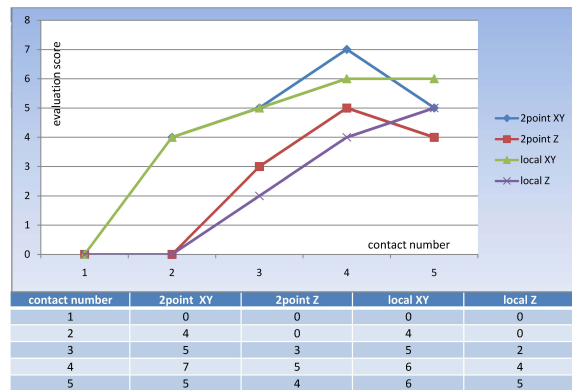


Figure 9: Evaluation using only force-torque sensor.

of the pad is used for tracking the edge. After this initial alignment, the first contact point is collected and the arm departs again. The first two steps are only executed once.

The third step is executed consecutively until a corner point and hence the end of the edge is detected. This step collects the next contact point, departs from the object, evaluates the current spatial alignment, and executes a correction of the alignment. If a corner is detected by the tactile perception, the fourth phase is triggered which involves the decision to change the direction of the exploration, if the first corner has been detected, or to stop the exploration if the final corner has been found. Figure 8 visualizes the outcome of such an exploration procedure: a set of points and lines.

4.2.2 Force-torque Sensor vs. Tactile Sensor

To compare both sensors, the implemented skill has been executed once only with the force-torque sensor and another time only with the tactile sensor matrix. In order to make the skill be executable for both sensors in a stand-alone application, several requirements must be considered. At first, the sensor pad was aligned with an angle of 10 degrees towards the object. Secondly, the exploration procedure only in-

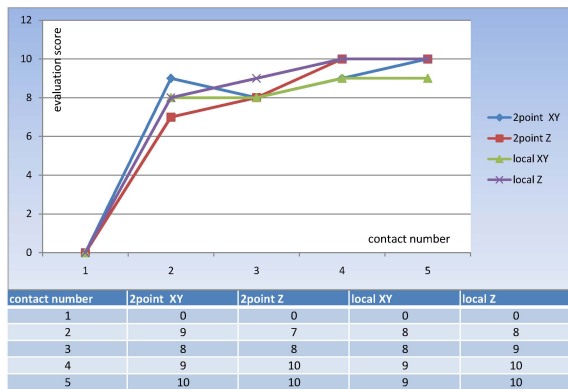


Figure 10: Evaluation using only tactile sensor.

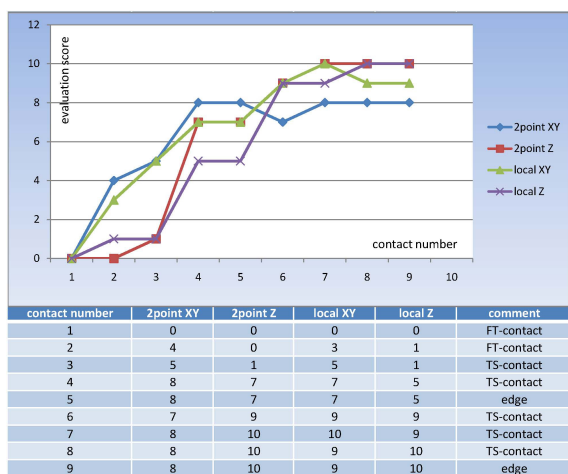


Figure 11: Evaluation using force-torque sensor and tactile sensor.

involves tracking the edge in one direction up to the first corner, as the force-torque is not able to detect the corner of an edge. After the second contact point, the alignment towards the edge is evaluated.

For tracking an edge two alignments are needed. At first, the imprint of the edge must be parallel to a specified boundary of the sensor pad. Secondly, the center of gravity of the contact point must be in the center of the sensor pad. For these two alignments, two quality measures were introduced in section 3. For the sake of convenience, these two quality measures will be scaled to a score from zero to ten whereas ten indicates a good score. The score *XY* refers to the difference between the center of the contact point and the center of the sensor pad, as described in equation 11. The second score *Z* refers to the deviation of the angle of the edge, as stated in equation 9. The prefix *2-Point* indicates that only the last two contact points are used for an estimate of the edge whereas the prefix *Local* points out that all extracted point so far are taken into account.

Both experiments have been repeated several times. Representative results of both experiments are shown in figures 9 and 10. The diagrams plot four curves which result from two quality measures for the current alignment, labelled *2-Point-XY* and *2-Point-Z* and two quality measures for the alignment over several exploration steps, labelled *Local-XY* and *Local-Z*.

As to be expected, the tactile sensor scores well with an average score of about 9 points for both quality measures. The results of the force-torque sensor is worse with an average score of 5 points. Furthermore, the tactile edge tracking converges faster towards the optimum of the quality measures. Both sensors enable the robot to explore an edge considering some restrictions but both sensors also complement each other. The FTS-determination of contacts points is less accurate but independent from the edge angle. It allows the alignment of the sensor pad so that the critical angle of 18 degrees is under-run and the tactile sensor can take over the exploration procedure in order to undertake a more precise computation of the edge.

4.2.3 Combination of Both Sensor

Finally, the edge tracking with a combination of both sensor is evaluated. The initial angle is 25 degrees which exceeds the critical angle. Figure 11 shows the result of the exploration procedure. The additional comment shows one of the three possible states of the contact point: FTS-contact, TS-contact, and corner point. As predicted, the orientation of the edge can be calculated only by the use of the force-torque sensor so that the tactile sensor can take over the exploration after two FTS-contacts. Significant is the increase of the quality measures after the tactile sensor has taken over. After this, the tactile sensor does not lose control over the exploration procedure at any time.

4.3 Continuous Skill for Edge Tracking

The next experiment involves a skill with continuous movement according to the proposed control loop presented in section 2. The task is to follow the edge without losing contact to the object surface. For this kind of skill only the accomplishment of certain phases is checked. This experiment has been performed five times.

The first phase involves the alignment of the pad towards the edge using zero-force control followed by the tracking of the edge until the first corner point. The third phase involves the return to the starting position and tracking towards the opposite direction. In all experiments, all phases were completely accomplished. Significant is that this exploration procedure

took one third of the time compared to the edge tracking skill based on discrete movements (250s vs. 80s).

4.4 Comparison of Both Skills

The skill with discrete movements as well as the skill with continuous movements are capable to completely explore an edge. Both skills have benefits and disadvantages. The benefit of the skill with continuous movements is its speed and its simplicity as it needs less states compared to the other skill. In particular, the fast alignment towards the edge using an adapted zero-force control has to be pointed out. As the skill with discrete movements has at first to collect single points to perform the alignment gradually, the skill needs more time and more control processes. Otherwise it is easier for a superior level to supervise these discrete movements. As outliers can be detected by collecting a great amounts of contact points, the exploration behavior becomes very stable. For the exploration of unknown structures, collecting single points is still favored, as it provides the superior level with more possibilities for interaction. Skills only based on a control algorithms need the whole flow of information for the spatial alignment and are not made for interaction. A combination of both approaches seems to be promising.

5 CONCLUSIONS AND FUTURE WORKS

This work introduced a novel framework for the exploration of objects using haptic feedback. This framework consists of three layer: an exploration planner, a skill library with reactive exploration behaviors and a haptic perception layer. Two different skill schemes have been introduced: skills with discrete and skills with continuous movements. The performance of our approach has been evaluated in the evaluation scenario of tracking an edge which is arbitrarily located in space. The tactile sensor and the force-torque complement one another. The determination of contact points using a force-torque sensor is less accurate but independent from the edge angle. It allows the alignment of the sensor pad so that the critical angle for the tactile sensor is under-run and the tactile sensor can take over the exploration procedure in order to undertake a more precise exploration procedure.

Furthermore it became apparent that the exploration with discrete and the exploration with continuous movements have both benefits and drawbacks.

The continuous exploration based on a control behavior is faster but provides less possibilities for interaction. Exploration procedures with discrete movements are slower but are more robust and better to supervise. A combination of both approaches seems to be promising. Future work will include the extension of the skill library and the transfer of an exploration behavior on a humanoid robot hand.

REFERENCES

- Chen, N., Zhang, H., and Rink, R. (1995). Edge tracking using tactile servo. In *IROS '95: Proceedings of the International Conference on Intelligent Robots and Systems-Volume 2*, page 2084, Washington, DC, USA. IEEE Computer Society.
- Heidemann, G. and Schoepfer, M. (2004). Dynamic tactile sensing for object identification. In *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2004*, pages 813–818, New Orleans, USA. IEEE.
- Hu, M.-K. (1962). Visual Pattern Recognition by Moment Invariants. *IEEE Transactions on Information Theory*, 8(2):179–187.
- Klatzky, R., Lederman, S., and Reed, C. (1987). There's more to touch than meets the eye: The salience of object attributes for haptics with and without vision. In *Journal of Experimental Psychology: General*, 116(4), pages 356–369.
- Lederman, S. and Klatzky, R. (1987). Hand movements: A window into haptic object recognition. In *Cognitive Psychology*, 19(3), pages 342–368.
- Schmidt, P. A., Maël, E., and Würtz, R. P. (2006). A sensor for dynamic tactile information with applications in human-robot interaction and object exploration. *Robot. Auton. Syst.*, 54(12):1005–1014.
- Weiss, K. and Woern, H. (2005). The working principle of resistive tactile sensor cells. In *Proceedings of the IEEE International Conference on Mechatronics and Automation, Canada*.

USING THE OAG TO BUILD A MODEL DEDICATED TO MODE HANDLING OF FMS

Nadia Hamani

Departement of Informatics, Paris X University, 200 avenue de la République, 92001 Nanterre, France
nadia.hamani@u-paris10.fr, nadia.hamani@ec-lille.fr

Nathalie Dangoumau, Etienne Craye

LAGIS, Ecole Centrale de Lille, BP48 cité scientifique, Villeneuve d'Ascq, France
nathalie.dangoumau@ec-lille.fr, etienne.craye@ec-lille.fr

Keywords: Flexible Manufacturing Systems, control system, supervision, mode handling, functional modeling.

Abstract: This paper deals with a modeling approach for *mode handling* of Flexible Manufacturing Systems (FMS). We show that using the plant model enables to establish aggregate operations. These are generic entities which depend only on the plant and do not depend on production goals. Aggregate operations are then used to build the model dedicated to mode handling. This study is illustrated through an example of a flexible manufacturing cell.

1 INTRODUCTION

We are interested in problems of monitoring and supervision in a fault tolerant control system dedicated to Flexible Manufacturing Systems (FMS) (Ranky, 1990). According to our approach, the supervision is made up of three functions: *decision*, *piloting*, and *mode handling*. The *monitoring* function (Elkhatabi et al., 1995; Toguyeni et al., 1996) detects and localizes the failures at the plant level. The *decision* function (Berruet et al., 2000) determines the new configuration of the FMS. The functions of *mode handling* and *piloting* (Tawegoum et al., 1994) implement the decisions about the new configuration of the FMS.

In order to achieve the role of *mode handling* within the control system, one should provide models representing the operating modes of the production system and its subsystems. The existing modeling approaches of operating modes of Automated Production Systems (APS) are compared in (Hamani et al., 2006). The advantages of functional modeling approaches are showed. Such approaches are concerned with the services delivered by the FMS rather than production means. Our approach (Hamani et al., 2006) is based on a functional modeling method. This approach is well

adapted to FMSs because it is based on the mission concept (a production goal) which represents the flexibility which characterizes the FMS production. The obtained model is generic. For a given FMS, the predefined functional subsystems (called entities) are instantiated to generate the model. An aggregate operation is a generic entity depending only on the plant and not on production goals.

The purpose of this paper is to present a method to calculate aggregate operations from the plant model. The paper is organized as follows. Section 2 reminds the basic concepts of our modeling method and the steps of building the FMS functional model. Section 3 presents a method to determine aggregate operations from the plant model. An example of a flexible manufacturing cell is used to illustrate this study.

2 THE FMS MODEL

2.1 Basic Concepts

An FMS produces simultaneously a set of parts. Usually we desire to change production goals. That is why the mission concept is introduced in (Hamani et al., 2006). A mission (\mathcal{M}) is the subset of Logical Operating Sequences (LOS) which are

produced simultaneously. A LOS is a set of ordered machining functions performed on some parts. A LOS is noted $LOS f_1 \dots f_n$ or $LOS f_i (i = 1, n)$.

With each function of a Logical Operating Sequence is associated its possible achievements. They are aggregate operations for which the machining operation is defined. An aggregate operation is a generic entity which depends only on the FMS plant and not on production goals. An aggregate operation corresponding to a machining Major Characteristic Area (MCA) noted $Op_{MCA_machining}$ is a set of the corresponding elementary machining operations and Access Transfers. MCA concept is defined in (Hamani et al., 2006).

In an FMS, an operation (Op) is defined as a function carried out by a resource (Berruet et al., 2000; Toguyeni et al., 2003). An operation is noted Op_{R_i, f_i} where f_i is the performed function and R_i the resource which implements it. An elementary operation is an operation carried out only once, continuously, i.e. without the possibility to choose another alternative during the normal execution of the operation.

Access Transfers (TrA) associated with a machining area (or a MCA), noted $TrA_{machining_MCA}$, correspond to the set of elementary transfer operations that connect this area to the other MCA of the FMS. An elementary transfer (TrE) is performed by one resource between two MCA. An elementary transfer is noted $TrE_{R_i}^{S \rightarrow D}$ with S a source CA, D a destination CA, and R_i the transfer resource.

2.2 The Specification Steps

The specification steps (Figure 1) of the FMS functional model are described in the following.

- 1st Step:** Identification of the entities of the model
- list the missions that the FMS should carry out
 - list for each mission its corresponding Logical Operation Sequences
 - for each Logical Operating Sequence identify the corresponding machining functions
- A machining function is implemented by one or several elementary machining operations. Each one is belonging to an aggregate operation.
- identify the aggregate operations of the FMS (see the 2nd step)
 - for each aggregate operation, identify the resources which perform it (see the 3rd step)

2nd Step: Determination of aggregate operations (Figure 2). For each machining area of the FMS:

- identify elementary machining operations which are performed in this area
- identify the Access Transfers related to this area
- gather elementary machining operations together with Access Transfers identified previously to obtain aggregate operations

3rd Step: Determination of the resources that perform elementary operations

- For each aggregate operation:
- associate with each elementary machining operation the resource or the configuration of the resource (in the case of a polyvalent resource) which performs it
 - associate also with each elementary transfer operation the resource (or the resources) which performs it, redundant resources are linked with a logical OR.

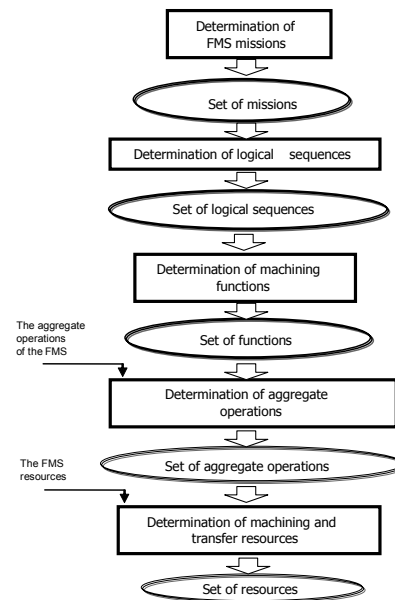


Figure 1: Specification steps of the FMS entities.

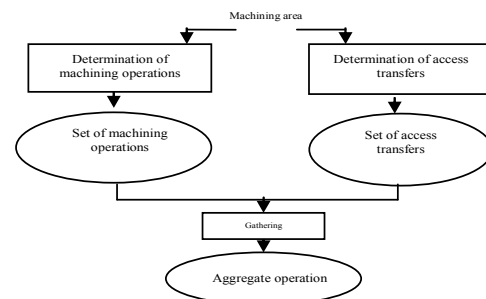


Figure 2: Aggregate operations specification.

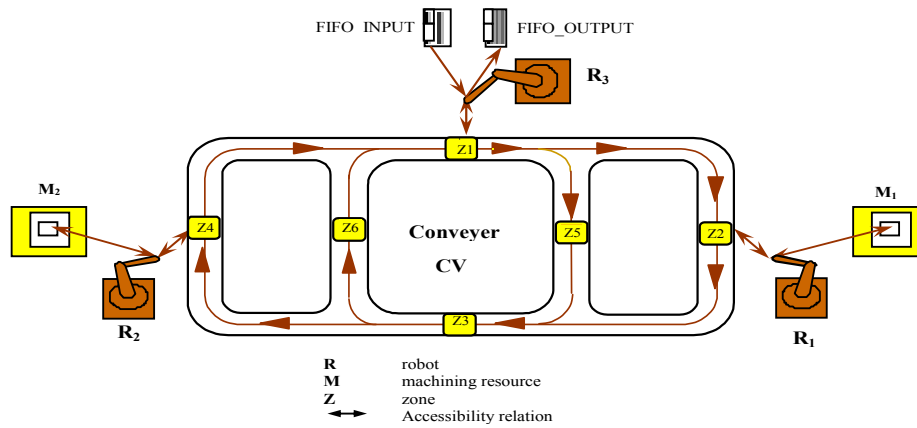


Figure 3: An example of a flexible cell.

The functional model of the machining cell is represented using the following entities:

- The missions
- The Logical Operating Sequences
- The machining functions
- The aggregate operations
 - elementary machining operations
 - Access Transfers (set of transfer operations)
- Transfer resources, machining resources

2.3 Illustration Example

Consider an example of a flexible manufacturing cell (Figure 3) with two machines M_1 and M_2 and INPUT/OUTPUT buffers. The machines are loaded with a transport system using three robots R_1 , R_2 and R_3 and a conveyor (CV). Moving directions of CV are $Z_1 \rightarrow (Z_2 \text{ or } Z_5)$, $(Z_2 \text{ or } Z_5) \rightarrow Z_3$, $Z_3 \rightarrow (Z_4 \text{ or } Z_6)$, $(Z_4 \text{ or } Z_6) \rightarrow Z_1$. It is assumed that M_1 is loaded with R_1 and M_2 is loaded with R_2 . The parts are loaded on the conveyor using the robot R_3 . The machining functions performed by the system are turning (t) and milling (m). Turning is carried out by M_1 , milling by M_1 and M_2 .

According to the functional requirements of this illustration example, three missions can be required by the operator: \mathcal{M}_1 , \mathcal{M}_2 and \mathcal{M}_3 . The corresponding Logical Operating Sequences are the following: \mathcal{M}_1 : LOS_1 and LOS_2 , \mathcal{M}_2 : LOS_1 , LOS_2 and LOS_{12} and \mathcal{M}_3 : LOS_1 , LOS_{12} and LOS_{21}

The machining functions which compose each Logical Operating Sequence are the following: LOS_1 : turning; LOS_2 : milling; LOS_{12} : turning then milling; LOS_{21} : milling then turning.

Turning function is performed by the elementary machining operation $Op_{M1,t}$ belonging to the aggregate operation Op_{M1} . Milling function is

performed by the elementary machining operation $Op_{M1,m}$ belonging to the aggregate operation Op_{M1} or by the elementary machining operation $Op_{M2,m}$ belonging to the aggregate operation Op_{M2} .

For the machining area M_1 : the elementary machining operations performed by M_1 are $Op_{M1,t}$ and $Op_{M1,m}$. Access Transfers related to M_1 are $Tr_{A_{M1}} = AND (Tr_{MCA_source \rightarrow M1}, Tr_{M1 \rightarrow MCA_destination})$.

This notation is using the logical AND and OR and also three distinct levels: ‘{’ for the first level, ‘[’ for the second level and ‘(’ for the third level.

Section 3 presents a method to determine Tr_A using the plant model.

The aggregate operation related to the machining area M_1 is $Op_{M1} = AND [OR (Op_{M1,t}, Op_{M1,m}), Tr_{A_{M1}}]$. The aggregate operation related to the machining area M_2 is obtained in the same manner.

Op_{M1} is performed by the following resources: the polyvalent machining resource M_1 performs the elementary operations $Op_{M1,t}$ et $Op_{M1,m}$. For transfer resources: R_1 performs the elementary transfer operations $TrE_{R1}^{Z2 \rightarrow M1}$ and $TrE_{R1}^{M1 \rightarrow Z2}$; R_2 performs the elementary transfer operations $TrE_{R2}^{M2 \rightarrow Z4}$ and $TrE_{R2}^{Z4 \rightarrow M2}$; R_3 performs the elementary transfer operations $TrE_{R3}^{IN \rightarrow Z1}$ and $TrE_{R3}^{Z1 \rightarrow OUT}$; CV performs the elementary transfer operations $TrE_{CV}^{Z1 \rightarrow Z2}$, $TrE_{CV}^{Z2 \rightarrow Z3}$, $TrE_{CV}^{Z1 \rightarrow Z5}$, $TrE_{CV}^{Z5 \rightarrow Z3}$, $TrE_{CV}^{Z3 \rightarrow Z4}$, $TrE_{CV}^{Z4 \rightarrow Z1}$, $TrE_{CV}^{Z3 \rightarrow Z6}$ and $TrE_{CV}^{Z6 \rightarrow Z1}$.

The obtained model (AND/OR graph) for the machining cell is represented in Figure 4. The underlined entities are not developed. AND nodes do not have any notation, however OR nodes are denoted using +. These nodes correspond to an

inclusive OR or an exclusive OR according to the constraints given in the functional requirements. For example, an exclusive logical OR is necessary for safety reasons, like two machining operations which are performed by the same resource for instance.

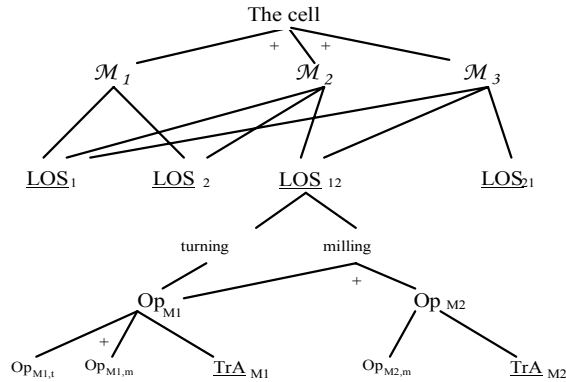


Figure 4: An extract of the functional model of the machining cell.

3 DETERMINATION OF ACCESS TRANSFERS

In order to determine TrA, a first step consists in listing symmetrical transfers between MCA representing both source and destination areas. Then it is necessary to refine these transfers until obtaining elementary transfer operations.

Once Access Transfers are determined, it is necessary to identify elementary transfers which compose them. If there is a direct accessibility between two MCA then $Tr_{MCA_source \rightarrow MCA_destination}$ corresponds to an elementary transfer. If not, it is necessary to refine the transfers between the Characteristic Areas until obtaining elementary transfers. The possible paths are then established and those which are redundant are linked together with a logical OR. For example:

$$Tr_{IN \rightarrow M1} = AND (TrE_{R3}^{IN \rightarrow Z1}, TrE_{CV}^{Z1 \rightarrow Z2}, TrE_{R1}^{Z2 \rightarrow M1}).$$

Due to increasing complexity of FMS, it could be difficult to identify all the elementary transfers which compose Access Transfers. That is why we propose to determine them from the Operational Accessibility Graph (OAG) (Berruet et al., 2000), a graph which represents the FMS plant. The OAG formalizes all the accessibilities between the characteristic areas more precisely than informal specifications provided in the functional requirements.

To build an OAG, a partition of all elementary operations is carried out and the concept of node is introduced to simplify the modeling process. This concept is defined in the following.

A **node** consists of an elementary operation or some elementary operations. This regrouping is governed by rules about the operations taxonomy (Berruet et al., 2000; Toguyéni et al., 2003). The nodes form OAG entities and allow relating the operations using accessibility relations.

Based on this definition, several nodes are defined: storage, machining, assembly, link, and transfer nodes. The nodes are then linked together using accessibility relations in order to build the OAG.

3.1 The Operational Accessibility Graph

The Operational Accessibility Graph (OAG) is a directed graph where nodes are subsets of **operations** performed by the resources of the system and the arcs represent the **accessibility relations** between operations (Toguyéni et al., 2003). The OAG represents all the flexibilities of an existing plant or a plant being designed. It is obtained following these steps:

1st step- Identification of elementary operations of the FMS: in this step elementary operations of machining, storage (passive, active), and transfer are identified.

2nd step- Regrouping the elementary operations: the elementary operations carried out on the same area and the equivalent elementary transfer operations are gathered. A partition of all the operations is thus obtained.

3rd step- Building the graph: a node is associated with each operations subset established in the previous step. The nodes of the OAG are thus obtained. Then these nodes are connected with respect to the accessibility between operations. The OAG structure is then determined.

The method is applied to the illustration example (Figure 3).

1) The elementary machining operations are already identified ($Op_{M1,t}$, $Op_{M1,m}$, $Op_{M2,m}$) and the elementary transfer operations ($TrE_{R1}^{M1 \rightarrow Z2}$, $TrE_{R1}^{Z2 \rightarrow M1}$, $TrE_{R2}^{M2 \rightarrow Z4}$, $TrE_{R2}^{Z4 \rightarrow M2}$, $TrE_{R3}^{IN \rightarrow Z1}$, $TrE_{R3}^{Z1 \rightarrow OUT}$, $TrE_{CV}^{Z1 \rightarrow Z2}$, $TrE_{CV}^{Z2 \rightarrow Z3}$, $TrE_{CV}^{Z1 \rightarrow Z5}$, $TrE_{CV}^{Z5 \rightarrow Z3}$, $TrE_{CV}^{Z3 \rightarrow Z4}$, $TrE_{CV}^{Z4 \rightarrow Z1}$, $TrE_{CV}^{Z3 \rightarrow Z6}$, $TrE_{CV}^{Z6 \rightarrow Z1}$).

It is necessary to add the following storage operations:

- Storage IN and storage OUT which are passive;
- Storage Z_1 , storage Z_2 , storage Z_3 , storage Z_4 , storage Z_5 and storage Z_6 which are active.

2) Concerning the regroupings:

- One gathers $Op_{M1,t}$ and $Op_{M1,m}$ in a complex operation on M_1 .
- Linking operations are: Link Z_1 , Link Z_2 , Link Z_3 , Link Z_4 , Link Z_5 and Link Z_6 .
- The functions fulfilled by the elementary transfer operations are all distinct. There is no regrouping of transfers.

3) Table 1 summarizes the correspondence between the nodes and the operations which compose them. The resulting OAG is represented in Figure 5.

Note: on Figure 5, storage nodes IN and OUT, link nodes as well as machining nodes correspond to characteristic areas of the cell. The subset formed only by storage nodes and machining nodes corresponds to main characteristic areas.

The obtained model is used to calculate the elementary transfers as shown in the following.

3.2 A Procedure for Determination Elementary Transfers

Based on the OAG, the following procedure is proposed in order to calculate Access Transfers.

Beginning of the procedure:

1st Step: determination of Access Transfers associated with machining nodes

For each machining node of the OAG:

- determine the paths which connect it with the others machining nodes and the input of the cell;
 - determine the paths which enable unloading parts onto other machining nodes and the output of the cell;
- The obtained paths are linked with a logical OR;

End For;

2nd Step: determination of elementary transfers which compose the Access Transfers:

Do again for each identified path in the previous step

If the path relates two successive nodes of the OAG

Then the path is an elementary transfer

If not determine the paths which compose it

The redundant transfers are linked with a logical OR; do not consider the paths which go over a transfer node twice and those that contain intermediary machining nodes;

Until all the obtained paths are elementary.

End of the procedure.

Table 1: The correspondence between nodes and operations.

N1	N2	N3	N4	N5	N6	N7	N8
Storage IN	$TrE_{R_3}^{IN \rightarrow Z_1}$	Link Z_1	$TrE_{CV}^{Z_1 \rightarrow Z_2}$	Link Z_2	$TrE_{R_1}^{Z_2 \rightarrow M_1}$	Machining M_1 $Op_{M1,t}$ $Op_{M1,m}$	$TrE_{R_1}^{M_1 \rightarrow Z_2}$
N9	N10	N11	N12	N13	N14	N15	N16
$TrE_{CV}^{Z_1 \rightarrow Z_5}$	Link Z_5	$TrE_{CV}^{Z_2 \rightarrow Z_3}$	$TrE_{CV}^{Z_5 \rightarrow Z_3}$	Link Z_3	$TrE_{CV}^{Z_3 \rightarrow Z_4}$	Link Z_4	$TrE_{R_2}^{Z_4 \rightarrow M_2}$
N17	N18	N19	N20	N21	N22	N23	N24
Machining M_2 $Op_{M2,m}$	$TrE_{R_2}^{M_2 \rightarrow Z_4}$	$TrE_{CV}^{Z_3 \rightarrow Z_6}$	Link Z_6	$TrE_{CV}^{Z_6 \rightarrow Z_1}$	$TrE_{CV}^{Z_4 \rightarrow Z_1}$	$TrE_{R_3}^{Z_1 \rightarrow OUT}$	Storage OUT

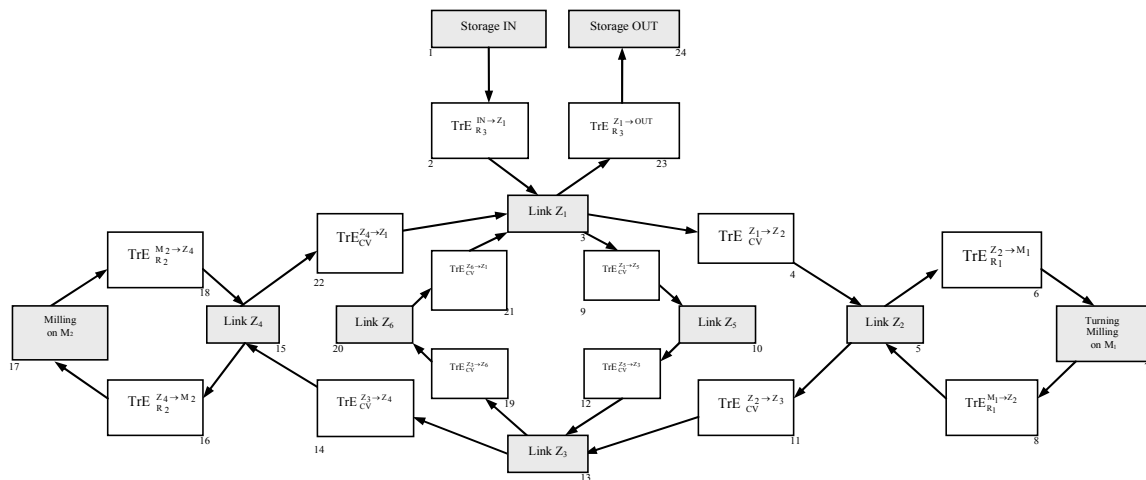


Figure 5: The OAG of the illustration example.

For each machining node, the access paths which are associated with it, added with machining operations carried out on this node, are linked with logical AND. This regrouping is an aggregate operation.

For the illustration example, the access paths calculated in the first step of the procedure for the machining area M_1 are the following: OR ($Tr_{IN \rightarrow M_1}$, $Tr_{M_2 \rightarrow M_1}$) and OR ($Tr_{M_1 \rightarrow M_2}$, $Tr_{M_1 \rightarrow OUT}$). The following elementary transfers are then obtained using the second step of the procedure.

$$Tr_{IN \rightarrow M_1} = \text{AND} (TrE_{R_3}^{IN \rightarrow Z_1}, TrE_{CV}^{Z_1 \rightarrow Z_2}, TrE_{R_1}^{Z_2 \rightarrow M_1});$$

$$Tr_{M_2 \rightarrow M_1} = \text{AND} (TrE_{R_2}^{M_2 \rightarrow Z_4}, TrE_{CV}^{Z_4 \rightarrow Z_1}, TrE_{CV}^{Z_1 \rightarrow Z_2},$$

$$TrE_{R_1}^{Z_2 \rightarrow M_1}); Tr_{M_1 \rightarrow M_2} = \text{AND} (TrE_{R_1}^{M_1 \rightarrow Z_2}, TrE_{CV}^{Z_2 \rightarrow Z_3},$$

$$TrE_{CV}^{Z_3 \rightarrow Z_4}, TrE_{R_2}^{Z_4 \rightarrow M_2}); Tr_{M_1 \rightarrow OUT} = \text{AND} \{ TrE_{R_1}^{M_1 \rightarrow Z_2},$$

$$TrE_{CV}^{Z_2 \rightarrow Z_3}, \text{OR} [\text{AND} (TrE_{CV}^{Z_3 \rightarrow Z_4}, TrE_{CV}^{Z_4 \rightarrow Z_1}), \text{AND}$$

$$(TrE_{CV}^{Z_3 \rightarrow Z_6}, TrE_{CV}^{Z_6 \rightarrow M_2})], TrE_{R_3}^{Z_1 \rightarrow OUT} \}.$$

$$\text{Finally, } Tr_{A_{M_1}} = \text{AND} [\text{OR} (Tr_{IN \rightarrow M_1}, Tr_{M_2 \rightarrow M_1}), \text{OR} (Tr_{M_1 \rightarrow M_2}, Tr_{M_1 \rightarrow OUT})].$$

4 CONCLUSIONS

In this paper our modeling method dedicated to FMS *mode handling* is extended. The FMS functional model is obtained by a modular and hierarchical decomposition leading to the elementary machining and transfer operations. For large scale systems, it is difficult to obtain all possible redundancies of a plant. So we propose to determine aggregate operations associated with machining areas from the plant model represented by the OAG. The aggregate operations are generic concepts which depend only on the plant and not on production goals. Such method enables to generate automatically aggregate operations for an existing system or a system being designed. The proposed modeling steps are then illustrated through an example of a manufacturing cell.

Further works aim at implementing the proposed method within the information system *CASPAIN_soft* (Ndiaye et al., 2002).

REFERENCES

- Berruet, P., Toguyéni, A.K.A., Craye, E., 2000. Towards implementation of recovery procedures for FMS supervision. *Computers in Industry*, 43, 227-236.
- Elkhattabi, S., Craye, E., Gentina, J.C., 1995. Supervision by the behavior modeling. In *SMC'95, IEEE International Conference on Systems, Man, and Cybernetics*.
- Hamani, N., Dangoumau, N., Craye, E., 2005. A comparative study of mode handling approaches. In *CiE'05, 35th International Conference on Computers & Industrial Engineering*.
- Hamani, N., Dangoumau, N., Craye, E., 2006. A functional modeling approach for Flexible Manufacturing Systems mode handling. In *INCOM'06, 12th IFAC Symposium on Information Control Problems in Manufacturing*. Elsevier Press.
- Ndiaye, D., Bigand, M., Corbeel, D., Bourey, J.-P., 2002. Information system for production engineering. *Int. J. of Computer Integrated Manufacturing*, 15(3), 233-241.
- Ranky, P., 1990. *Flexible manufacturing cells and systems in CIM*, CIMware Ltd. Guildford, U.K.
- Tawegoum, R., Castelain, E., Gentina, J.-C., 1994. Real time piloting of flexible manufacturing systems. *European J. of Operational Research*, 78, 252-261.
- Toguyéni, A.K.A., Craye, E., Gentina, J.-C., 1996. A framework to design a distributed diagnosis in FMS. In *SMC'96, IEEE International Conference on Systems, Man, and Cybernetics*.
- Toguyéni, A.K.A., Berruet, P., Craye, E., 2003. Models and algorithms for failure diagnosis and recovery in Flexible Manufacturing Systems. *Int. J. of Flexible Manufacturing Systems*, 15(1), 57-85.

be expressed as

$${}^{\circ}\bar{m}_i = {}^{\circ}t^* + {}^{\circ}R_* {}^* \bar{m}_i \quad (3)$$

where ${}^{\circ}t^*(t)$ is the translation between the two frames, and ${}^{\circ}R_*(t)$ is the rotation matrix which brings ${}^* \mathcal{F}$ onto \mathcal{F} .

Intuitively, the control objective can be regarded as the task of moving the robot so that ${}^{\circ}\bar{m}_i(t)$ equals ${}^* \bar{m}_i \forall i$ as $t \rightarrow \inf$. However, an image-based visual servoing system is not expected to calculate the real coordinates of these feature points. Instead, it can only extract the image coordinates of those same points. That is, the coordinates ${}^{\circ}p_i$ and *p_i of the projection of the feature points onto the image plane, given by: ${}^{\circ}p_i = A {}^{\circ}m_i$ and ${}^*p_i = A {}^*m_i$, where A is the matrix of the intrinsic parameters of the camera. So, the real control objective becomes that of moving the robot so that ${}^{\circ}p_i$ equals *p_i .

This idea will be further detailed in the following section.

2 DESIGN OF THE CONTROLLER

As mentioned above, the control objective is to regulate the camera to a desired position relative to the target object. In order to achieve this control objective the image coordinates at the desired position have to be known. This can be done by taking an image of the target object at the desired position and extracting the feature points using an image processing algorithm. Once a picture is taken and the image coordinates are extracted, those coordinates can be stored for future reference. It is assumed that the motion of the camera is unconstrained and the linear and angular velocities of the camera can be controlled independently. Furthermore the camera has to be calibrated, i. e. the intrinsic parameters of the camera A must be known.

As we mentioned earlier, in the Euclidean space the control objective can be expressed as:

$${}^{\circ}R_*(t) \rightarrow I_3 \quad \text{as} \quad t \rightarrow \inf \quad (4)$$

$$\|{}^{\circ}t^*(t)\| \rightarrow 0 \quad \text{as} \quad t \rightarrow \inf \quad (5)$$

and the translation regulation error $e(t) \in \mathbb{R}^3$ can be defined using the extended normalized coordinates as:

$$\begin{aligned} e &= {}^{\circ}m_e - {}^*m_e \\ &= \begin{bmatrix} \frac{{}^{\circ}x_i}{{}^{\circ}z_i} - \frac{{}^*x_i}{{}^*z_i} & \frac{{}^{\circ}y_i}{{}^{\circ}z_i} - \frac{{}^*y_i}{{}^*z_i} & \ln\left(\frac{{}^{\circ}z_i}{{}^*z_i}\right) \end{bmatrix}^T \end{aligned} \quad (6)$$

The translation regulation objective can then be quantified as the desire to regulate $e(t)$ in the sense that

$$\|e(t)\| \rightarrow 0 \quad \text{as} \quad t \rightarrow \inf. \quad (7)$$

It can be easily verified that if (7) is satisfied, the extended normalized coordinates will approach the desired extended normalized coordinates, i. e.

$${}^{\circ}m_i(t) \rightarrow {}^*m_i(t) \quad \text{and} \quad {}^{\circ}z_i(t) \rightarrow {}^*z_i(t) \quad (8)$$

as $t \rightarrow \inf$. Moreover, if (7) and (8) are satisfied, (5) is also satisfied.

Similarly, the rotation regulation objective in (4) can be expressed in terms of its quaternion vector $q = [q_0 \ \tilde{q}]^T$, $\tilde{q} = [q_1 \ q_2 \ q_3]^T$ (Chou and Kamel, 1991) by:

$$\|\tilde{q}(t)\| \rightarrow 0 \quad \text{as} \quad t \rightarrow \inf. \quad (9)$$

In that case, if (7) and (9) are satisfied, the control objective stated in (4) is also satisfied.

For such translational and rotational control objectives, it was shown in (Hu et al., 2006) that the closed-loop error system is given by:

$$\dot{q}_0 = \frac{1}{2} \tilde{q}^T K_{\omega} (I_3 - \tilde{q}^{\times})^{-1} \tilde{q} \quad (10)$$

$$\dot{\tilde{q}} = -\frac{1}{2} K_{\omega} (q_0 I_3 - \tilde{q}^{\times}) (I_3 - \tilde{q}^{\times})^{-1} \tilde{q} \quad (11)$$

$${}^*z_i \dot{e} = -K_v e + \tilde{z}_i L_{\omega} \omega_c \quad (12)$$

and the control inputs by:

$$\omega_c = -K_{\omega} (I_3 - \tilde{q}^{\times})^{-1} \tilde{q} \quad (13)$$

$$v_c = \frac{1}{\alpha_i} L_v^{-1} (K_v e + {}^* \hat{z}_i L_{\omega} \omega_c) \quad (14)$$

where ${}^* \hat{z}_i = e^T L_{\omega} \omega_c$ is an estimation for the unknown *z_i ; \tilde{q}^{\times} is the anti-symmetric matrix representation of the vector \tilde{q} ; L_v, L_{ω} are the linear and angular Jacobian-like matrices; $K_{\omega}, K_v \in \mathcal{R}^{3 \times 3}$ are diagonal matrices of positive constant control gains; and the estimation error $\tilde{z}(t) \in \mathcal{R}$ is defined as $\tilde{z}_i = {}^*z_i - {}^* \hat{z}_i$.

A proof of stability for the controller above can be found in (Hu et al., 2006).

3 IMPLEMENTATION

In this work the task of controlling a robot with a visual-servoing algorithm was divided into four major parts:

1. Capturing images using a video camera.
2. Processing the images to get the coordinates of the feature points.
3. Calculating the input variables, i. e. the velocities of the robot endeffector.
4. Moving the robot according to the given input variables.

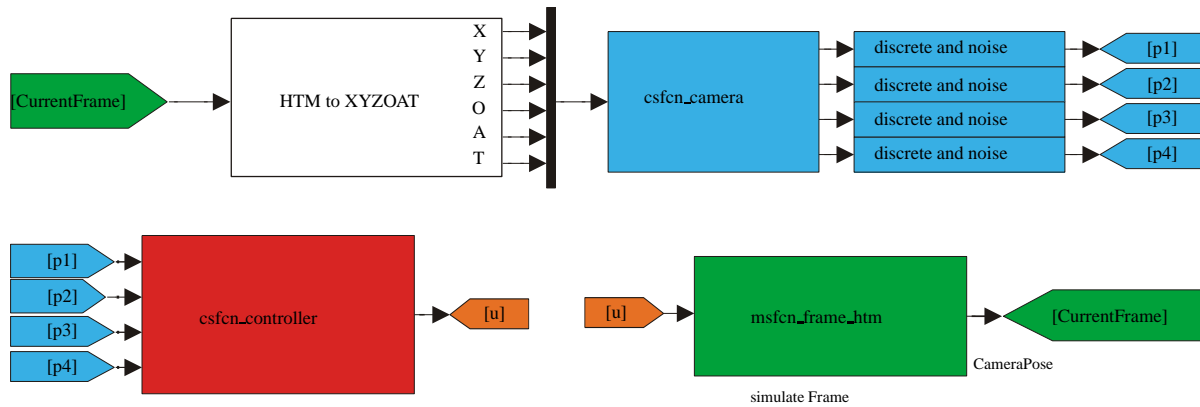


Figure 2: The Simulink-Model.

The hardware available for this task consisted of a Kawasaki industrial robot with a camera mounted on its endeffector and vision-sensor network for capturing and processing images from the camera.

The controller was implemented as a C++ class, in order to guarantee the future reusability of the code in different scenarios (as it will be better explained below). Initially, the controller requires the camera intrinsic parameters and the desired image coordinates of the four feature points. Next it extracts the current image coordinates and then it calculates the linear and angular velocities of the endeffector, i. e. the input variables of the controller.

In order to safely test the controller, the real pair robot/camera was replaced by two different simulators. The first simulator, which was implemented in MATLABSIMULINK, simulates an arbitrary motion of the camera in space. The camera is represented by a coordinate frame as it is briefly described later. With this simulator it is possible to move the camera according to the exact given velocities.

In all testing scenarios performed, the camera simulator needed to output realistic image coordinates of the feature points. To achieve that, the simulator relied on a very accurate calibration procedure (Hirsh et al., 2001) as well as exact coordinates of the feature points in space with respect to the camera. Given that, the simulator could then return the image coordinates of the feature points at each time instant t . The camera simulator was also implemented as a C++ class.

The second simulator is a program provided by Kawasaki Japan. This simulator can execute the exact same software as the real robot and therefore it allowed for the testing of the code used to move the real robot. This code is responsible for performing the forward and inverse kinematics, as well as the dynamics of the robot.

The basic structure of the simulink model can be seen in Figure 2.

In this work, we will not report the results from the tests with the real robot. So, in order to demonstrate the system in a more realistic setting, noise was added to the image processing algorithm and a time discretization of the image acquisition was introduced to simulate the camera.

3.1 Describing the Pose and Velocity of Objects

The position and orientation (pose) of a rigid object in space can be described by the pose of an attached coordinate frame. There are several possible notations to represent the pose of a target coordinate frame with respect to a reference one, including the homogeneous transformation matrix, Euler Angles, etc. (Saeed, 2001) and (Spong and Vidyasagar, 1989). Since we were using the Kawasaki robot and simulator, we adopted the XYZOAT notation as defined by Kawasaki. In that system, the pose of a frame \mathcal{F} with respect to a reference frame ${}^*\mathcal{F}$ is described by three translational and three rotational parameters. That is, the cartesian coordinates X, Y, and Z, plus the Orientation, Approach, and Tool angles in the vector form:

$$x = [x \ y \ z \ \phi \ \theta \ \psi]^T$$

This notation is equivalent to the homogeneous transformation matrix:

$$H = \begin{bmatrix} C\phi C\theta C\psi - S\theta S\psi & -C\phi C\theta S\psi - S\phi C\psi & C\phi S\theta & x \\ S\phi C\theta C\psi + C\theta S\psi & -S\phi C\theta S\psi + C\phi C\psi & S\phi S\theta & y \\ -S\theta C\psi & S\theta S\psi & C\theta & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

where $S\alpha = \sin(\alpha)$ and $C\alpha = \cos(\alpha)$ and the angles ϕ , θ , and ψ correspond, respectively, to O, A, and T and represent:

- Rotation of ϕ about the \bar{a} -axis (z-axis of the moving frame) followed by
- Rotation of θ about the \bar{b} -axis (y-axis of the moving frame) followed by
- Rotation of ψ about the \bar{a} -axis (z-axis of the moving frame).

4 RESULTS

The controller was tested in three different scenarios: pure linear movement; pure angular movement; and combined movement. In all these cases, the camera intrinsic parameters were:

$$A = \begin{bmatrix} 122.5 & -3.7737 & 100 \\ 0 & 122.6763 & 100 \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$

The four feature points were arranged in a square around the origin of the reference frame and had coordinates:

$$\begin{aligned} {}^w c_1 &= [-5 \ -5 \ 0]^T \\ {}^w c_2 &= [5 \ 5 \ 0]^T \\ {}^w c_3 &= [5 \ -5 \ 0]^T \\ {}^w c_4 &= [-5 \ 5 \ 0]^T \end{aligned} \quad (17)$$

The desired pose of the camera was the same for all the simulations, only the start poses differ. The desired pose of the camera was 20 units above the target object, exactly in the middle of the four feature points. The camera was facing straight towards the target object, i. e. its z-axis was perpendicular to the xy-plane and pointing out. The x-axis of the camera was antiparallel to the x-axis of the reference frame and the y-axis parallel to the y-axis of the reference frame. This pose can be described by:

$${}^*x_w = [0 \ 0 \ 20 \ 0^\circ \ 180^\circ \ 0^\circ]^T \quad (18)$$

The control gains used for the controller were:

$$K_v = \begin{bmatrix} 25 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 25 \end{bmatrix} \quad (19)$$

$$K_\omega = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.5 \end{bmatrix} \quad (20)$$

As mentioned in Section 3, we simulated both the noisy and the discrete aspects of a real camera. That is, we added noise to the image coordinates to simulate a typical accuracy of 0.5 pixels within a random error of 2 pixels in any direction. These values were obtained experimentally using real images and a previously developed feature extraction algorithm.

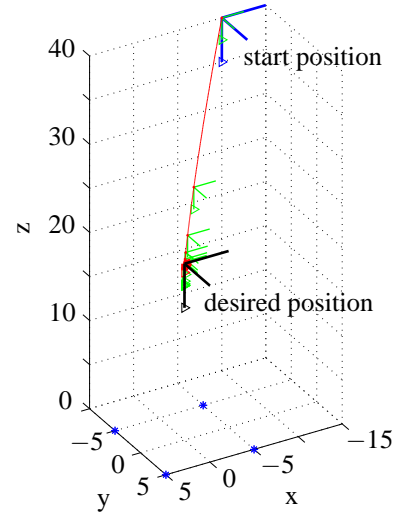


Figure 3: Linear motion simulation in euclidean space.

4.1 Linear Motion

In this simulation the camera did not rotate, i. e. the orientation of the camera in the start pose was the same as in the desired pose. The camera was simply moved 20 units along the z-axis of the reference frame and -10 units along the x- and the y-axis of the reference frame. The start pose was given by

$${}^\circ x_w = [-10 \ -10 \ 40 \ 0^\circ \ 180^\circ \ 0^\circ]^T \quad (21)$$

Figure 3 shows the pose of the camera at ten time instants. The z-axis of the camera – the direction in which the camera is “looking” – is marked with a triangle in the figure. The four points on the target object, lying in the xy-plane, are marked with a star. In Figure 4 the image coordinates of the four points are shown. The image coordinate at the start pose is marked with a circle, the image coordinate at the desired pose with a star.

In Figure 5 the control inputs are shown. The first part shows the linear velocities, the second part the angular velocities of the camera.

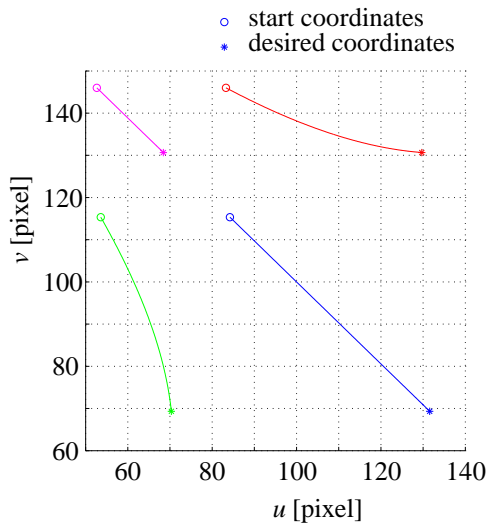


Figure 4: Coordinates of the feature points in image space for the linear motion.

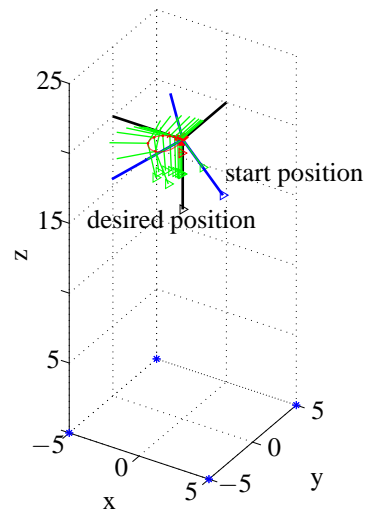


Figure 6: Angular motion simulation in euclidean space.

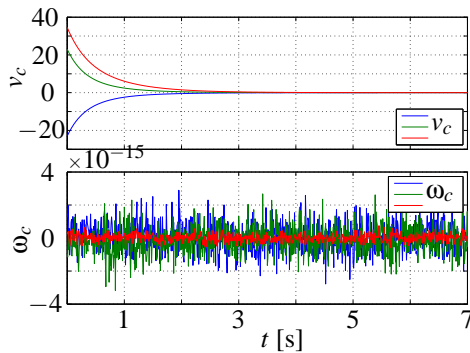


Figure 5: Control input for the linear motion.

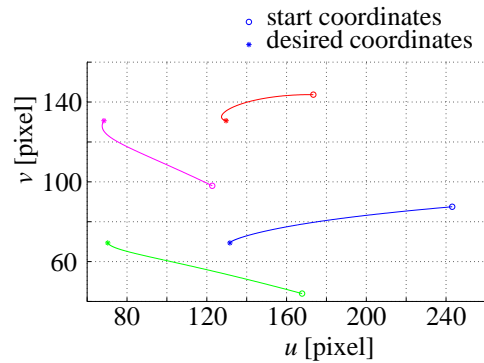


Figure 7: Coordinates of the target points in image space for the angular motion.

4.2 Angular Motion

In this simulation the start position of the camera was identical to the desired position, only the orientation differed. The start pose is given by:

$${}^0\chi_w = [0 \ 0 \ 20 \ 45^\circ \ 150^\circ \ 5^\circ]^T \quad (22)$$

As in Section 4.1 the figures 6, 7 and 8 show the movement of the camera in euclidean space, the image coordinates of the four feature points and the control variables.

4.3 Coupled Motion

In this simulation the camera could perform any generic movement, i. e. both the position and the orientation at the beginning differ from the desired pose of the camera. The start pose is given by:

$${}^0\chi_w = [10 \ -10 \ 40 \ 90^\circ \ 140^\circ \ 10^\circ]^T \quad (23)$$

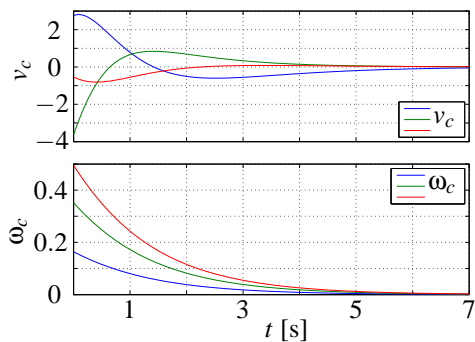


Figure 8: Control input for the angular motion.

As in Section 4.1 the figures 9, 10 and 11 show the movement of the camera in euclidean space, the image coordinates of the four feature points and the control variables.

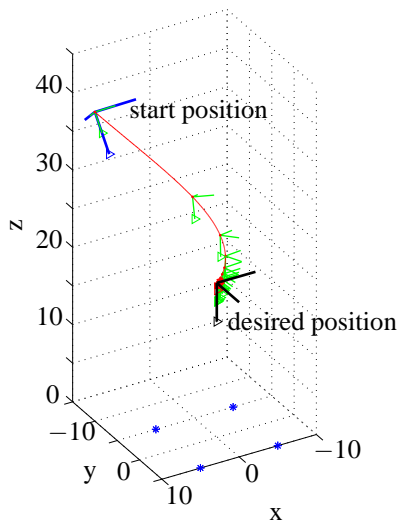


Figure 9: Coupled motion simulation in euclidean space.

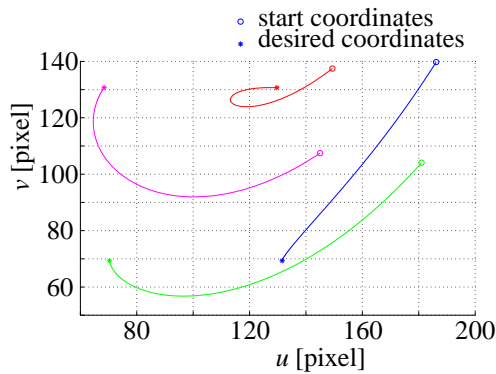


Figure 10: Coordinates of the target points in image space for the coupled motion.

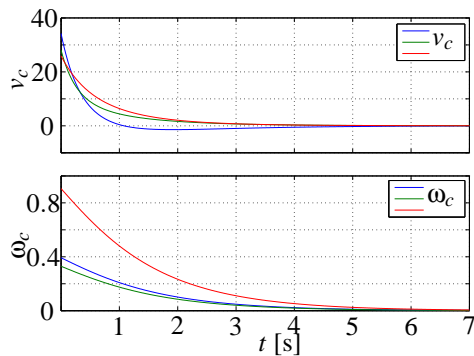


Figure 11: Control input for the coupled motion.

4.4 Influence of Noise

The setup for this simulation is the same as in Section 4.3, but with noise added to the pixels. That is, at each discrete time an image is grabbed a random

Gaussian noise $N(0.5,2)$ is added to the pixel coordinates.

As in Section 4.1 the figures 12, 13 and 14 show the movement of the camera in euclidean space, the image coordinates of the four feature points and the control variables.

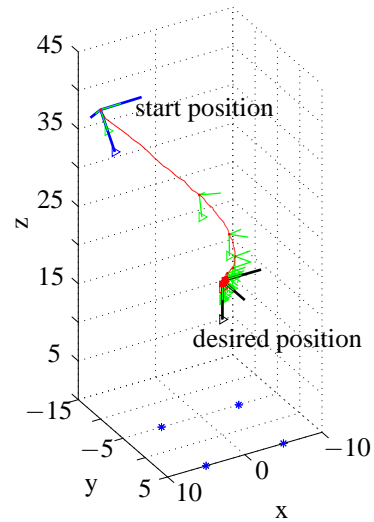


Figure 12: Coupled motion with noise simulation in euclidean space.

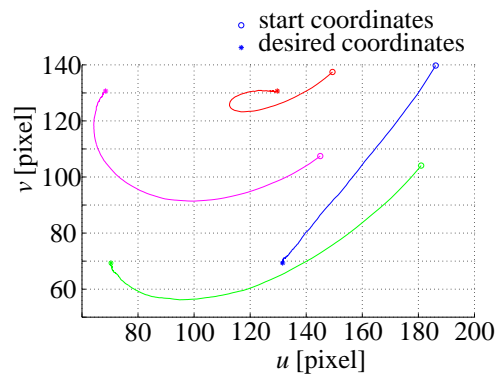


Figure 13: Coordinates of the target point in image space for the coupled motion with noise.

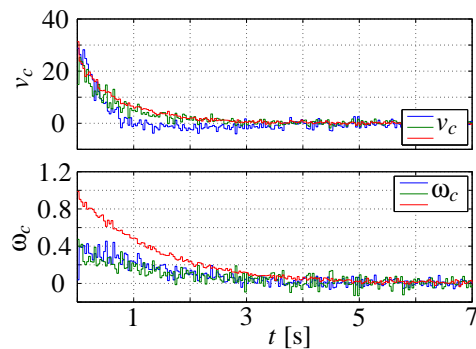


Figure 14: Control input for the coupled motion with noise.

5 CONCLUSIONS

An implementation of an image-based visual servo controller using Matlab and C++ was presented. Various simulations with and without noise were conducted and the controller achieved asymptotic regulation in all cases. This implementation experimentally validates the controller developed in (Hu et al., 2006) and now that the controller is safe to use, new experiments using the real robot can be carried out.

At this point, the control gains were kept small and the discretized intervals were based on a normal camera (30fps). Those choices let us achieve a convergence in less than 7 seconds. However, for real-world applications, those same choices must be revised so that the convergence can be made a lot faster.

The control model and all software modules used in this paper will be made available on line at <http://vigir.missouri.edu>

REFERENCES

- Chou, J. C. K. and Kamel, M. (1991). Finding the position and orientation of a sensor on a robot manipulator using quaternions. *International Journal of Robotics Research*, 10(3):240–254.
- DeSouza, G. N. and Kak, A. C. (2004). A subsumptive, hierarchical, and distributed vision-based architecture for smart robotics. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 34(5).
- Hirsh, R., DeSouza, G. N., and Kak, A. C. (2001). An iterative approach to the hand-eye and base-world calibration problem. In *Proceedings of 2001 IEEE International Conference on Robotics and Automation*, volume 1, pages 2171–2176. Seoul, Korea.
- Hu, G., Dixon, W., Gupta, S., and Fitz-Coy, N. (2006). A quaternion formulation for homography-based visual servo control. In *IEEE International Conference on Robotics and Automation*, pages 2391–2396.
- Hutchinson, S., Hager, G. D., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics & Automation*, 12(5):651–670.
- Saeed, B. (2001). *Introduction to Robotics, Analysis, Systems, Applications*. Prentice Hall Inc.
- Spong, M. W. and Vidyasagar, M. (1989). *Robot Dynamics and Control*. John Wiley & Sons.

ROBOT NAVIGATION MODALITIES

Ray Jarvis

Intelligent Robotics Research Centre, Monash University, Australia

Ray.Jarvis@eng.monash.edu.au

Keywords: Navigation, Modalities.

Abstract: Whilst navigation (robotic or otherwise) consists simply of traversing from a starting point to a goal, there are a plethora of conditions, states of knowledge and functional intentions which dictate how best to execute this process in a manageable, reliable, safe and efficient way. This position paper addresses the broad issues of how a continuum of choices from pure manual or teleoperation control through to fully autonomous operation can be laid out and then selected from, taking into account the variety of factors listed above and the richness of live sensory data available to describe the operational environment and the location of the robot vehicle within it.

1 INTRODUCTION

The dominance of ‘Simultaneous Localisation and Mapping’ (SLAM) (Leonard, 1991) in recent publications on robot navigation can give the false impression that this approach is always the best way of carrying out this task, largely ignoring the fact that there are very few situations where such an approach is either necessary or even feasible, given normal expectations of prior knowledge and functional/safety requirements.

As a simple counter example, why would one want to carry out complex SLAM style navigation in a building for which exact plans are available? Alternatively, if a rich database concerning the geometry and appearance of a reasonably static environment can be constructed off-line with accuracy and convenience and this need only be done once, why not just use this 3D colour rendered map data for continuing robot operations on a day to day basis ever more? In the other extreme, in highly complex and dynamic environments with high risk potentials, such as robotic bushfire fighting operations, why not navigate a robotic vehicle under human teleoperation control to allow the full judgement of human reasoning to apply throughout whilst the operator is in a safe and comfortable place?

There are many other examples between the extremes described above, each requiring its own appropriate navigation modality. In what follows the essentials of robot navigation will be described,

various navigational modalities outlined and a number of case studies presented for illustration purposes. Discussion and conclusions then follow.

2 ROBOT NAVIGATION ESSENTIALS

Six sub-system requirements govern the task of robot navigation:

(a) Localisation (Jarvis, 1993) concerns the fixing of the position and pose of the robot vehicle within its working environment, whether by following the pre-laid lines on the floor, detecting beacons or interpreting natural landmarks (or general environmental metrics and/or appearances). The less preparation required the better but not at the expense of overall efficiency, accuracy and safety. The recent tendency is to try and use on-board acquired sensory data of the operational environment with minimal purposeful marking up of it by way of specific signs.

(b) Environmental mapping concerns the provision or acquisition of data specifying the occupancy, geometry, topology or essential nature of the physical operational environment and sometimes also the identification of relevant objects within it. Such a map may assist localization but must also provide the basis for obstacle avoidance and path planning.

(c) Path planning (Jarvis, 1994) concerns the determination of efficient collision-free and safe

paths from start to goal locations or, in some cases, a coverage pattern of the accessible environment. In many cases paths can only be constructed incrementally as environment mapping data is acquired from on-board sensors (possibly indicating the location of previously unknown obstacles), if not provided beforehand.

(d) Motion Control involves the mechanistic operation of wheels, legs, propellers etc. to drive the robot along the planned path.

(e) Communication amongst sensors, operator, computational resources and mechanism components is also essential. The distribution (and redundancy) of these provisions on-board and off (where there might be a remote base station) are critical to efficiency, timeliness, safety and reliability.

(f) Function refers to the intended operation, whether it be directing water at a fire, picking up suspicious baggage or apprehending a terrorist, or some other requirement. This aspect is often neglected or regarded as a “do last” task in the system design process but should actually be considered first, not only because the type of vehicle, its sensory capabilities and its reliability are dependent on its function but also because the navigation modality may be less critical than the manipulation (or some other task required) when the goal is reached. For example, if the task requires the close supervision by a remote operator (e.g. in defusing a bomb) then a sophisticated autonomous navigation strategy may not be justifiable, even if possible.

Just how the above six aspects are sensibly integrated is critically dependent on the functional requirements, the available prior knowledge of the environment, the dynamics of the situation and, not least, on human risk related considerations.

3 NAVIGATION MODALITIES

For the sake of structure, three dimensions of the robot navigation modality choice process can be identified (See Figure 1):

The first is that of degree of availability of prior knowledge (e.g. maps, views, 3D geometry) or the ease with which this can be acquired off-line (e.g. via laser scanners, stereo views, appearance mapping, etc.). When environmental knowledge suitable for supporting robot navigation (localisation, obstacle avoidance/path planning and

function) is readily available, it makes good sense to use it as it is likely that such an approach would lead to better accuracy, reliability and efficiency than learning such knowledge using on-board sensors alone.

The second dimension is that of the complexity of the defined function and whether human agencies would be required to handle them, whether or not the pure navigational aspects could be automated to some degree. For example, if the complex operation of defusing a bomb via delicate teleoperated manipulation with rich sensory feedback needs the application of expert human skill, the necessary attendance of the expert suggests that the navigation may as well be by teleoperation also, unless this part of the overall task is particularly tedious or time consuming.

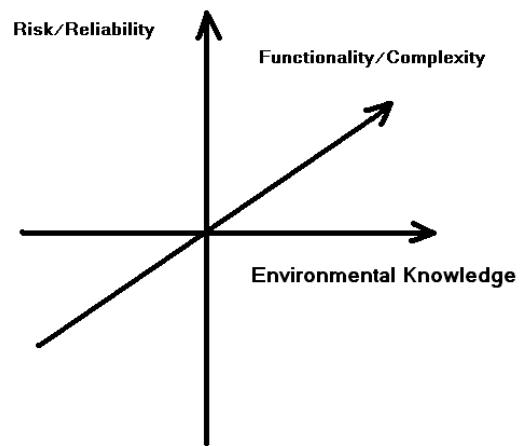


Figure 1: Robot Navigation Modality Choice Factors.

The third dimension is that of risk and reliability requirement factors. For example, having a robot clean a carpet or mow a lawn fully autonomously to obviate human tedium makes good sense, since degrees of unreliability and inefficiency can be tolerated and very little human risk is involved. On the other hand, using the bomb defusing example again, the remoteness of the operator for risk minimisation is the essential factor and the question of modality of navigation may be considered relatively irrelevant, so a flexible mixture of automation and direct teleoperation may be suitable for this application. Guiding a fire tanker to a fire fighting location too hazardous for humans to attend should perhaps be handled entirely by rich sensor feedback supported teleoperation, since the safety of other personnel operating in the vicinity may be more severely jeopardised if a fully autonomous system were used, especially as the situation is likely

to be subject to severe dynamic variation with a moving fire front, changing wind conditions and the extent of other fire fighting vehicle and personnel deployment.

4 FLEXIBLE APPROACH TO ROBOT NAVIGATION

Rather than accepting one rigidly defined robot navigation modality along the spectrum from pure teleoperation to fully autonomous operation, it makes sense to devise ways in which these extremes can be moved between gracefully with smooth variation of the degree of human intervention applied in a hybrid strategy where levels of autonomy can be adjusted for particular tasks and adapt to changing conditions over time. A good example of this approach is where a disabled person is using a wheelchair in complex environments with the aid of robotically inspired sensory and control mechanisms (Jarvis, 2001). The disabled occupant may be permitted a user-adaptive degree of control of the wheelchair within an envelope of safety provided by the robotic instrumentation which adjusts the degree of intervention to the capability of the user to handle the situation over variations of physical reflex, poor vision, degrees of fatigue etc.

Using a three level control strategy (see Figure 2) nicely complements the notion of flexible navigation modality selection. The lowest level can be purely reaction based collision avoidance through stopping or minor trajectory adjustments using close range obstacle sensing as a trigger. The second level can be thought of as “local guidance” which indicates a safe passage over a limited range of movement, generally in the intended direction. The top level is global and includes complete path planning and control transition strategies. In the robotic wheelchair example, the human occupant provides the top level strategy, the second level provides the user with steering advice and the lowest level simply avoids collisions.

In the more general robot navigation situation, the top level could drift between fully human control via teleoperation and fully autonomous operation, with the lower two levels playing their roles in supporting the global strategy. For example, a teleoperator, like the wheelchair user, can direct the activities of the robot using the advice of the second level and accepting the collision avoidance reaction level as a safety precaution should his attention stray.

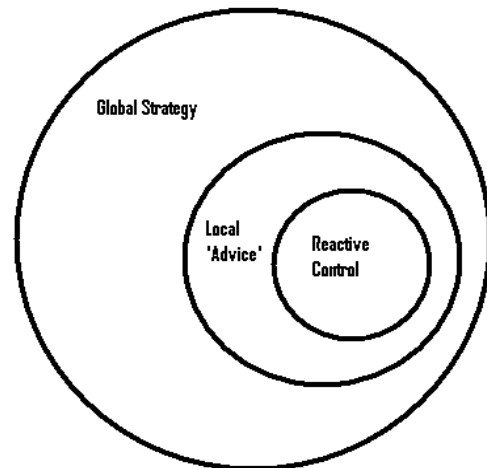


Figure 2: Multi-level Control Hierarchy.

5 CASE STUDIES

The user adaptive robotic wheelchair (Jarvis, 2001) described above is shown in Figure 3. The user can indicate navigation intention using human gaze detection but near collisions impose increasing degrees of instrument driven navigation intervention, with control being handed back to the user gradually as near collision statistics improve. The main environment sensor is a Erwin Sick laser range finder. GPS is also provided for guidance as a non-essential convenience.

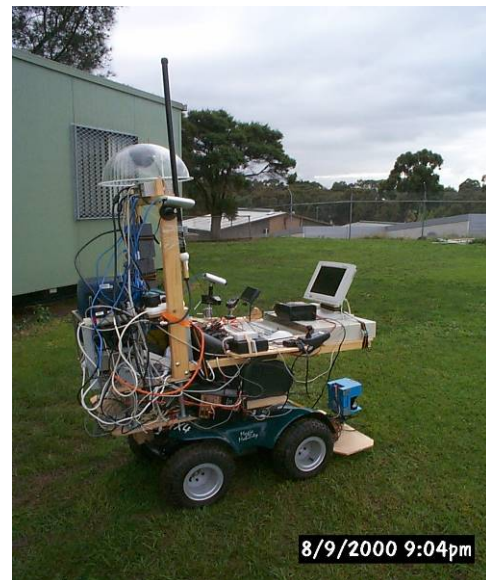


Figure 3: User-Adaptive Robotic Wheel Chair.

Figure 4 shows a fully autonomous rough terrain tracked vehicle (Jarvis, 1997) which uses GPS localisation, laser range finder obstacle mapping, and Distance Transform (Jarvis, 1994) path planning. Only the goal location is indicated on an environmental map which is populated with obstacles as they are discovered by on-board sensors. Collision-free optimal paths to the goal are recomputed on a fairly continuous basis.



Figure 4: Autonomous Rough Terrain Tracked Robotic Vehicle.

Figure 5 shows an indoor fully autonomous robot (Jarvis, 1997) which can map its obstacle strewn environment and continuously replan its paths to a nominated goal. Localisation is achieved using a Denning laser bar code reading localiser with bar code beacons placed at known locations in the floor plan.



Figure 5: Autonomous Indoor Beacon Localised Robot.

Figure 6(a) shows a teleoperated boom lift (Jarvis, 2006) and Figure 6(b) a teleoperated fire tanker (Jarvis, 2008). Teleoperation is supported by video cameras, GPS, laser range finders and pitch/tilt sensors. Figure 7(a and b) shows some of the types of environmental mapping data available to the teleoperator.



Figure 6(a): Teleoperated Boom Truck.



Figure 6(b): Teleoperated Fire Tanker.

Figure 8 illustrates a very recent experiment where detailed off-line environmental mapping (Jarvis, 2007) was carried out using a Riegl LMS-Z420i laser scanner provided with registered colour imaging capabilities. Navigation tasks in the “cyberspace” created by this environmental data could be replicated in the real physical space from which the model data was acquired using a physical robot. The robot could localise itself using panoramic images which were matched against images extracted from the pre-scanned “cyberspace” data. This approach does rely on the prior collection of detailed environmental data but this process need only be done once. The generality of this approach

and the ease of extension into 3D highly recommends it for situations where prior data collection can be justified e.g. in public spaces, malls, air terminals etc.



Figure 7(a): Colour Rendered 3D Environmental Data.

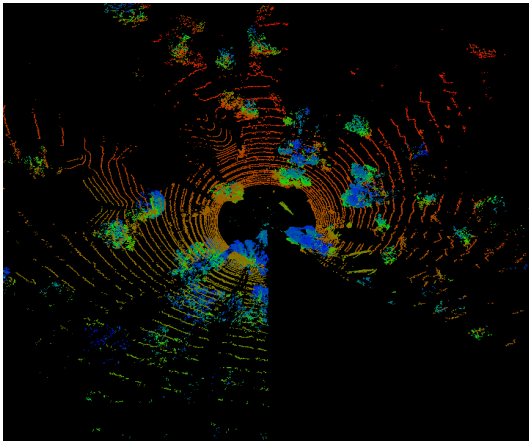


Figure 7(b): Plan View of 3D Laser Range Scan.

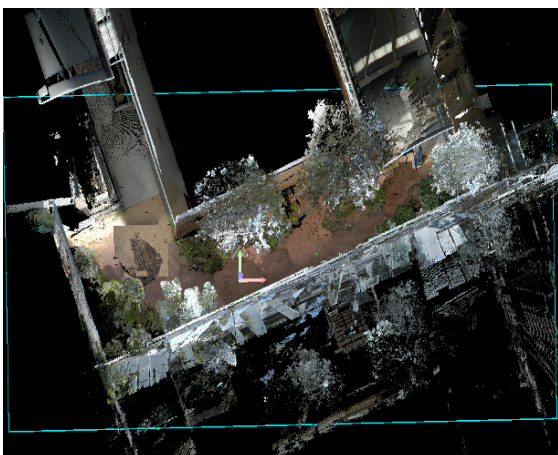


Figure 8: Dense Laser/ Colour Vision Environmental Data Collected Off-Line.

6 DISCUSSION AND CONCLUSIONS

The idea that robot navigation solutions should be flexible to span pure teleoperation to fully autonomous operations with a three level control strategy and smooth variations of human intervention is a very practical one, since it can be adapted to individual situations and changes of circumstances at will. Also, as new methods, improved sensor instrumentation and increased affordable computation come to hand various aspects of this approach can be tuned so that the balance of control may shift but the continuum maintained.

As the quality of SLAM solutions improve, human/machine interfaces evolve, swarms replace individual robots on distributed tasks, questions of risk and responsibilities resolved and co-operative interplays with human agencies developed, maintaining the type of flexibility promoted by this paper becomes even more reasonable and practical, particularly as the inclusion of this kind of flexibility does not impose any great additional cost and provides a graceful degradation path.

In conclusion, this paper has advocated a flexible approach to the selection of robot navigation modalities to suit particular circumstances relating to knowledge, risk, complexity, efficiency and reliability factors so that working solutions to important robot application domains can be applied now and improved in the future without the stagnation which may result from a more rigid approach.

REFERENCES

- Leonard, J. J., and Durrant-Whyte, H. F. (1991) Simultaneous map building and localization for an autonomous mobile robot. In *IROS-91* (Osaka, Japan), pp. 1442-1447.
- Jarvis, R.A. (1993) A Selective Survey of Localisation Methodology for Autonomous Mobile Robot Navigation, accepted for presentation at the Robots for Competitive Industry Conference, July 14-16, Brisbane Australia, pp. 310-317.
- Jarvis, R.A. (1994) On Distance Transform Based Collision-Free Path Planning for Robot Navigation in Known, Unknown and Time-Varying Environments, invited chapter for a book entitled 'Advanced Mobile Robots' edited by Professor Yuan F. Zang World Scientific Publishing Co. Pty. Ltd. pp. 3-31.

- Jarvis, R.A.(2001) A Vision Assisted Semi-Autonomous user-Adaptive Rough Terrain Wheelchair, Proc. 4th Asian Conference on Robotics and its Applications, 6 - 8th June, Singapore, pp.45-50.
- Jarvis, R.A.(1997)An Autonomous Heavy Duty Outdoor Robotic Tracked Vehicle, Proc. International Conference on Intelligent Robots and Systems, Grenoble, France, Sept. 8-12, pp.352-359.
- Jarvis, R.A. (1997) Etherbot - An Autonomous Mobile Robot on a Local Area Network Radio Tether, Proc. Fifth International Symposium on Experimental Robotics, Barcelona, Catalonia, June 15-18, pp.151-163.
- Jarvis, R.(2006) Four Wheel Drive Boom Lift Robot for Bush Fire Fighting, 10th International Symposium on Experimental Robotics (ISER 2006), July 6-10, Rio de Janeiro, Brazil.
- Jarvis,R.A.,(2008) Sensor Rich Teleoperation Mode Robotic Bush Fire Fighting, International Advanced Robotics Program/EURON WS RISE'2008,International Workshop on Robotics in Risky Interventions and Environmental Surveillance,7th to 8th Jan. Benicassim, Spain.
- Jarvis,R.A., Ho,Ngia and Byrne, J.B, (2007)Autonomous Robot navigation in Cyber and Real Worlds, Accepted for presentation, CyberWorlds 2007, Hanover, Germany, Oct. 24th to 27th.

A WAIT-FREE REALTIME SYSTEM FOR OPTIMAL DISTRIBUTION OF VISION TASKS ON MULTICORE ARCHITECTURES

Thomas Müller, Pujan Ziaie and Alois Knoll

*Robotics and Embedded Systems Group, Technische Universität München
Boltzmannstr. 3, 85748 Garching, Germany
muelleth@cs.tum.edu, ziaie@cs.tum.edu, knoll@cs.tum.edu*

Keywords: Robot Vision, Multithreaded Realtime System, Asynchronous Data Management, Interpretation-Based Preselection, Optimal-Backoff Scheduling.

Abstract: As multicore PCs begin to get the standard, it becomes increasingly important to utilize these resources. Thus we present a multithreaded realtime vision system, which distributes tasks to given resources on a single off-the-shelf multicore PC, applying an optimal-backoff scheduling strategy. Making use of an asynchronous data management mechanism, the system also shows non-blocking and wait-free behaviour, while data access itself is randomized, but weighted. Furthermore, we introduce the top-down concept of *Interpretation-Based Preselection* in order to enhance data retrieval and a tracking based data storage optimization. On the performance side we prove that functional decomposition and discrete data partitioning result in an almost linear speed-up due to excellent load balancing with concurrent function- and data-domain parallelization.

1 INTRODUCTION

The multicore integration of off-the-shelf PCs is clearly observable with recent hardware development. Correlated to this, algorithms have to be developed that exploit parallel resources and generate the expected proportional speed-up with the number of cores. A computer vision (CV) system is a perfect prove of the algorithmic concept we present in this paper, because it requires high computational effort and realtime performance. The vision system is part of the JAST (“Joint Action Science and Technology”) human-robot dialog system. The overall goal of the JAST project is to investigate the cognitive and communicative aspects of jointly-acting agents, both human and artificial (Rickert et al., 2007).

Vision processing in the JAST system (Figure 1) is performed on the output of a single camera, which is installed directly above the table looking downward to take images of the scene. The camera provides an image stream of 7 frames per second at a resolution of 1024×768 pixels. The output of the vision process (recognized objects, gestures, and parts of the robot) has to be sent to a multimodal fusion component, where it is combined with spoken input from the user to produce combined hypotheses represent-



Figure 1: The JAST human-robot interaction system.

ing the user’s requests.

According to our research field of interest, the vision system is required to publish object, gesture, and robot recognition results simultaneously and in realtime, although continuous realtime result computation is not feasible. Therefore the JAST vision

setup is well suited for investigations on parallelization techniques and data flow coordination. We propose a multithreaded vision system based on a high level of abstraction from hardware, operating system, and even lower level vision tasks like morphological operations. This minimizes the overhead for communicational tasks, as the amount of data transferred decreases in an abstract representation. Furthermore, the scalability of the system with integration of multiple cores can be examined soundly by connecting different machines to the JAST system, each running a copy of the vision system (details in Section 4).

2 PARALLEL COMPUTATION

On an abstract level two major parallelization scenarios may be identified: distribution of processing tasks on multiple machines on one side and distribution of tasks on a single machine with multiple processors and / or cores on the other.

Many approaches employing the distributed scenario have been proposed, see (Choudhary and Patel, 1990) for an overview regarding CV or (Wallace et al., 1998) for a concrete implementation. However, with recent development in integration of multiple cores the latter scenario also becomes more relevant. Thus there is increasing demand for algorithms fully exploiting parallel resources on a single PC. This is especially the case, where computational power easily reaches the limits – e.g. in computer vision.

2.1 Communication

In parallel environments one can generally apply either synchronous or asynchronous communication strategies for data exchange between processes or threads. Though being robust, due to its blocking nature a synchronous approach can cause problems especially for realtime systems where immediate responses have to be guaranteed. For this case asynchronous *non-blocking* communication mechanisms (ACM) have been proposed. With ACMs information is dropped when capacities exceed – which is acceptable as long as the system does not block. Non-blocking algorithms can be distinguished into being *lock-free* and *wait-free* (Sundell and Tsigas, 2003). Lock-free implementations guarantee at least one process to continue at any time (with the risk of starvation). Wait-free implementations avoid starvation as they guarantee completion of a task in a limited number of steps (Herlihy, 1991).

According to (Simpson, 2003), ACMs can be classified based on the destructiveness of data access. The

classification of ACM protocols by (Yakovlev et al., 2001) distinguishes data access with respect to their overwriting and re-reading permission. One can find manifold implementations of ACMs regarding each of these classification schemes. Some common implementations, e.g. from (Sundell and Tsigas, 2003) use lock-free priority queues or employ FIFO-buffers (Matsuda et al., 2004).

2.2 Parallelization Techniques

According to (Culler et al., 1999) we have to distinguish parallelization techniques by means of *data-domain* or *function-domain*. With function-domain parallelization the overall computation process is divided into stages and each thread works on a separate stage. In contrast to this, with data-domain parallelization data is partitioned and each partition requires the same computation performed by equally designed threads (Chen et al., 2007). This distinction may be correct and worthy for low level vision tasks like edge detection, but this paper will show, that on a higher level a carefully modeled CV system does not require this distinction. Moreover a combined approach can be derived and, on the basis of an asynchronous data management, a system implementing both aspects can perform very well in practice.

Aiming this goal, we first have to deliberately design anchor points for distributed computation. Also, the level of abstraction considering computational tasks matters in terms of parallelization. In order to avoid unnecessary overhead regarding communication and take full advantage of the multicore environment, we decided to model concurrent computation on a high level of abstraction. Therefore, we do not intend to parallelize primitive control-structures – like *for-loops* – specific to a programming language. Instead we try to identify major and subsequently minor tasks of computation (see Figure 2).

For function-domain parallelization we assume, that the division into well-defined functional submodules is feasible. In the processing layer of the proposed CV system this is obviously the case, as we can identify three major functional stages: *Preprocessing*, *Analysis and Interpretation* and *Postprocessing*. Further refinement divides these stages into subtasks. Modules implementing a task independently pick a data partition (also called data item below), analyze it and write it back. In case new items are created within the analysis, these are also stored in the corresponding data management queue (see Section 3).

As the recognition process is decomposable in the function-domain, we now have to achieve data-domain parallelization in order to prove our claim.

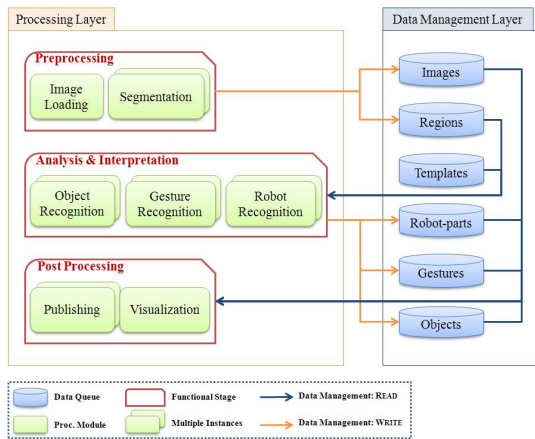


Figure 2: Architectural overview.

Hence we have to specify the functional tasks according to the need of multiple instantiation of the processing modules. We therefore derive the following approach from the non-blocking paradigm of ACMs: as we want to publish cyclicly in realtime, we rather publish incomplete analysis results of a scene than waiting for a complete analysis that would block the system meanwhile. This allows multiple concurrent module instances for the analysis of data items as long as the data management is implemented threadsafe (see Section 3). According to (Chen et al., 2007) we are thus able to implement data-domain parallelization, which is the second part of our claim.

2.3 Scheduling

There is one catch in such an implementation of the ACM: we risk that a module requests certain data from the data management, which is not available at the moment. In this case the data management delivers a NULL-data item, so modules have to deal with these items as well.

Therefore we propose an algorithm which, whenever a NULL-item is received, tries to suspend module instances for an optimal amount of time, until a correct data item is expected to be delivered again. An incremental back-off time $b(c)$ may be calculated as follows:

$$b(c) = \min \left(c \cdot i, \left(\frac{a \cdot j}{n} \right) \right) \quad (1)$$

In (1) the parameter c denotes the counter for the number of tries since the last correct data item has been received by the module, i denotes the predefined back-off increment in milliseconds, a is the maximum age of a data item until it is deleted, j the number of module instances operating on the same task and n the current number of items matching the request. If

a NULL-data item is retrieved, c is incremented and the module is immediately suspended for a time $b(c)$ again. In case a correct item could be delivered, c is reset to 0 and the item is processed.

The back-off strategy tries to optimally calculate suspension periods for instances not needed at the moment, but at the same time to provide an instance whenever needed. The first argument of \min calculates an incremental amount of time for the module instance to sleep and the second argument represents the expected mean time until the next correct data item can be delivered. This value is then used as the maximum amount of time to suspend a module instance.

3 DATA MANAGEMENT

Implementing an adequate data access strategy for concurrent requests is crucial for the proposed system. The strategy has to ensure integrity and consistency of data and as well provide error management policies. One also has to consider prioritization whenever a module requests to write while another simultaneously wants to read data from or write data to the storage. Another important point is the deletion of data items when they expire.

Considering modularity, we organize data access in a data management layer (right part of Figure 2). A natural approach for the implementation is based on the *Singleton* design pattern (Gamma et al., 1998). Singleton implementations only provide a single instance of an object to the overall system, so in our case any request from an analysis module must call the single instance of the data management (DM). Here, derived from common standards (Message Passing Interface Forum, 1995), data items are managed in limited-size priority-queues.

Error handling in the DM layer can be implemented straight forward, as the layer simply delivers NULL-data items whenever an erroneous request was received, a queue was empty or no suitable data item could be found. The error handling approach utilizing NULL-data items is wait-free, because it completes in a limited number of steps.

Organizing the single instance in a threadsafe manner concerning read and write accesses ensures integrity and consistency. In order to achieve this, the DM module is organized as a bundle of queues, each queue for a different type of data item (see Figure 2).

3.1 Data Access

Threadsafe concurrent data access is realized by encapsulating synchronization. Concerning ACMs, the

CV system proposed here implements a *Pool-ACM* in either classification scheme mentioned in Section 2.1. Regarding the Simpson classification, as we do have non-destructive read operations, but write operations include deletion of items, and respectively regarding the Yakovlev classification, as we allow overwriting in a write operation and do not delete items when reading them from the storage.

Concretely, an instance of a processing module sends a request for storage or retrieval of a data item of a certain kind by calling one of the DM operations provided to the processing layer:

```
write<Queue>(Item):void
read<Queue>():Item
```

The retrieval strategy selects a data item to deliver according to the evaluation of a stochastical function. The function is based on the assumption that a data item (re-)detected in the near past must be prioritized to one that last occurred many cycles ago – as it may have already disappeared or removed. Since each item in a queue Q has a timestamp, we weigh the items $i \in Q$ according to their age $a_i = now - timestamp(i)$ such that the weight increases, the younger items are:

$$\forall i \in Q : w_i = 1 - \frac{a_i}{maxage} \quad (2)$$

A new queue of pointers to data items from the original queue is built afterwards. The new queue, on which the actual retrieval operation is performed, is filled with at least one pointer to each data item. In fact, according to the weight w_i of an item i , a number of duplicates d_i of each pointer is pushed to the queue:

$$\forall i \in Q : d_i = \frac{1}{\operatorname{argmin}_{j \in Q}(w_j)} \cdot w_i \quad (3)$$

Subsequently the random selection on the pointer queue is performed where more recent items are prioritized automatically as more pointers to the corresponding data-items exist.

3.2 Locking

Before applying the weight to the items of a queue, we have to exclude elements that match the precondition described below. As an item cannot be altered by two processing modules concurrently, we introduce a locking-mechanism for items. Nevertheless the “non-blocking” nature of data access can still be guaranteed due to the error handling approach described earlier. Before a data item is delivered to the processing layer, the state of the item is changed to *locked*. Locked items are not allowed to be delivered to any other instance and so are excluded from the weighting step.

Releasing the lock is in responsibility of the module processing the item.

Another important problem to discuss is the behaviour of the system in case of concurrent WRITE or READ operations concerning a specific queue. Concurrent READ operations are allowed at any time, but in case a WRITE operation is requested all retrieval requests and concurrent WRITE requests must be blocked meanwhile. Therefore the system has to implement a mechanism utilizing cascaded mutual exclusions.

Again a single operation may be blocked, but the overall system is not. If a mutex can not be acquired at the moment, in case of a READ operation a NULL-item is delivered and in case of a WRITE operation no operation is executed. This behaviour is conform to the definition of an asynchronous non-blocking algorithm, as it is wait-free.

3.3 Enhancements

In order to enhance performance of READ operations, we introduce the concept of *Interpretation-Based Preselection*. We assume that certain data items are not relevant for dedicated tasks. For example a gesture-recognition module could only be interested in a region, that enters the scene from the bottom (Ziaie et al., 2008) or a visualization module might only display objects from within the last 100ms, but skipping gestures totally.

In order to completely leave the relevance decision to the processing modules, we propose a mechanism evaluating a predicate, that is passed within the request. According to the predicate the exclusion step before weighting a queue’s items is adapted: now not only locked, but also items that do not match the predicate are removed. Thus the search space for retrieval can be restricted, but the non-deterministic selection algorithm can still be applied. We now extend the trivial retrieval definition from Section 3.1 to the following:

```
read<Queue>(Predicate):Item
```

Predicate is a non-empty binary predicate that evaluates to True or False on each data item of the specified queue. Processing modules are allowed to use item attributes for the implementation of their own predicates. For sophisticated predicate designs some items provide state attributes for tracking or attributes indicating the status of the analysis (*analyzedBy*<module>). We call these attributes *Priority Attributes*.

An enhancement strategy for WRITE operations can also be implemented by our data management module. Considering that data items in a queue are

timed, it is possible to *track* them from one cycle to the following. Therefore we define a compare-method that is applied automatically on a storage request. The method evaluates symbolic or meta attributes like classification, color, approximate position, number of points or width and height. Whenever the DM module receives a storage request for a formerly recognized item, only necessary attributes are updated, all priority attributes (especially the unique id) instead are kept. For example, considering an item fixed and fully analyzed, the existing item just gets all non-priority attributes (such as the timestamp, position, etc.) updated, but the updated item is not marked for analysis again.

4 RESULTS AND CONCLUSIONS

For the evaluation of the system a dual core Intel © ,Pentium IV system and a quad core Intel © ,Xeon system were utilized. For comparison the hardware configurations using a sequential version of the system are also shown. We used a sample video with a resolution of 1024×768 and a duration of 30 seconds at a sampling rate of 7 frames per second. The results refer to the analysis without data maintenance enhancements and postprocessing switched off.

Performance results shown in the tables below are only approximate values due to high dependency on the scenes that have to be analyzed. The more objects exist, and the more complex objects get (in the JAST-project also object assemblies are to be analyzed) the longer the analysis takes. In case there are very few objects or the scene remains static, overall analysis is possibly performed in realtime. This would be contradicting on of our preconditions from Section 1, so for our evaluation video we feed the system with dynamic input data, like humans continuously moving objects on the table and the robot picking pieces.

In Table 1 the first column describes the hardware configuration, the second column shows the total system load and the third column, the (mean) LOAD RATIO, weighs the core with highest against the core with the lowest load.

Table 1: Total core utilization.

CONFIGURATION	CPU %	LOAD RATIO
Dual Core (seq.)	52.12	8.20
Dual Core (parallel)	84.13	1.03
Quad Core (seq.)	27.55	36.32
Quad Core (parallel)	51.63	1.07

The values shown in the table were computed from 5-10 averaged samples, each taken with `mpstat`

over a period of five seconds. For example a mean ratio of 36.32 on the quad core is caused by an average load of 93.84% on the core with highest load compared to only 2.58% on the core with lowest load. The total utilization in this configuration clearly shows that de facto only one core is used for processing while the others remain idle. In contrast to this, one can see an almost optimal distribution in the parallel scenarios with a load ratio of around 1.0.

Table 2 shows the processing performance of the hardware configurations described above. Now the reason for the quad core parallel configuration only having a total load of 51.63 % becomes clear: there is simply nothing to do for the machine as the input video is only sampled at 7 frames per second.

Table 2: Performance of Processing.

CONFIGURATION	TIME	FREQUENCY
Dual Core (seq.)	228.7s	0.92fps
Dual Core (parallel)	30.0s	3.57fps
Quad Core (seq.)	196.6s	1.07fps
Quad Core (parallel)	30.0s	6.95fps

Processing with the sequential version of the system is slightly faster on the quad core compared to the dual core machine due to internal OpenCV parallelization and scheduling of the operating system. But still it is only capable of processing the video in ≥ 3 minutes. Regarding this, another important property of the asynchronous parallel version becomes clear: processing a 30s-video only takes 30 seconds. This can be achieved because of the non-blocking behaviour. In case computing power exceeds (see second row of Table 2) the asynchronous implementation drops frames, regions and objects from data queues in order to keep the system from blocking. We find that the system still reaches the desired realtime publishing frequency, but the results published are not complete. In fact we see, that it takes a few cycles until each region extracted from a frame is analyzed and results are present.

Normally this does not influence the result, as items can be tracked. But in case of quickly moving objects, it remains as a drawback, because the simple feature-based tracking method applied in the current system often fails to map these objects correctly in a sequence of frames. Consequently, the system assumes items having appeared and begins the analysis: the new items are locked (although they are just duplicates of existing ones) and computing power is wasted. This problem particularly occurs for quick hand movements. The worst case would be a moving hand shortly occluding formerly recognized objects, as both the hand and the objects are probably

lost and their regions need to be redetected and reanalyzed completely.

Figure 3 shows a performance estimation for the analysis frequency in two input scenes. As we expect, the results show, that parallelizing in the data domain produces almost linear performance gain with the number of processors. This can be achieved, because heavy computing is mainly done within the analysis and interpretation stage, where tasks can be distributed very well. In Figure 3 the measuring points for one core are inferred from performance of sequential version, as we have seen in Table 1 that only one core is used there.

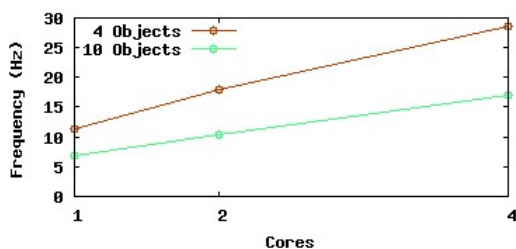


Figure 3: System performance with data domain parallelization.

Still to mention is that with the data management enhancement from Section 3.3 the performance even in a sequential version of the system improves up to 25 Hz as long as the extracted regions can be tracked. When the scene changes, computational effort is needed, so the performance decreases. Here the advantage of the multithreaded system becomes clear: due to function domain parallelization and non-blocking behaviour the system still publishes in real-time, although the results may be incomplete.

A further factor influencing the system performance is the system configuration. The vision system configuration can be customized via an XML file. Here one can specify the number of module instances. This corresponds to a prioritization within the data domain: one could for example start a larger number of objectrecognition modules while on the other hand just starting one or two gesturerecognition modules. Due to the scheduling strategy of the operating system, the objectrecognition would be prioritized in this case.

ACKNOWLEDGEMENTS

This research was supported by the EU project JAST (FP6-003747-IP), <http://www.euprojects-jast.net/>.

REFERENCES

- Chen, T. P., Budnikov, D., Hughes, C. J., and Chen, Y.-K. (2007). Computer vision on multi-core processors: Articulated body tracking. pages 1862–1865. Intel Corporation, IEEE ICME.
- Choudhary, A. N. and Patel, J. H. (1990). *Parallel Architectures and Algorithms for Integrated Vision Systems*. Kluwer.
- Culler, D. E., Singh, J. P., and Gupta, A. (1999). *Parallel Computer Architecture: A Hardware/Software Approach*. Morgan Kaufmann Publishers.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1998). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series.
- Herlihy, M. (1991). Wait-free synchronization. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 13(1):124–149.
- Matsuda, M., Kudoh, T., Tazuka, H., and Ishikawa, Y. (2004). The design and implementation of an asynchronous communication mechanism for the mpi communication model. pages 13–22. IEEE ICCS.
- Message Passing Interface Forum (1995). MPI, A Message-Passing Interface Standard. Technical report, University of Tennessee, Knoxville, Tennessee.
- Rickert, M., Foster, M. E., Giuliani, M., By, T., Panin, G., and Knoll, A. (2007). Integrating language, vision, and action for human robot dialog systems. Proc. ICMI.
- Simpson, H. R. (2003). Protocols for process interaction. volume 150, pages 157–182. IEE Proceedings on Computers and Digital Techniques.
- Sundell, H. and Tsigas, P. (2003). Fast and lock-free concurrent priority queues for multi-thread systems. Int. Parallel and Distributed Proc. Symp.
- Wallace, A. M., Michaelson, G. J., Scaife, N., and Austin, W. J. (1998). A dual source, parallel architecture for computer vision. *The Journal of Supercomputing*, 12(1-2):37–56.
- Yakovlev, A., Xia, F., and Shang, D. (2001). Synthesis and implementation of a signal-type asynchronous data communication mechanism. pages 127–136. Int. Symp. on Advanced Research in Async. Circuits and Systems.
- Ziaie, P., Müller, T., Foster, M. E., and Knoll, A. (2008). A naïve bayes classifier with distance weighting for hand-gesture recognition. CSICC.

FROM CAD MODEL TO HUMAN-SCALE MULTIMODAL INTERACTION WITH VIRTUAL MOCK-UP

An Automotive Application

Damien Chamaret, Paul Richard and Jean-Louis Ferrier

Laboratoire d'Ingénierie des Systèmes Automatisés (LISA), Université d'Angers

62 Avenue Notre Dame du Lac, 49000 Angers, France

damien.chamaret@univ-angers.fr, paul.richard@istia.univ-angers.fr, ferrier@istia.univ-angers.fr

Keywords: Virtual environment, human-scale, multimodal interaction, CAD model, virtual mock-up.

Abstract: This paper presents and validates a new methodology for the efficient integration of CAD models in a physical-based virtual reality simulation. User interacts with virtual mock-up using a string-based haptic interface that may provides haptic sensation to both hands in a large workspace. Visual and tactile displays provide users with sensory feedback and improve both user performance and immersion. Stereoscopic images are displayed on a 2m x 2.5m retro-projected screen and viewed using polarized glasses. The proposed methodology implemented in a low-cost system, has been tested with an automotive application task. However, the presented approach is general enough to be applicable to a large variety of industrial applications.

1 INTRODUCTION

Most research on virtual environments dedicated to Computer Aided Design (CAD) application are confronted to difficult problems related for instance to real-time 3D simulation including physics, integration of multisensorial feedbacks, etc. Another problem to overcome is the transformation CAD models to virtual reality (VR) models.

In this paper we present and validate a new methodology for an efficient integration of CAD models in a physical-based virtual reality simulation that provides the user with multimodal feedback. Haptic interaction is based on the SPIDAR system illustrated in Figure 1 (Bouguila et al. 2000). The methodology has been tested with an automotive application task. However, the presented approach is general enough to be applicable to other tasks and industrial applications requiring realistic interaction.

2 RELATED WORK

2.1 Visuo-haptic VR Configuration

Projection-based Virtual Environments such as CAVEs™ (Cruz-Neira et al., 1993) Workbenches

(Krueger and Froehlich, 1994) or immersive wall (Richard et al., 2006) are the most popular VR configurations. They provide a large number of performance/immersion factors like stereoscopic visualization, large screens, large manipulation space, etc. However, adding force feedback to these configurations without degrading their performance/immersion factors is not an easy task.

Most general purpose haptic devices, like the PHANToM (Massie and Salisbury, 1994) are often used with desktop visualization configurations. Most of the time, they are not able to adapt to VR configurations, leading to a degradation of some of the performance/immersion factors of the VR configuration.

Some general purpose haptic systems have been integrated within large screen projection-based VR configurations (Brederson et al., 2000), (Grant and A. Helser, 1998), (Garrec et al., 2004).

The only large screen projection based VR configurations equipped with non-intrusive haptic system involves the SPIDAR system (Bouguila et al., 2000), (Tarrin et al., 2003), (Ishii and Sato, 1994) (Richard et al., 2006).

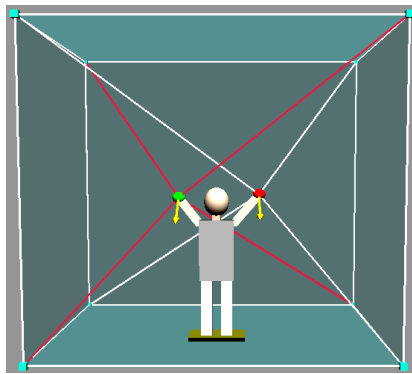


Figure 1: Schematics of the human-scale SPIDAR system. This non-intrusive haptic device provides force feedback to user's both hands while moving in a large-scale workspace.

2.2 Grasp Feedback

Grasp feedback includes both tactile and shape feedback. Haptic systems exist for both but they are different and rarely integrated.

The best known solution for providing a realistic grasp feedback consists of using props. Props are physical objects held in hand by the user. Props have been proposed for tasks such as application control (Coquillart and Wesche, 1999), 3D objects manipulation (Hinckley, 1994), (Tarrin et al., 2003), and design. Several psychophysics experiments demonstrate the benefits of props (Hinckley, 1994). Props provide stable grasp feedback, intuitive manipulation as well as realistic shape and texture rendering.

Props do not allow sensation of the collision with a surface touched by the prop itself. Combining props with force feedback is again a difficult task because most force feedback systems can't attach props in a flexible way.

The next section of the paper presents a CAD to VR methodology. Section 4 describes the developed prop-based stringed haptic configuration. An industrial application is presented in Section 5. Section 6 concludes the paper and describes future work.

3 CAD TO VR METHODOLOGY

The CAD to VR methodology is illustrated in Figure 2. Our methodology involves different steps such as model tessellation (1), model integration (2-3) and sensorial feedback (4-5-6). The CAD to VR model transformation is illustrated in Figure 3.

3.1 Model Tessellation (1)

The first step was to choose an appropriate common exchange file format between standard CAD software (such as CATIA) and 3D general purpose modelling software (such as 3D Studio Max). The tessellation procedure consists in decreasing the number of faces without degrading the 3D shape of the model.

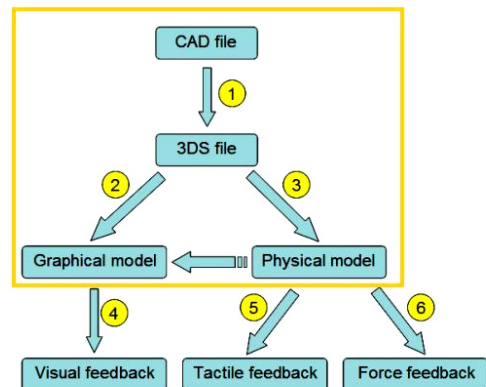


Figure 2: Schematics of the CAD to VR model transformation.

3.2 Model Integration (2-3)

Many loaders may be used to integrate 3D models in a real time C/C++ 3D application. However, this leads to graphic simulation in which the virtual objects do not have physical properties. In order to give physical properties and behaviour to the model, we developed a single procedure that allows to automatically obtain both graphical and physical models of loaded objects. Moreover, the physical model on which is based the real time simulation, exactly corresponds to the graphical one.

The physical model is built using PhysX™ an efficient well-known open source physic engine from AGEIA (<http://www.ageia.com>).

3.3 Sensorial Feedbacks (4-5-6)

In order to increase both realism of the simulation and operator performance during interaction with the virtual mock-up, different sensorial feedbacks are provided. Simulated forces are calculated by the physic engine and displayed on user hand using the SPIDAR system.

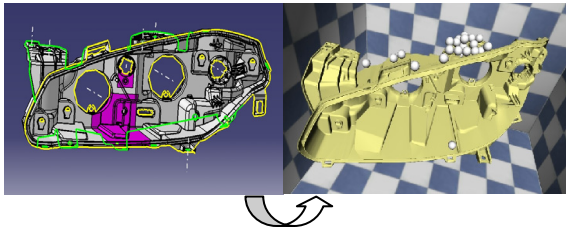


Figure 3: illustration of a CAD model transformation to a physicalized VR model.

4 PROP-BASED STRINGED HAPTIC INTERACTION

In the context of the automotive application described in the following section, a prop has been integrated into the SPIDAR system in order to provide both realistic and low-cost grasp sensation while performing the task. Figure 4 shows the attachments of the prop to the SPIDAR (Figure 4a). As shown in Figure 4b, the user grasps the prop (a real car lamp is integrated into a plastic part). In order to increase accuracy, both position and orientation of the prop are obtained using a Patriot 3D tracking system (<http://www.polhemus.com>).

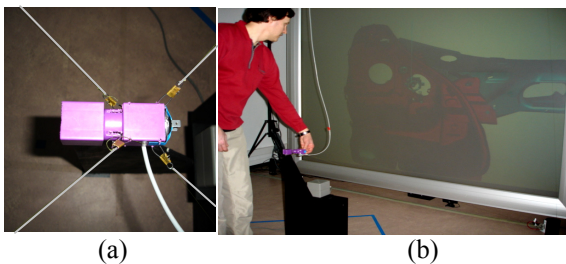


Figure 4: Top view of the prop attached to the SPIDAR system (a), grasping of the prop by a user (b).

5 AUTOMOTIVE APPLICATION

Our VR human-scale platform opens the door to many CAD applications requiring realistic integration modalities such as visualization, audio, force and tactile feedback. One such application, from the automotive industry, is described in this section. It concerns accessibility and maintenance of car lamps.

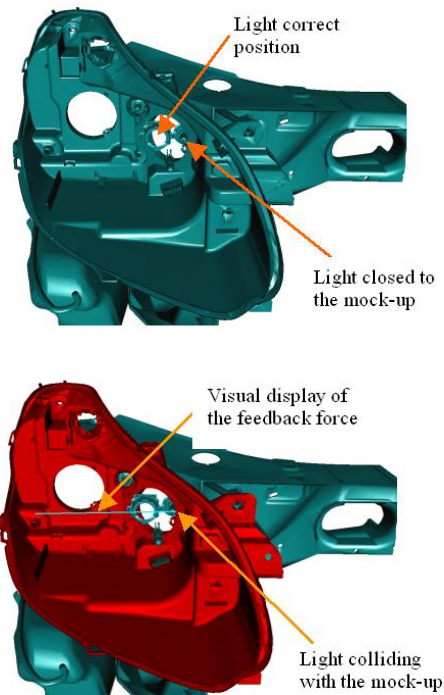


Figure 5: Illustration of the final stage of the task: the application provides both haptic and visual force feedback to the operator.

5.1 Description of the Application

During the conception stage, car designers have to make sure that anyone will easily be able to achieve maintenance task concerning car lamps. In this context, special attention has to be paid to the following aspects:

- Accessibility: to be able to reach and pick-up car lamps,
- Replacement: to be able to remove broken lamps and replace them by new ones.

Currently, the only solution is to build a mock-up of the car. The process is of course slow and expensive. A cheaper and faster solution consists in realizing the tests on virtual mock-ups. An additional advantage is that it can be done earlier in the conception process, which eases modifications. Our methodology associated to the human-scale haptic virtual environment highly contributes to the widespread use of virtual mock-up in a realistic simulation.

This methodology has been validated in the previously mentioned context (car lamps maintenance). The Figure 5 shows screenshots of the virtual mock-up during the maintenance task. The

final stage of this task involves the correct placement of a car lamp. As illustrated on the bottom screenshot, both collision detection and force feedback are visually displayed respectively using a red colour and clear blue line (orientation of the force).

5.2 Hardware and Software Architecture

As opposed to most of the existing virtual reality human-scale platforms that are based on clusters, our hardware architecture is based on only one Personal Computer (bi-Xeon 5150, 4Go RAM and 8800 GTX Graphic board).

The frame rate is however, in the described application (600 000 Faces), maintained to about 30 frames per second. Thus, the use of a physical Processing Unit is not necessary.

6 CONCLUSIONS AND FUTURE WORK

We presented and validated a new methodology for the efficient integration of CAD models in a physical-based virtual reality simulation. User interacts with virtual mock-up using a string-based haptic interface that may provides haptic sensation two both hands in a large workspace. Visual and haptic displays provide users with sensory feedback and improve both user performance and immersion. Stereoscopic images are displayed on a 2m x 2.5m retro-projected screen and viewed using polarized glasses. The proposed methodology has been tested with an automotive application task. However, the presented approach is general enough to be applicable to other tasks and industrial applications.

In the next future we plan to add a virtual hand with physical properties to allow dexterous manipulation of 3D objects. We will also replace the magnetic tracking system by an optical MOCAP solution. We will also use our methodology for other CAD applications.

ACKNOWLEDGEMENTS

The authors would like to thanks the representative of Valeo Lighting System (Angers - france) involved in the project, especially Sébastien DENIS and Xavier GALLARD.

REFERENCES

- L. Bouguila, M. Ishii and M. Sato, "Multi-Modal Haptic Device for Large-Scale Virtual Environment", 8 th ACM International Conference on Multimedia, 2000.
- C. Cruz-Neira, D. Sandin, and T. Defanti, "Surround Screen Projection Based Virtual Reality". Proceedings of SIGGRAPH, 1993.
- W. Krueger and B. Froehlich, "The Responsive Workbench", IEEE Computer Graphics and Applications, 1994.
- P. Richard, D. Chamaret, F.-X. Inglese, P. Lucidarme J.-L. Ferrier: Human-Scale Haptic Virtual Environment for Product Design: Effect of Sensory Substitution. International Journal of Virtual Reality, 2006, 5(2): pp. 37-44.
- T. H. Massie and J. K. Salisbury, "The PHANTOM Haptic Interface: A Device for Probing Virtual Objects", Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems , 1994.
- J.D. Brederson, M. Ikits, C.R. Johnson, C.D. Hansen, and J.M. Hollerbach, "The Visual Haptic Workbench", Proceedings of the Fifth PHANTOM Users Group Workshop, 2000.
- B. Grant and A. Helser, "Adding Force Display to a Stereoscopic Head-Tracked Projection Display", VRAIS, 1998.
- P. Garrec, J.P. Friconneau and F. Louveau, "Virtuose 6d: A New Industrial Master Arm using Innovative Ball-Screw Actuators", Proc. of International Symposium on Robotics, 2004.
- N. Tarrin, S. Coquillart, S. Hasegawa, L. Bouguila, and M. Sato, "The Stringed Haptic Workbench : a New Haptic Workbench Solution", EUROGRAPHICS Proceedings, 2003.
- M. Ishii and M. Sato. "A 3d Spatial Interface Device using Tensed Strings", Presence, 3(1), 1994.
- S. Coquillart and G. Wesche, "The Virtual Palette and the Virtual Remote Control Panel: A Device and an Interaction Paradigm for the Responsive Workbench. Proc IEEE VR, 1999.
- K. Hinckley, R. Pausch, J. Goble, and N. Kassell, "Passive Real-World Interface Props for Neurosurgical Visualization", CHI, 1994.

DETECTING TRANSIENT WEATHER PHENOMENA ON MARS

An Overview of the Design and Calibration of Multiparametric Detection Algorithms for the REMS/MSL Mission

J. Verdasca, J.-A. Manfredi and V. Peinado
*Centro de Astrobiología, Instituto Nacional de Técnica Aeroespacial
Ctra. de Torrejón a Ajalvir Km 4, Torrejón de Ardoz, Madrid, Spain
alvesvja@inta.es*

Keywords: Multiple-scale phenomena, change detection, autonomous decision, multiparametric algorithms.

Abstract: In its baseline operation mode, the REMS/MSL mission is not well suited to characterize transient weather phenomena evolving on multiple scales. In this paper we argue that this limitation can be overcome by enabling REMS to autonomously detect sudden and unexpected changes in the acquired data and subsequently extend the time allocated for measurements. Detection is accomplished by means of multiparametric Event-Finding Algorithms running on the REMS micro-controller. We describe the principles underlying the design of such algorithms and the methods for their calibration.

1 DESCRIPTION OF REMS

The Rover Environmental Monitoring Station (REMS) is an instrument suite to be carried aboard the Mars Science Laboratory (MSL) rover. It is being developed and integrated at the Centro de Astrobiología (CAB/(CSIC-INTA)) as a contribution of the Spanish Government. REMS comprises two booms attached to the rover mast at an angle of 120 degrees (Figure 1).

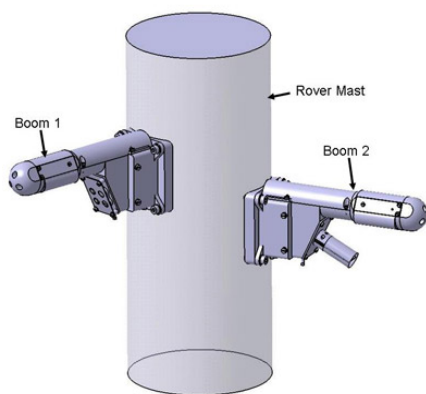


Figure 1: REMS booms with sensors.

Each of the booms carries a wind sensor and an air temperature sensor. An infrared sensor mounted on the electronics casing of boom 1, facing the ground,

will measure the brightness temperature of the martian surface. On boom 2 there is a humidity sensor (the protruding cylindrical structure on the figure) provided by the Finnish Meteorological Institute (FMI). A pressure sensor, (not shown here) also built by the FMI, is mounted below the rover deck, being in contact with the atmosphere through a small opening. Finally, an UV sensor (also not shown), located on top of the rover deck, will measure the irradiation intensity in six different bands. REMS features a small embedded computer system whose main task is to act as an Interface Control Unit but is also capable of executing simple programs.

2 THE REMS/MSL MISSION

2.1 Science Goals

The MSL mission will assess the present and past habitability of Mars. REMS will contribute by monitoring the physical processes taking place in the layer of the atmosphere closest to the surface. Scientists believe that if life exists on Mars it will most likely be found in tiny pockets of liquid water in porous rock beneath the surface. Local weather patterns likely play a fundamental role in the ability of sheltered environments to retain liquid subsurface

water. For instance, the regime of local winds may impact the cycles of freezing and thawing of permafrost soils inside craters and on the slopes of mountains, cliffs or hills. Indeed, recent images obtained from orbit show gullies and trenches that seem to have been made by water outpouring at certain localized spots and flowing down the slopes. An alternative explanation, which rules out liquid water, sustains that the gully-like features are caused by small landslides triggered by wind. Hence, one of the main goals of REMS is to shed light on the regime of local winds to enhance our ability to distinguish between features most likely due to aeolian erosion from those which could possibly have been carved by liquid water.

2.2 Limitations of REMS Baseline

The baseline currently being considered for REMS Nominal Mode (NM) operation consists in 24 sessions per sol of 5 minutes each, comprising a data acquisition cycle at a sampling rate of 1 Hz. Thus, REMS ability to characterize short range multiple-scale events is limited. In this position paper we argue that this limitation can be overcome by providing REMS with the capability to carry out Opportunistic Science. We outline the strategy we are pursuing to this aim, and lay down its scientific basis. We show how, based on a sequence of uploaded programs, the REMS computer can autonomously decide to switch from NM to the so-called Event Mode (EM) in case an anomalous pattern is detected. When functioning in EM the passage into Sleeping Mode is rescheduled to occur after a configurable amount of time, during which REMS will collect data at the nominal sampling rate.

3 OPPORTUNISTIC SCIENCE ONBOARD MARS ROVERS

3.1 MER/OASIS vs. MSL/REMS

Opportunistic science onboard the Mars Exploration Rover (MER) mission (Spirit and Opportunity) is being implemented at NASA's Jet Propulsion Laboratory (JPL) through a project known as the Onboard Autonomous Science Investigation System, (OASIS). This system identifies opportunistic targets (mainly rock samples), collects data on those targets, then analyzes them onboard to identify those which merit further investigation. It also executes onboard planning and scheduling software that can dynamically modify the projected rover traverse

plan in order to accommodate the new science tasks. Finally, the onboard computer evaluates if sufficient resources are available to meet the additional science requests (Castaño, 2007). Unlike OASIS, the REMS Opportunistic Science computer programs assume a purely passive role in planning and scheduling tasks. The REMS computer executes a sequence of instrument commands that the MSL main computer has uploaded into its flash memory. These are actions that REMS must execute at pre-scheduled times without further intervention by the rover computer – acquisition cycles, parameter configuration, housekeeping, etc. The EFA's are part of the set of programs uploaded into the REMS memory and executed once every session. Their only effect on the sequence of operations is to (eventually) extend the period of data acquisition, considering its previously allocated resources. This framework is designed not to interfere with pre-scheduled regular activities. The approach inevitably limits the scope of REMS Opportunistic Science activities but, on the other hand, improves its degree of autonomy and risk avoidance level.

3.2 Targets for REMS Opportunistic Science

The fast evolving phenomena that we wish to sample normally involve multiple timescales which cannot be fully accommodated within normal session periodicity and/or duration. These are the natural targets for Opportunistic Science actions. They can be effectively classified into i) boundary layer oscillations (BLO's), ii) dust devils and plumes, iii) microfronts, iv) local dust storms, v) orographic winds and vi) mountain waves. An algorithm to detect BLO's must be capable of recognizing fluctuations with timescales ranging from a few minutes to about one hour. Dust devils are much shorter events, generally not exceeding a few minutes. They consist on rotating columns of warm air made visible as sand and dust is sucked into the central core. Usually, convective plumes are even more ephemeral than dust-devils. Plumes are characterized by straight-line winds converging to a hot spot where the warmer air starts to rise. They do not display the rotation of the horizontal wind component that characterizes dust devils; rather, a pronounced increase in the vertical wind component is observed. Microfronts share the structure of their synoptic counterparts but arise on a local scale instead, taking a couple to a few hours to pass over a given point. The evolution of global and regional dust storms can be tracked from measurements in REMS nominal mode. However, data will be way

too sparse to provide an accurate description of short-scale, local events. The low atmospheric density of the martian atmosphere and the intense radiative processes drive robust orographic flows. Furthermore, the patterns of large-scale circulation are reinforced by the extreme regional asymmetries of the Mars elevation field. Sometimes these orographic flows appear associated with periodic buoyancy perturbations which develop when stable stratified air ascends a barrier – mountain waves. Like on Earth, these waves occur on the scale of individual mountains or chains with oscillation periods ranging from a few minutes to one sol.

4 CHANGE DETECTION ALGORITHM

4.1 Principles of EFA Design

Due to their multiple timescales, the above phenomena would be missed or insufficiently sampled should the unit operate always in NM. In the case of phenomena lasting a few dozen minutes to a few hours the measurements at high sampling rate in sequences of short acquisition cycles will mostly produce redundant data whereas the session periodicity is much too low to permit a smooth characterization. For the most brief phenomena, such as plumes and dust devils, the session duration would, in principle, be enough to fully capture the evolution of one or more of these events, and the sampling frequency adequate. However, because the unit will be sleeping for more than 90 % of the time the probability that such an event will fall within the 5 minute allocated slot is small; in most occasions REMS will miss the phenomenon entirely. Indeed, the experience of past missions, particularly Pathfinder, suggests that relying on serendipitous observations is a very poor strategy to detect dust devils. Dedicated campaigns to search for occasional phenomena are envisaged as part of an alternative 'human in the loop' strategy. Nevertheless, because these are, in general, rare events with uncertain periodicity they are difficult to target by measurements to be carried out at pre-selected, fixed times. One may, if the allocated measurement window is too narrow, miss the phenomena altogether. Alternatively, if it is too wide, valuable resources will be wasted. Performing decision-making onboard the rover allows for science goals to be realized under circumstances where a control loop from the ground is not possible because there is

not enough time for scientists to react to changes in data patterns (by organizing dedicated campaigns, for instance). However, the present framework is not designed to replace entirely human decision making but rather to optimize the combined action of human experts and the REMS embedded system. REMS planning and scheduling activities are performed on the ground, by means of the Ground Data System (GDS). The GDS provides all the computational tools required to accomplish these tasks as well as instrument health monitoring and data analysis. An Opportunistic Science Software (OSS) package is integrated in the GDS. Every day, the downloaded REMS data product is processed at the GDS and the part specific to the self-triggering procedure is fed into the OSS. The complete statistics of event detection is recorded for housekeeping, diagnosis and debugging purposes. On a regular basis, analysis of the results will permit fine tuning and in-flight calibration of the EFA's. This procedure will be supervised jointly by scientists and software engineers present at the GDS. The standard operation procedures involve generating activity plans for a few sols determining, whenever possible, if local meteorological events of interest are likely to occur in that period. Having negotiated with the MSL project management the resources for the period being considered, the extra measurement time available is distributed through a number of possible EM windows. These and other relevant parameters are updated at the next Schedule Table Script (ST_Script) uplink. The REMS computer system is then left to decide autonomously whether or not to switch into EM according to the disposition of allocated windows and the output of the EFA's.

4.2 Design Flexibility

The above framework requires an onboard software design capable of offering full flexibility as the process of calibration of the EFA's will continue on Earth during the MSL cruise phase. Design flexibility is also important so that diagnosis, fine-tuning and debugging of the EFA's are possible after operations begin on martian surface, with minimal resort to software patching. It is also expected to reduce substantially the risks to baseline operation associated with the relative increase in software complexity which the Opportunistic Science capability inevitably carries. The current design rationale is as follows: at the end of a measuring session the REMS micro-controller will run a prescribed sequence of EFA's. The order in which the EFA's on the list are executed can be modified at every ST_Script update. The first step in

any EFA is to process sensor raw data by applying *estimators* to compute *observables*. The latter emulate as close as possible the behavior of the actual physical parameters (e.g. temperature, pressure, wind magnitudes and direction, etc.). Estimators are thus simple procedures that involve functions and coefficients determined during the sensor's calibration. The reliability of observables as proxy values is assessed through comparison with the output of the GDS post-processing software that retrieves the 'true' physical parameters from raw data. Each EFA is divided in two parts running sequentially: detection and validation. Only validated detections will cause REMS to self-trigger into EM. The detection part of the EFA reads the relevant observables and, through a sequence of logical operations decides whether or not a detection has taken place. Then it assigns an initial confidence level to that detection, called the Baseline Confidence Level (BCL) to be compared with the uploaded Threshold Confidence Level (TCL); if $BCL > TCL$, the detection is validated: the EFA will immediately stop running, all subsequent EFA's on the list will be overruled and the unit will shift into EM at once. If at least one detection has occurred but nevertheless had been assigned a BCL value below threshold, then the EFA will compute a number of flags and/or consult a table of previously set flags. The final result of these computations is an integer value, called the Increase in Confidence Level (ICL), incremental to the BCL. Both intervene to calculate the final Detection Confidence Level (DCL). If $DCL > TCL$, then EM is triggered. Flags are encoded signals either internally generated by the EFA acting on observables or historic records, or engendered at the GDS. In the later case, the flags are generated by the OSS package and uploaded as parameters in the ST_Script. Flags have limited resolution (small integers and binary numbers) and intervene only at the validation stage. For instance, the orography flag may take up to 16 integer values representing a type of orographic feature – crater rim, mountain ridge, rock, etc. – a scale of elevation with respect to rover position and the feature's size. Internal generation of flags, within the EFA, will sometimes involve elementary statistical analysis while in other cases it implicates logical operations and/or Lookup Table (LUT) consultation. For instance, in the EFA for microfronts described in the next section, the PLO flag – for 'Pressure Low' – indicates, based on the history of measurements performed in past sessions, whether a sustained pressure decrease in the hours preceding the passage of a cold front has occurred. When no detection has taken place, or achieved the TCL required for validation, the system proceeds to execute the next EFA on the list. If, once all the EFA's on the list

have been executed, no detection occurred or was validated, the unit will continue its scheduled operations normally, going into sleeping mode at the pre-scheduled time. REMS will keep in memory the session averages of the observables as well as a history of the parameters involved in the EFA's, such as confidence levels, flags, etc. Normally, data for a complete sol should be kept but this may vary depending on total memory load.

4.3 Two-tier Example Algorithm

We provide an example of EFA structure in Figure 2 using a simplified flow-chart. The example EFA considered here targets microfronts. It is a Multiple-Session Detection Algorithm (MSDA), *i.e.* entails accessing data from more than one session. Note also that it requires detrending, *i.e.* the deviations with respect to the foreseeable trend have to be calculated in the beginning of the algorithm. A first level detection, L1, results from a qualitative divergence with respect to the trend, when temperature rises in late afternoon or drops before its normal apex, for instance. The persistence of this inversion for a number of consecutive sessions is interpreted as a sign of confidence build up, and thus the Baseline Confidence Level (BCL) is set accordingly. A second detection level, L2, is based on quantitative divergence only. Even in the case when the sign of temperature variation is the one expected, there still may be a discrepancy in its quantitative rate-of-change (ROC). For instance, the ROC value may exceed some threshold d above the expected temperature progression. As this condition is obviously weaker than trend inversion it shall be assigned a lower BCL. If the relevant BCL is lower than the threshold, then the EFA will proceed to compute flags – in this case the PLO flag mentioned above and also two other flags signaling, respectively, an increase in wind magnitude (WIM) and the persistence in direction (DIR) of the wind vector. Depending on their values, the DCL's may or may not be increased, as previously described.

5 FUNCTIONALITY TESTS AND CALIBRATION

5.1 Overview of the Procedure

Functionality tests and calibration of the EFA's will be carried out on a desktop computer emulation using selected signatures of the events as input. These test signatures can be created from: i) real

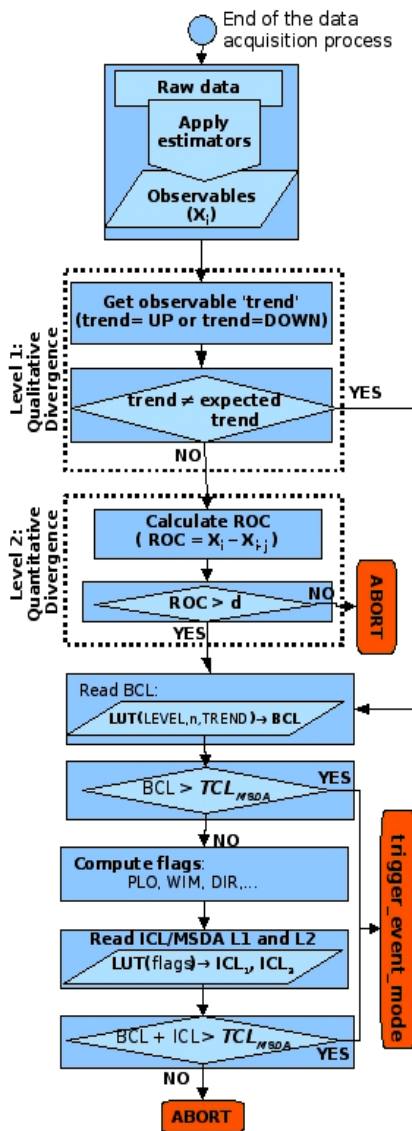


Figure 2: An example EFA flow-chart.

Martian weather data ii) output of mesoscale models ii) meteorological data from martian environmental analogues iii) engineered representative datasets and iv) field tests. Upon application of the fine-tuning procedure the percentage of false positives and false negatives is evaluated and once their numbers lie within acceptable levels the EFA is considered calibrated.

5.2 Martian Weather Data

Up until now only the Viking and Pathfinder missions have carried out successful meteorological experiments on the surface of Mars. The Phoenix lander - currently in its cruise phase and due to land

on May 25, 2008 – carries temperature and pressure sensors together with wind speed and direction indicators. Currently we are working with data from Viking and Pathfinder and, if Phoenix is successful, we shall use its data as well. Each of those datasets has its merits and faults: the Viking data have poor resolution but is extensive; Pathfinder data cover a much shorter period but display higher resolution. In the later case, in addition to signatures of dust devils registered by the sensors, there are, in some cases, simultaneous confirmation from images. We shall apply the EFA's to selected parts of the time series chosen because they reveal the characteristics of a given event, with particular focus on the patterns already presented in the scientific literature as symptomatic of the phenomena under scrutiny. After a number of refinement stages we expect to reach the minimal algorithmic complexity required to detect the relevant features.

5.3 Martian Environmental Analogues and Field Tests

Because weather data from Mars is scarce we will have to use data from martian environmental analogues and field tests on Earth. Overall, the best candidates for the role of environmental analogues are the dry valleys of Antarctica, because of the extremely cold and dry air and the fact that a fairly predictable regime of robust slope winds exists. When searching for dust devils and plumes the best places to look for on Earth are deserts. These phenomena have been extensively studied in the Arizona desert (Renno, 2004). It is an unfortunate yet unavoidable fact that the optimal locations to conduct field tests are harsh environments in remote areas of difficult (and thus expensive) access. This fact poses a constrain in our current development of a priority list for site selection for field tests. The sites being evaluated include Deception Island – home to a Spanish Antarctic base, where members of the REMS team regularly conduct research – as well as sites in the Atlas mountains and the Atacama desert. Nevertheless, locations in close reach of the CAB facilities will be favored, at least in the preparatory phases of the field test campaign. In this vein, we are currently assessing the reliability of certain arid and semi-arid regions in the Iberian peninsula as partial analogues. A first set of tests will be conducted with a commercial weather station. In a later stage, we shall use a REMS model, complete with rover mast and deck dummies, to carry out end-to-end performance and reliability tests of the EFA's under the geometrical constraints imposed by integration with the rover.

5.4 Engineered Artificial Datasets

In order to overcome the difficulties of using terrestrial data in calibration, due to dissimilarities between the atmospheres of the two planets, we shall work also with engineered data. Based on the analysis of field data and sensor calibration data obtained under martian conditions, we shall create artificial datasets to mimic the signatures expected from the phenomena. Integrating sensor calibration data with theoretical models and/or computer simulations, notably mesoscale models of the atmosphere, we can also create prototype signatures that minimize the constraints set by wind tunnels and martian atmosphere simulation chambers.

6 ANTICIPATION

We have outlined a strategy to enable autonomous decision-making aboard a mobile robotic surface probe, in order to optimize its science return. Provided with this capability, REMS will be able to detect sudden and unexpected changes in environmental parameters adjusting its operation mode in response, without the need for a control loop from Earth. As planetary exploration stretches out further away from the Earth, communication delays and interruptions will make remote control increasingly harder. Though in the case of flyby and orbital missions these delays can be accommodated without much hindrance the situation is very different for robotic probes evolving in a changing environment. For a vehicle exploring Titan's surface or Europa's putative ocean, onboard intelligent systems will be essential for virtually all craft operations including guidance and steering, obstacle avoidance, selection of scientific targets, data downlink management, etc. Sending probes to land on these far away bodies is not envisaged in the near future. Until then, the surface of Mars will be the cutting-edge operation scenario and the most demanding testing ground for the implementation of autonomous decision-making in the context of planetary exploration by robots. Inspired by the OASIS concept (<http://ml.jpl.nasa.gov/oasis/>), we decided to use REMS for an experimentation in Opportunistic Science. However, because the processing power of the REMS computer system is low, (even when compared to common household appliances), we are forced to use exceedingly simple programs whose ability to identify changes in environmental conditions cannot be guaranteed *a priori*. But even if some of the concepts being

developed do not mature fast enough to be implemented in REMS/MSL, they still may find application in future missions, to Mars and beyond. Because operational autonomy is intrinsic to the REMS design, the software under development needs not to be integrated with the spacecraft's flight software, and thus the employment of an upgraded version in future missions should be straightforward. This tempts us to anticipate a bold strategy. Indeed, one can envisage a scenario in which future landers and rovers being sent to Mars would each carry a station similar to REMS, provided with autonomous decision-making abilities. Thus, even after every other component of the mission has reached its end-of-life the spacecraft would not turn into waste but instead become a node in an ad-hoc network of weather stations. Of course, each future autonomous station would need dedicated power and data downlink systems; and would have to operate at a low sampling rate and use recurrent transfer windows in order to lower its power consumption and step-up the probability that the data streams be received unbroken by a (future) small network of low-orbit communication relay satellites. This network could be considered either as a precursor mission or a scalable inexpensive alternative to the MetNet mission (<http://www.ava.fmi.fi/metnet-portal/>), currently being proposed, and also NetLander (<http://smc.cnes.fr/NETLANDER/>), abandoned in 2003.

REFERENCES

- R. Castaño et al., *Journal of Field Robotics*, 24 (5), 379-397, 2007.
- N. Renno et al, *Journal of Geophysical Research*, 109, E07001, 2004.

(A) VISION FOR 2050

The Road Towards Image Understanding for a Human-Robot Soccer Match

Udo Frese

*Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), 28359 Bremen, Germany
Udo.Frese@dfki.de*

Tim Laue

*SFB/TR 8 Spatial Cognition, Faculty 3 - Mathematics and Computer Science, Universität Bremen, 28359 Bremen, Germany
timlaue@informatik.uni-bremen.de*

Keywords: Vision, Recognition, Image Processing, State Estimation, Sensor Fusion, Robot Soccer, Sport Robotics.

Abstract: We believe it is possible to create the visual subsystem needed for the RoboCup 2050 challenge – a soccer match between humans and robots – within the next decade. In this position paper, we argue, that the basic techniques are available, but the main challenge will be to achieve the necessary robustness. We propose to address this challenge through the use of probabilistically modeled context, so for instance a visually indistinct circle is accepted as the ball, if it fits well with the ball’s motion model and vice versa. Our vision is accompanied by a sequence of (partially already conducted) experiments for its verification. In these experiments, a human soccer player carries a helmet with a camera and an inertial sensor and the vision system has to extract all information from that data, a humanoid robot would need to take the human’s place.

1 INTRODUCTION

Soon after establishing the RoboCup competition in 1997, the RoboCup Federation proclaimed an ambitious long term goal.

“By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, comply with the official rule of the FIFA, against the winner of the most recent World Cup.”

Kitano and Asada (1998)

Currently, RoboCup competitions take place every year. Within a defined set of different sub-competitions and leagues, incremental steps towards this big goal are made (RoboCup Federation, 2008). Although, a rapid and remarkable progress has been observed during the first decade of these robot competitions, it is not obvious, if and how the final goal will be reached. There exist rough roadmaps, e.g. by Burkhard et al. (2002), but in many research areas, huge gaps must be bridged within the next 40 years.

While this is obvious for several areas, e.g. actuator design and control, we claim that the situation is surprisingly positive for vision:

Within the next decade, it will be possible to develop a vision system that is able to provide all environmental information necessary to play soccer on a human level.

Annual RoboCup competitions are always bound to strict rule sets (defined for the state of the art of the competing robots) and demand competitive robot teams. Thus only incremental progress adapting to actual rule changes (which continuously rise the level of complexity) is fostered. By developing the aforementioned vision system independently of these competitions, we hope to set a new landmark which could guide the incremental development.

Because a *real* human level soccer robot will not be available for a long time, our vision is accompanied by a (partially already conducted) set of experiments that verify our claim without needing a robot.

This paper is organized as follows: Section 2 roughly identifies the challenges for playing robot soccer and compares them to the state of the art in robotics. In Sect. 3 we explain, why the basic techniques for the vision system are available. We argue, why the remaining challenge is robustness, for which we present our idea of a solution in Sect. 4. Finally, a sequence of experiments to verify our claim is described in Sect. 5.

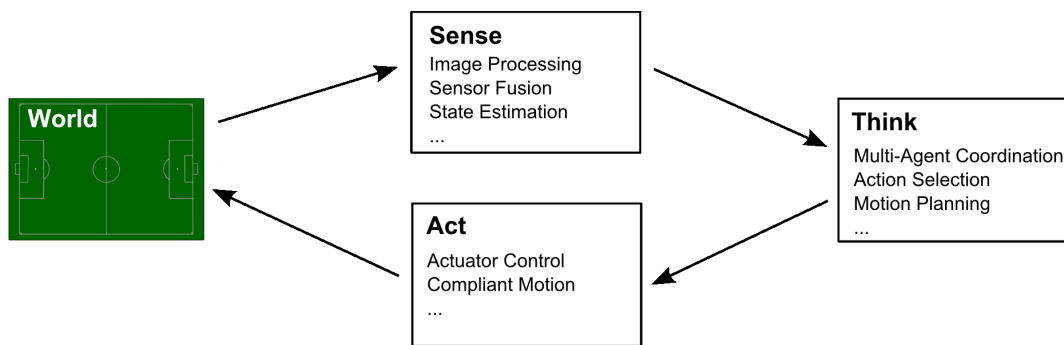


Figure 1: The Sense-Think-Act cycle roughly depicting major tasks for playing soccer with a humanoid robot.

2 CHALLENGES FOR PLAYING SOCCER

The global task of playing soccer consists of several different, interdependent challenges. We roughly categorize them according to the Sense-Think-Act cycle (see Fig.1). This should be considered as a possible architecture for illustration. In the following, the challenges are described in reverted order but with decreasing degree of difficulty.

2.1 Challenges for Actuation

The hugest obvious gap may be observed in the field of actuation. Nowadays, the probably most advanced humanoid robot, Honda's ASIMO, is capable of running at a top speed of six kilometers per hour (Honda Worldwide Site, 2007). This is an impressive result, but still more than five times slower than the top speed of a human soccer player. A similar gap regarding kicking velocity has been pointed out by Haddadin et al. (2007). They showed that a state-of-the-art robot arm (with a configuration comparable to a human leg) is six times slower than required to accelerate a standard soccer ball to an adequate velocity. It is still an open issue, whether today's motor technology could be developed further on enough, or if more efficient actuators, e.g. artificial muscles, will be needed. Since soccer is a contact sport leading to physical human-robot interaction (Haddadin et al., 2007), not only direct position control but also approaches for compliant motion, such as impedance control, need to be taken into account.

Additionally, the problems of energy efficiency and power supply need to be solved. The ASIMO robot for example is, according to Honda Worldwide Site (2007), capable of walking (with a speed of less than three kilometers per hour) for 40 minutes.

2.2 Challenges for Thinking

In this area, two different aspects may be distinguished: motion planning and high-level multi-agent coordination. The latter is a research topic in the RoboCup Soccer Simulation League since a while and has reached a remarkable level. Dealing with the offside rule as well as playing one-two passes are standard behaviors, complex group tasks as playing keepaway soccer serve as a testbed for learning algorithms (Stone et al., 2005). This area could be considered to be already quite close to human capabilities.

On the other hand, when playing with real humanoid robots, sophisticated methods for motion planning are needed. The current research frontier on humanoid motion control is balancing and dynamic foot placement for walking robots. Algorithms for full-body motion planning exist (Kuffner et al., 2002), but are subject to restrictions that make them inapplicable to tasks as playing soccer.

Here is a big gap to human level soccer. As an example consider volley-kicking. The player has to hit the ball exactly at the right time, position, and velocity, with a motion compatible to the step pattern, allowing balancing and considering opponents. Last but not least, all this must happen in real-time.

2.3 Challenges for Sensing

According to Kitano and Asada (1998), it is evident that the robots' sensorial capabilities should resemble the human ones. Thus, we could assume to deal with data from cameras and inertial sensors emulating the human eyes and vestibular system. The required information are estimates of the own position and the positions of the ball and of other players. In case of tackles or dribbling, the latter will be needed to be recognized in more detail (e.g. the positions of the feet and limbs).

Current solutions for these tasks and our idea how to bridge the remaining gap are presented in the following section.

3 THE VISION SYSTEM

Our main thesis is that the “sense” part of the RoboCup 2050 challenge can be realized within a decade starting from the current state of the art in computer vision. This is remarkable, since the “act” and “think” parts are apparently lightyears away from reaching human level performance and for computer vision in general, this is also true. The reason, why we believe such a vision system can be realized, is, that unlike a household robot for instance, a soccer robot faces a rather structured environment.

3.1 State of the Art

The objects relevant in a soccer match are the ball, the goals, the line markings and of course the players. Ball, goal and line markings are geometrical features, i.e. circles and lines. There is a large number of algorithms for detecting them in images, from the classical Hough transform (Davies, 2004) up to a range of more elaborate methods (Guru and Shekar, 2004).

Recognizing other players is more challenging. It is particularly difficult because we will probably need not only the general position but the detailed state of motion for close range tackling and to infer the player’s action for tactical purposes. Fortunately, people tracking is an important topic in computer vision with a large body of literature (Price, 2008; Ramanan and Forsyth, 2003).

Furthermore, soccer scenes are lightly colored with green lawn and the players wearing colored clothes of high contrast. In the RoboCup competition, this idea is taken to an extreme, where most teams rely on color segmentation on a pixel-per-pixel basis as their primary vision engine. This will not be possible for real-world soccer, mainly due to changing lighting conditions. Still color can provide a valuable additional cue, at least when looking below the horizon, where objects are in front of green lawn.

The background above the horizon, including the stadium and the audience is of course also visible and unfortunately rather undefined. However, if it is relevant for the soccer robot at all, then not for recognition, but only in the sense of a general landmark. For this purpose there are nowadays well working techniques, such as the Scale Invariant Feature Transform (SIFT) (Lowe, 2004).

Overall, understanding a soccer scene from the player’s perspective seems much easier than for instance understanding an arbitrary household, traffic or outdoor scene. Indeed there are already half-automatic systems in the related area of TV soccer scene analysis, for example the ASPOGAMO system by Beetz et al. (2006, 2007) proving that soccer scene understanding in general is on the edge of being functional.

3.2 Open Problems

So, is a vision system for the RoboCup 2050 challenge an easy task? We believe it is not. It is surprisingly a realistic task but well beyond the current state of the art. The first problem is, that the camera is moving along with the head of the humanoid soccer robot. To predict a flying ball, the orientation of the camera must be known very precisely. It seems unrealistic that the necessary precision can be obtained from the robot’s forward kinematic, since unlike an industrial robot, a humanoid robot is not fixed to the ground. So our solution is to integrate an inertial sensor with the camera and fuse the complementary measurements of both sensors in a probabilistic least-square framework.

The second problem is the player’s perspective. It is much more difficult than the overview perspective used in TV soccer scene analysis. In the TV perspective the scale of an object in the image varies by a factor of about 3 (Beetz et al., 2006, Fig. 5) whereas in the player’s perspective it can vary by a factor of 250 assuming the distance to an object ranging from 0.5m to 125m. Hence, for instance the people detection algorithm must handle both extreme cases, a person only the size of a few pixels, where an arm or a leg maybe thinner than a single pixel and a person much larger than the camera’s field of view, only partially visible. Furthermore, in an image from the player’s perspective, other players will extend beyond the green lawn of the field into the general background. Hence it is not possible to search for non-green blobs as an easy first processing step. This can also happen for a flying ball, which is then particularly difficult to detect.

However, the third and most severe problem is, that from our experience, most of the academic computer vision systems perform on the level of lab demonstrators requiring nicely setup scenes and lighting conditions and usually considerable parameter tweaking. So, to summarize, for the vision part of the RoboCup 2050 challenge, we do not need a new level of functionality as for many other grand challenges, but we need a new level of robustness.

4 ROBUSTNESS THROUGH CONTEXT

We propose to address the question of robustness by utilizing probabilistically modeled context information, formulating the overall scene understanding and prediction problem as a global likelihood optimization task. This idea in general is not entirely new (Ullman, 1995; Binnig, 2004; Leibe et al., 2007), but we believe it is particularly well suited to this task and also the task is well suited to study this methodology.

4.1 Data-Driven Bottom-Up Processing

Most current vision systems use a data-driven bottom-up approach (Frese et al., 2001; Röfer et al., 2005; Beetz et al., 2007, as examples). Usually, low level features are extracted from the image and then aggregated through several stages to high level information. Each stage may incorporate some background knowledge at its particular level but does not take information from higher levels into account. It simply takes some input from the previous lower level and passes the result of the computation to the next higher level.

As an example, a classical Hough transform starts by classifying pixels as edge or not by thresholding the result for instance of a Sobel filter. Similar the system by Beetz et al. starts by classifying pixels as lawn or not on the basis of their color. This is a hard decision taken on the lowest level without any higher level knowledge, such as the fact that we are looking for a ball or the ball's motion model. Such a pixel-wise classification can be very ambiguous. Often we could, for instance, classify a borderline pixel correctly as belonging to the ball, although it looks rather greenish, if we considered the context of the ball or its motion model. However, in conventional vision systems, on the low level this knowledge does not exist and on the higher level, the fact, that this pixel was borderline in the classification, is lost due to committing to a hard decision on the lower level.

4.2 Global Likelihood Optimization

We believe, that much of the brittleness of current vision systems originates from this phenomenon. So our approach for increased robustness is an overall likelihood optimization. In the example above, the variables to be optimized would be the 2D image circle (center, radius) and the 3D position and velocity of the ball over time. The likelihood would be the product of the following likelihoods for all images:

1. a motion model likelihood binding the 3D positions and velocities over time;



Figure 2: Our proposed experiment: Mount a camera and an inertial sensor on the head of a human soccer player and use them to extract all the information, a humanoid soccer robot would need to take the human's place.

2. a camera model likelihood binding 2D circles to 3D positions;
3. a circle edge likelihood, indicating how much contrast there is in the image along the outline of the hypothesized 2D circle;
4. a circle color likelihood, indicating how well the color inside the 2D circle corresponds to the ball.

The first two factors are Gaussians expressing the models as formulas with uncertainty (Birbach, 2008). The last two look directly into the image for a specific circle, returning a gradual result. In this approach, an indistinct ball would get a lower likelihood in 3. and 4. but this could be compensated by 1. and 2. if it fits well to the context of a flying ball.

The problem is understanding an image sequence, i.e. estimating over time. Indeed, successive images are linked by a motion model and this provides most of the context we want to build upon. However, we propose not to use incremental filters, such as EKF or a particle filter, but to look back into the raw images of the last few seconds at least. This approach has surprising advantages. Imagine the ball is kicked, but during the first 100ms there is too little contrast to the background so it is not detected. Now when it is detected, there is new information on where the ball has been before from the ball's motion model. The old images are still in memory and tracking the ball back in time is much less ambiguous than finding the ball without context and will probably succeed. Paradoxically, once the system has detected the ball it has

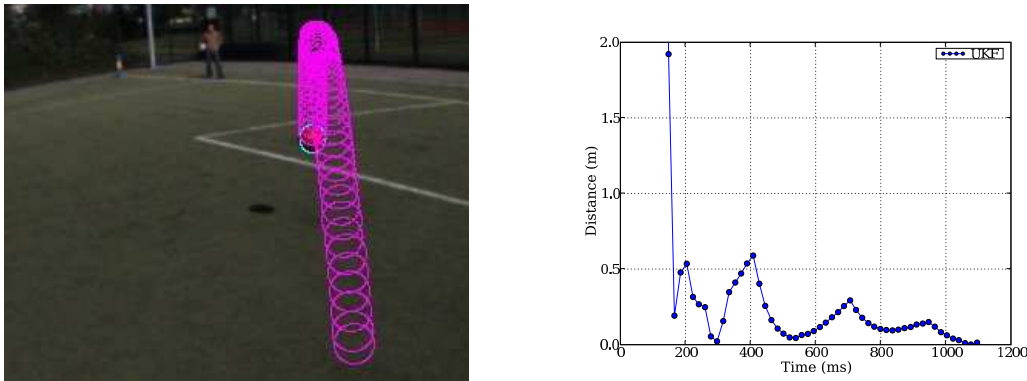


Figure 3: Predicting the trajectory of a flying ball from a moving camera-inertial system. As an initial study, the ball, the lines and the goal corners have been manually extracted from the images. From this data, the trajectory of the ball is predicted (left). The right plot shows the error of the predicted touch down point varying over time. It shows, even though the camera is moving, the prediction is roughly precise enough for interception. See <http://www.sport-robotics.com/icinco/>.

already observed it for 100ms. The first prediction is not delayed at all, because prior to that the ball must have been observed for some time anyway.

Overall, we believe that the approach of a global likelihood optimization directly in the images is an elegant way to greatly increase robustness.

5 PROPOSED EXPERIMENTS

For a vision to become reality, realistic intermediate steps are necessary. It would not help, if we build a vision system now but then had to wait until a human level soccer robot is available. So we propose a sequence of experiments, that, without a humanoid robot, ultimately allows to verify that the proposed system is appropriate for human level soccer (Fig.2).

5.1 Helmet Camera with Inertial Sensor

The basic idea is to let a human soccer player wear a helmet with a camera and an inertial sensor and verify, that the information extracted by the vision system from the sensor data, would allow a humanoid robot to take the human's place.

As a first experiment we propose to record data from a soccer match and run the vision system on that data offline. Since it is hard to obtain ground-truth data, we would use our expert's judgment to assess, whether the result would be enough for a humanoid robot to play soccer. It is very advantageous to work on recorded data allowing to reproduce results for debugging and analysis and to run the system even if it still not real-time. Overall, it allows to first concentrate on functionality and robustness instead of computational efficiency and technical integration.

We have already conducted a very first experiment (Kurlbaum, 2007; Birbach, 2008), where the ball and the field lines are manually extracted from the recorded images (available on request). The ball's trajectory is predicted by least-square estimation using the likelihood functions 1. and 2., as well as corresponding equations for how the inertial sensor observes the free motion of the camera (Fig.3). The results indicate, that if the ball can be detected in the image with about one pixel precision, the prediction would be precise enough. We believe that this kind of studies which deliberately discard essential aspects, such as integration, real-time computation, or autonomy are undervalued by the community who favors full system approaches. But even from a full system perspective, it is much more valuable to obtain an extensive result on a subsystem which then can guide the full system design than to do another increment on a full system.

5.2 Motion Capture Suit

Departing from the experiment above, one might ask whether more sensors are needed than just camera and inertial. Both human and humanoid robot can derive their own motion from the joint angles. This provides the horizontal motion (odometry) and the height over ground. The horizontal motion facilitates localization and the height derived from vision is much less precise. Indeed, we experienced that the uncertain height is a major part of the error in Fig. 3.

An intriguing idea is to equip the human player with a *tracker-less* motion capture suit (Xsens Technologies B.V., 2007) measuring joint angles. Apart from providing the kinematic information discussed above, it also provides the trajectory of both feet. If

the player hits the ball, one can compare the predicted ball trajectory with the real foot trajectory and evaluate the precision. This is important since ground truth is not available.

5.3 Virtual Reality Display

The experiments above have the drawback that they are evaluated by an expert looking at the vision system's output. The most direct proof that this is all you need for playing soccer would be to give a human just that output via a head mounted display and see whether s/he can play.

The approach is of course fascinating and direct, but we have some concerns regarding safety. Anyway, this experiment becomes relevant only after we are convinced in principle, that the system is feasible. So this is something to worry about later.

6 CONCLUSIONS

In this position paper, we have outlined the road to a vision system for a human-robot soccer match. We claim that, since soccer is a rather structured environment, the basic techniques are available and the goal could be reached within a decade. The main challenge will be robustness, which we propose to address by optimizing a global likelihood function working on a history of raw images. We have outlined a sequence of experiments to evaluate such a vision system with data from a camera-inertial system mounted on the head of a human soccer player.

The reason, we are confident such a system can be realized within a decade is the insight that it does not need general common-sense-reasoning AI. This is good news for the RoboCup 2050 challenge. But it suggests that, even when we meet that challenge, it does not imply we have realized the dream of a thinking machine, the whole challenge had started with.

That would not be the first time.

REFERENCES

- Beetz, M., Gedikli, S., Bandouch, J., Kirchlechner, B., v. Hoyningen-Huene, N., and Perzylo, A. (2007). Visually tracking football games based on tv broadcasts. *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India*.
- Beetz, M., v. Hoyningen-Huene, N., Bandouch, J., Kirchlechner, B., Gedikli, S., and Maldonado, A. (2006). Camerabased observation of football games for analyzing multiagent activities. In *International Conference on Autonomous Agents*.
- Binnig, G. (2004). Cellenger automated high content analysis of biomedical imagery.
- Birbach, O. (2008). Accuracy analysis of camera-inertial sensor based ball trajectory prediction. Master's thesis, Universität Bremen, Mathematik und Informatik.
- Burkhard, H.-D., Duhaut, D., Fujita, M., Lima, P., Murphy, R., and Rojas, R. (2002). The Road to RoboCup 2050. *IEEE Robotics and Automation Magazine*, 9(2):31–38.
- Davies, E. R. (2004). *Machine Vision. Theory, Algorithms, Practicalities*. Morgan Kaufmann.
- Frese, U., Bäuml, B., Haidacher, S., Schreiber, G., Schaefer, I., Hähle, M., and Hirzinger, G. (2001). Off-the-shelf vision for a robotic ball catcher. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui*, pages 1623 – 1629.
- Guru, D. and Shekar, B. (2004). A simple and robust line detection algorithm based on small eigenvalue analysis. *Pattern Recognition Letters*, 25(1):1–13.
- Haddadin, S., Laue, T., Frese, U., and Hirzinger, G. (2007). Foul 2050: Thoughts on Physical Interaction in Human-Robot Soccer. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Honda Worldwide Site (2007). Honda World Wide — Asimo. <http://world.honda.com/ASIMO/>.
- Kitano, H. and Asada, M. (1998). RoboCup Humanoid Challenge: That's One Small Step for A Robot, One Giant Leap for Mankind. In *International Conference on Intelligent Robots and Systems, Victoria*, pages 419–424.
- Kuffner, J. J., Kagami, S., Nishiwaki, K., Inaba, M., and Inoue, H. (2002). Dynamically-stable Motion Planning for Humanoid Robots. *Auton. Robots*, 12(1):105–118.
- Kurlbaum, J. (2007). Verfolgung von ballflugbahnen mit einem frei beweglichen kamera-inertialsensor. Master's thesis, Universität Bremen, Mathematik und Informatik.
- Leibe, B., Cornelis, N., Cornelis, K., and Gool, L. V. (2007). Dynamic 3D Scene Analysis from a Moving Vehicle. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110.
- Price, K. (2008). The annotated computer vision bibliography. <http://www.visionbib.com/>.
- Ramanan, D. and Forsyth, D. (2003). Finding and tracking people from the bottom up. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- RoboCup Federation (2008). RoboCup Official Site. <http://www.robocup.org>.
- Röfer, T. et al. (2005). GermanTeam RoboCup 2005. <http://www.germanteam.org/GT2005.pdf>.
- Stone, P., Sutton, R. S., and Kuhlmann, G. (2005). Reinforcement Learning for RoboCup-Soccer Keepaway. *Adaptive Behavior*, 13(3):165–188.
- Ullman, S. (1995). Sequence seeking and counter streams: A computational model for bidirectional information flow in the visual cortex. *Cerebral Cortex*, 5(1):1–11.
- Xsens Technologies B.V. (2007). *Moven, Inertial Motion Capture, Product Leaflet*. XSens Technologies.

WHAT'S THE BEST ROLE FOR A ROBOT?

Cybernetic Models of Existing and Proposed Human-Robot Interaction Structures

Victoria Groom

*Department of Communication, Stanford University, 450 Serra Mall, Stanford, U.S.A.
vgroom@stanford.edu*

Keywords: Cybernetics, human-robot interaction, robot teams, social robotics.

Abstract: Robots intended for human-robot interaction are currently designed to fill simple roles, such as task completer or tool. The design emphasis remains on the robot and not the interaction, as designers have failed to recognize the influence of robots on human behavior. Cybernetic models are used to critique existing models and provide revised models of interaction that delineate the paths of social feedback generated by the robot. Proposed robot roles are modeled and evaluated. Features that need to be developed for robots to succeed in these roles are identified and the challenges of developing these features are discussed.

1 INTRODUCTION

Human-robot interaction (HRI) is the study of humans' interactions with robots. While the field of robotics focuses primarily on the technological development of robots, HRI focuses not just on the robot, but on the broader experience of a single or group of humans interacting with robots. Researchers have long sought to deploy robots alongside humans as human-like partners, minimizing humans' involvement in dangerous or dull tasks. While robots have demonstrated some promise as coordination partners, in practice they contribute little to achieving humans' goals, often requiring more attention and maintenance and eliciting more frustration than their contributions are worth. Through these failures, it has become clear that not only must robots' technical abilities be improved; so must their abilities to interact with humans.

Humans prefer that all interaction partners that exhibit social identity cues display role-specific, socially-appropriate behavior (Nass & Brave, 2005; Reeves & Nass, 1996). A robot must cater to this human need to facilitate a successful interaction, but designers of robots are rarely attuned to human psychological processes.

Discounting human needs and expectations has led HRI researchers to propose design goals for robots that fail to fully consider the needs of humans. Creating a "robot teammate" has become a

guiding goal of the HRI community, even though the needs and expectations of humans intended to team with robots have not been properly considered (Groom & Nass, 2007). Because HRI has yet to become a fully-established field, putting careful thought into the goals of HRI now is essential for its future success.

Cybernetics--the study of complex systems, particularly those that feature self-regulation--places a strong emphasis on the value of modelling interactions and provides an established framework for understanding and talking about systems, something much needed in HRI. While HRI researchers often model systems within a robot, little attention has been paid to modelling the interaction between a human and a robot.

Cybernetic models featuring a goal, comparator, actuator, and sensor clearly delineate the relationship between systems and their environments. The system's goal is to affect the environment in some manner within some parameters. The system's comparator determines if the goal has been achieved and transmits this information to the actuator, which takes some action on the environment. A sensor then detects some feature of the environment, and this information is passed to the comparator. With cybernetic models, systems continually influence and are influenced by their environments and other systems.

In this paper, I draw on cybernetics to represent the models shaping the design of robots intended for

close human interaction. I critique these models and offer revised models that include the human, the robot, and the interaction between them. I also model the conversational abilities required of teammates, identifying those features that must be developed in robots for humans to accept them. The difficulty of meeting these requirements raises questions as to whether the field of HRI is pursuing optimal goals.

2 HRI DESIGN TODAY

Today's robots are not yet capable of serving in roles like teammate that require sophisticated social capabilities. While designers are working on creating robots capable of filling these roles, the majority of existing robots fill less demanding roles. These roles have lower requirements for autonomy, intentional action, and socially-appropriate behavior, and are similar to those roles filled by other advanced technologies such as computers.

2.1 Robot Roles

One role that robots are often designed to fill is *task completer*. In this role, robots complete a task designated by a human. Many military robots, such as bomb-detecting and bomb-defusing robots, are modelled in this role. In some cases the robot's system may be non-cybernetic and in others it may be cybernetic. With non-cybernetic task-completer robots, the human sets the goal of the robot and the robot affects the environment in a manner intended to achieve the goal. In the case of a bomb-detecting robot, the robot may run tests on a potential bomb and send data back to distantly-located humans. The process terminates at this point, as the system lacks a sensor, comparator, or both. The process used by the human to select the goal is not modelled, nor is there any indication that the robot's behavior affects the humans' goals.

A cybernetic task completer is generally more robust and capable of more complex tasks than a non-cybernetic task completer. The Roomba is a popular example of a cybernetic task-completer robot. As with non-cybernetic task completers, the goal of a cybernetic task completer is set by a human. Unlike non-cybernetic task completers, the Roomba features sensors and a comparator that partly comprise a cybernetic system, which enables the Roomba to navigate obstacles. As with the non-cybernetic task-completer, the human is considered only peripherally in the design process. In the case

of the Roomba, the human is modelled as having little interaction with the robot. The human provides the robot power, maintains and cleans it, and initiates its activities by turning it on.

The *tool* is another model commonly used for the design of robots. A tool extends humans' influence on the environment or grants humans power over the environment that they do not normally possess. Because a robot tool is much like an extension of the self, attention is paid to the human operating the robot: the goals and processes of the humans are often considered in the design of the robot. The robot is designed to help a human complete a task or range of tasks. As a tool, the robot is outside the human system, acting within the environment on the environment.

Search and rescue robots often take the form of a tool. One reason robot tools are useful in search and rescue situations is because they enable people to examine and influence areas that are inaccessible or too dangerous for humans to access (Casper & Murphy, 2003). The model of the robot tool differs from models of the robot task-completer in that the influence of the robot on the human is acknowledged. However, the influence of the robot is indirect, as the human senses only the environment which contains the robot. Additionally, the influence of the robot on the human is limited to the humans' selection of the best means to implement a task strategy. The design of the robot as tool does not model the robot as influencing the human directly nor directly affecting the human's higher level goals, such as selecting a task strategy.

2.2 Social Feedback

The existing models of robots as task completers and tools fail to delineate the powerful direct influence of the robot on the human. Most designers of robots, even those within the HRI community, fail to fully recognize the social feedback that robots generate. The behaviors of humans that interact with bomb-detecting and defusing robots, Roomba, or search and rescue robots indicate that they are receiving information from the robot beyond that which is intentionally designed.

An ethnographic study of the use of the Roomba in family homes found that half of all families studied developed social relationships with it (Forlizzi & DiSalvo, 2006). These families named the robot, spoke to it, described social relationships between it and pets, and even arranged "play dates" for multiple Roombas to clean together. In addition, the Roomba affected the cleaning strategies of

household members, with males assuming a greater involvement in house-cleaning. Anecdotal evidence suggests that soldiers who interact closely with robots in high stakes situations, like bomb diffusion and search and rescue, form close emotional bonds with robots, giving them names and grieving when the robots sustains serious injuries.

In these cases, humans are responding to social information generated by the robot. Computers as Social Actors theory (CASA) was developed by Nass (Reeves & Nass, 1996). CASA posits that that even when technologies lack explicit social cues, people respond to them as social entities. Research performed under this paradigm has shown that even computer experts are polite to computers (Nass, Moon, & Carney, 1999), apply gender stereotypes to computers (Lee, Nass, & Brave, 2000), and are motivated by feelings of moral obligation toward computers (Fogg & Nass, 1997). Even unintentional cues of social identity elicit powerful attitudinal and behavioral responses from humans.

Research indicates that some of the reasons that people respond to computers socially is because computers exhibit key human characteristics (Nass, Steuer, Henriksen, & Dryer, 1994), including using natural language (Turkle, 1984) and interacting in real time (Rafaeli, 1990). Robots generally demonstrate even more human characteristics than

robots. Some robots, such as Asimo or Robosapien, feature a humanoid form. Many robots, such as Nursebot or Roomba, feature some form of locomotion, an indicator of agency. In addition, robots often exhibit at least some autonomous action and appear to humans to sense their environments, make judgments, and act on their environments. The very nature of robots make them appear even more like social entities than most other existing technologies and elicit an even more powerful social response. But only when one of the primary design goals is to foster a social relationship, as with entertainment robots like Aibo or Robosapien, is the social influence on the robot considered.

As indicated in Figure 1, the robot's behavior has a powerful influence on operators' higher-level goals. People have a high-level goal of recognizing and evaluating social information transmitted by others. Humans wish to respond to the behavior of others in a socially-appropriate manner (Reeves & Nass, 1996). While designers may have intended for the robot to be an invisible tool, it is in fact sending powerful cues indicating that it is a social entity. The robots behavior may affect humans' task strategies, either through direct feedback or by influencing humans' higher level goals to act socially appropriately.

Recognizing the influence of social feedback on

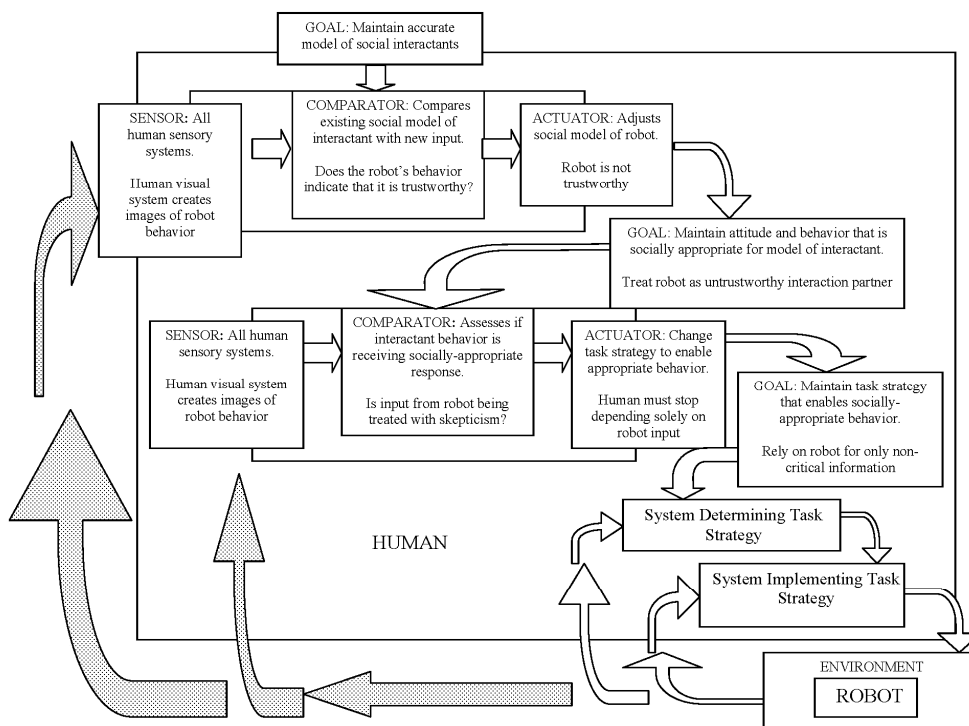


Figure 1: Model of a robot tool. Social feedback indicated with dotted arrows. Component boxes contain examples.

humans interacting with robots has important implications for the design of robots. Designers are more likely to consider which aspects of their design are likely to generate a social response from humans. Designers may be more inclined to create intentional cues to foster a social relationship or to elicit the desired social response. For example, it has been demonstrated that humans apply gender stereotypes to voices—even those that are obviously synthetic (Nass & Brave, 2005). Awareness of this effect may lead designers to choose robot voices not only based on the clarity of the robot's voice, but also based on the desired social response.

Considering social feedback when designing robots plays a key role in setting humans' expectations of robots. The fewer and weaker the cues of social identity, the lesser the likelihood is that a robot will elicit a social response. Robotocist Masahito Mori (1970) coined the term the "Uncanny Valley" to describe humans' responses of discomfort when a robot's visual or behavioral realism becomes so great that humans' expectations of human-like behavior are set too high for the robot to meet. When a robot is less realistic, humans have lower expectations and are able to tolerate non-humanlike behavior. As visual and behavioral indicators of humanness increase and human-like behaviour doesn't, people become negative. Only when the humanness of robots' behaviors catch up to their highly human-like appearance will robots emerge from the valley of uncanniness. When designing robots for interaction with humans, recognizing the role of social information in setting user expectations will enable designers to manage social cues and set expectations that the robot is capable of satisfying.

3 FUTURE OF HRI DESIGN

The roles that robots are successfully filling today, such as task completer and tool, fail to take advantage of robots' full potential. Computers also succeed in these roles, but robots have features that computers do not. Robots have the potential to move about their environments, sensing the world around them, and either transmitting that information to distantly-located humans or making decisions and acting on the environment directly.

The ultimate goal for designers involved with HRI is to create a robot capable of serving as a member of a human team. Few researchers have sought to define "team" or "team member" or identify the requirements for creating a robot team

member. The robot team member has been generally accepted as a lofty but worthy and attainable goal. (For a summary and criticism of the "robot as teammate" model, see Groom & Nass, 2007).

A well-established body of research is dedicated to the study of teams. Successful teammates must share a common goal (Cohen & Levesque, 1991), share mental models (Bettenhausen, 1991), subjugate individual needs for group needs (Klein, Woods, Bradshaw, Hoffman, & Feltovich, 2004), view interdependence as positive (Gully, Incalcaterra, Joshi, & Beaubien, 2002), know and fulfill their roles (Hackman, 1987), and trust each other (Jones & George, 1998). If a human or robot does not meet these requirements, they may never be accepted into a team or may be rejected from the team when problems arise (Jones & George, 1998).

One key requirement of teammates that underlies all other requirements is the ability to engage in conversation with other teammates. To be a successful conversation partner, a robot teammate must be able to both convey meaning in a way that other teammates can understand and understand the meaning intended in the communications of other teammates. If a robot cannot do this, human teammates can never be certain if the team shares a common goal, which makes the human unable to trust the robot in risky situations. Likewise, humans would be uncertain if the robot was subjugating its needs, viewing interdependence as positive and knowing and fulfilling its role. Without conversation, humans would feel certain that the robot was incapable of sharing a mental model.

Figure 2 provides a model of conversation between teammates that is derived from cybernetician Gordon Pask's (1975) Conversation Theory (CT). One key element of this model of conversation is the emphasis on *both* conversation partners' involvement in the communication. Another related element is that both partners construct the meaning of a message in their mind. Meaning is not directly transmitted from one conversation partner to the other, so each partner must be capable of deriving meaning from a message. A successful conversation requires that each person not only ascribe their own meaning to messages, but also infer the meaning of others and compare the meaning of each partner to determine if they are in agreement. While some robots are capable of recognizing words or gestures and responding appropriately, no robot has come close to being able to fully engage in conversation.

Figure 2 highlights those features that must be developed in robots for them to achieve the most

basic requirement of teammates: the ability to engage in conversation. These requirements may be broken down into three general categories: concepts, knowledge, and systems. To communicate and behave in a manner that allows humans to interpret meaning, robots must demonstrate awareness of basic concepts, including goals and motivation. Robots lack humans' complex hierarchy of goals. Human teammates deployed in a high-stakes situation like search and rescue maintain many goals at once, including a goal to survive, a goal to protect other teammates, and a goal to succeed at the task at hand. Robots maintain a limited number of simple goals that are always set at some point by a human.

One of the most important areas of knowledge that robots lack is an understanding of common human motivating factors and the relationship between specific goals, motivations, and actions. In order for robots to be useful in uncontrolled, changing situations, they must possess a broad body of knowledge. Robots' lack of knowledge of common goals, motivations, and actions also make them difficult for humans to understand, eliciting unintended negative responses from humans. While a human's motivation to avoid harm encourages

void, and destroy itself. Human teammates are likely to feel frustration, disappointment, and betrayal when a robot acts in a manner that is self-destructive and detrimental to the team.

In order for robots to be accepted by humans in situations that rely on conversations and mutual dependence, robots must exhibit behavior that appears to humans to imply an underlying systems much like the system used by humans to create and use mental models. Human conversation partners rely on their own mental models and their abilities to create mental models of others' mental models. While it is possible that robots could successfully fake mental models, they must rely on a system that can serve a similar purpose to mental models and appear to humans as a mental model. If robots are unable to do this, humans will never feel certain they share the same model of a goal.

Teams rely on a willingness of teammates to subjugate their own personal goals for a team goal. To do this, robots must demonstrate a sophisticated goal hierarchy and effective communication skills. Teams also depend on a high level of trust. Any breakdowns in conversations may result in the unraveling of the team. Maintaining trust requires

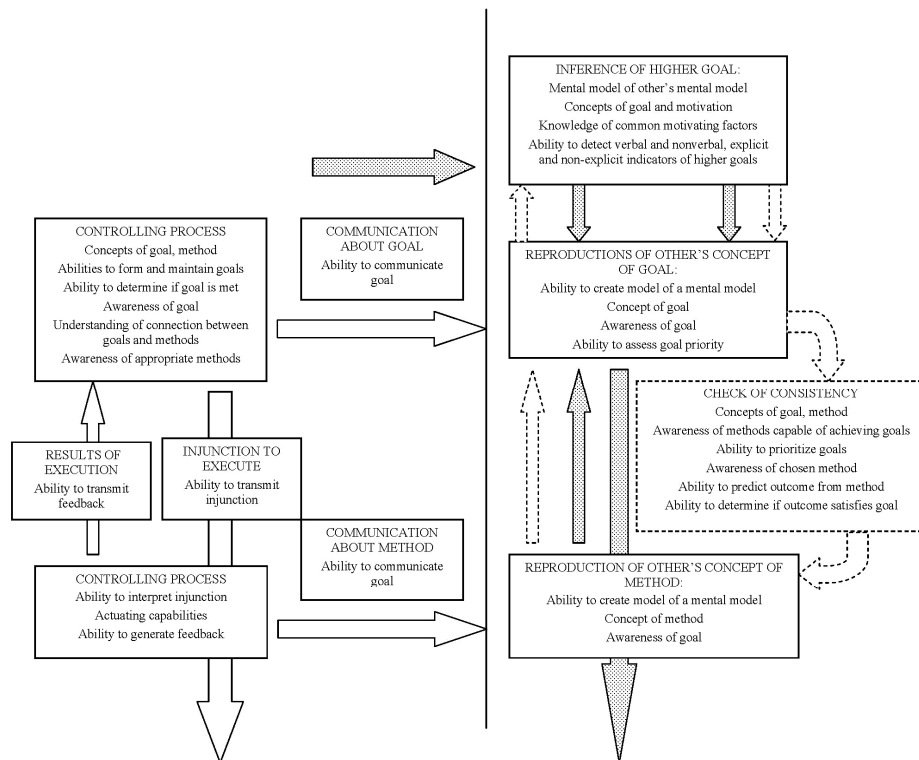


Figure 2: Model of a robot conversation partner. Robot must be capable to both communicate (left side) and interpret (right side) meaning. Robot abilities that must be developed are indicated in each box. Dotted arrows indicate inferences. Dashed arrows indicate checks of consistency.

and seek to repair the relationships (Jones & George, 1998). Even if a robot meets the basic requirements of a conversation partner, its conversational abilities will need to be further developed to meet the higher expectations of teammates.

4 CONCLUSIONS

If designers wish to place robots in roles that have previously been filled only by humans, they must design robots that demonstrate the social behavior and communication skills that humans expect of people in these roles. To create robot teammates, robots' concepts of goals, motivations, actions, and the relations between them must become further developed and nuanced. Achieving this requires the development of systems so complex that they generate behaviors that enable humans to infer the existence of shared mental models. Once researchers recognize that creating a robot teammate takes far more than improving a robot's performance and introducing it into a human team, the HRI community can weigh the challenges of developing a robot teammate to determine if creating a robot teammate is indeed the best goal to guide the direction of HRI research.

ACKNOWLEDGEMENTS

The author thanks Hugh Dubberly, Paul Pangaro, and Clifford Nass for their feedback and support.

REFERENCES

- Bettenhausen, K. L. (1991). Five years of groups research: What we have learned and what needs to be addressed. *Journal of Management*, 17(2), 345-381.
- Casper, J., & Murphy, R. R. (2003). Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 33(3), 367-385.
- Cohen, P. R., & Levesque, H. J. (1991). Teamwork. *Nous*, 25(4), 487-512.
- Fogg, B. J., & Nass, C. (1997). Do users reciprocate to computers? Paper presented at the ACM CHI.
- Forlizzi, J., & DiSalvo, C. (2006). Service robots in the domestic environment: a study of the roomba vacuum in the home. *ACM SIGCHI/SIGART Human-Robot Interaction*, 258-265.
- Groom, V., & Nass, C. (2007). Can robots be teammates? Benchmarks in human-robot teams. *Psychological Benchmarks of Human-Robot Interaction: Special issue of Interaction Studies*, 8(3), 483-500.
- Gully, S. M., Incalcaterra, K. A., Joshi, A., & Beaubien, J. M. (2002). A meta-analysis of team-efficacy, potency, and performance: Interdependence and level of analysis as moderators of observed relationships. *Journal of Applied Psychology*, 87(5), 819-832.
- Hackman, J. R. (1987). The design of work teams. In J. W. Lorsch (Ed.), *Handbook of organizational behavior*. Englewood Cliffs, NJ: Prentice Hall.
- Jones, G. R., & George, J. M. (1998). The experience and evolution of trust: implications for cooperation and teamwork. *The Academy of Management Review*, 23(3), 531-546.
- Klein, G., Woods, D. D., Bradshaw, J. M., Hoffman, R. R., & Feltovich, P. J. (2004). Ten challenges for making automation a "team player" in joint human-agent activity. *IEEE Intelligent Systems* 19 (6), 91-95.
- Lee, E.-J., Nass, C., & Brave, S. (2000). Can computer-generated speech have gender? An experimental test of gender stereotypes. Paper presented at the CHI 2000, The Hague, The Netherlands.
- Mori, M. (1970). The uncanny valley. *Energy*, 7(4), 33-35.
- Nass, C., & Brave, S. B. (2005). *Wired for speech: How voice activates and enhances the human-computer relationship*. Cambridge, MA: MIT Press.
- Nass, C., Moon, Y., & Carney, P. (1999). Are people polite to computers? Responses to computer-based interviewing systems. *Journal of Applied Social Psychology*, 29(5), 1093-1110.
- Nass, C., Steuer, J., Henriksen, L., & Dryer, D. C. (1994). Machines, social attributions, and ethopoeia: Performance assessments of computers subsequent to "self-" or "other-" evaluations. *International Journal of Human-Computer Studies*, 40(3), 543-559.
- Pask, G. (1975). *An Approach to Machine Intelligence*. In N. Negroponte (Ed.), *Soft Architecture Machines*: MIT Press.
- Rafaelli, S. (1990). Interacting with media: Para-social interaction and real interaction. In B. D. Rubin & L. Lievrouw (Eds.), *Mediation, Information, and Communication: Information and Behavior* (Vol. 3, pp. 125-181). New Brunswick, NJ: Transaction.
- Reeves, B., & Nass, C. (1996). *The media equation: how people treat computers, television, and new media like real people and places*: Cambridge University Press New York, NY, USA.
- Turkle, S. (1984). *The second self: Computers and the human spirit*. New York: Simon & Schuster.

AUTHOR INDEX

Abdel-Wahab, S.	171	Gorges, N.	214, 274
Aickelin, U.	5	Groom, V.	323
Amin, S.	171	Gruer, P.	145
Antich, J.	61	Guerrero, J.	45
Aphiratsakun, N.	151	Hamani, N.	282
Aragues, R.	252	Hebbel, M.	93
Baek, Y.	197	Herrero-Jaraba, E.	45
Barata, J.	228	Hossu, A.	140
Bernardi, R.	245	Hossu, D.	140
Bízdoacă, N.	202	Jarvis, R.	295
Blažič, S.	145	Jouffroy, G.	109
Boudaba, M.	214	Klančar, G.	145
Braga, R.	115	Knite, M.	234
Brüggemann, B.	29	Knoll, A.	301
Canureci, G.	202	Koenig, T.	288
Cardoso, J.	177	Kondo, H.	23
Carreras, M.	61	Kraemer, N.	53
Casals, A.	214	Krätzig, M.	122
Chamaret, D.	307	Kräußling, A.	29
Chatila, R.	101	Kueon, Y.	197
Chirungsarpsook, K.	151	Laue, T.	317
Contet, J.	145	Lefebvre, D.	15
Correia, L.	228	Leuxner, C.	262
Costa, P.	189	Lim, H.	23
Craye, E.	282	Madani, K.	183
Cremers, A.	29	Manfredi, J.	311
Cruz, J.	245	Matia, F.	208
Dangoumau, N.	282	Matko, D.	145
Degeratu, S.	202	Maudes, J.	268
Devy, M.	101	Mitreă, M.	239
Diaconu, I.	202	Moreira, A.	189
Diez-Pastor, J.	268	Müller, T.	301
Drighiciu, A.	202	Na, C.	85
Dumitru, O.	239	Nemec, B.	159
El-Fakdi, A.	61	Nisticò, W.	93
Ferreira, M.	177	Nunes, U.	37
Ferrier, J.	307	Obradovic, D.	85
Frese, U.	317	Ogura, Y.	23
Gaa, S.	274	Oliveira, H.	189
García-Osorio, C.	268	Omer, A.	23
Garibaldi, J.	5	Ortiz, A.	61
Gechter, F.	145	Parnichkun, M.	151
Goedemé, T.	222	Pedraza, L.	208
Goetz, F.	85	Peinado, V.	311
Gomez, J.	45	Petrişor, A.	202

AUTHOR INDEX (CONT.)

Petry, M.	115	Xiao, L.	133
Podins, G.	234	Yan, J.	183
Ponweiser, W.	165	Yin, C.	133
Potts, A.	245	Yoon, H.	197
Prêteux, F.	239	Yu, W.	183
Puente, P.	208	Zavickis, J.	234
Rauchhaupt, L.	122	Zeiger, F.	53, 69
Reis, L.	115	Zheng, D.	133
Richard, P.	307	Ziaie, P.	301
Richtsfeld, M.	165	Zike, S.	234
Rodríguez, J.	268	Zlajpah, L.	159
Rodríguez-Losada, D.	208	Zureiki, A.	101
Rückert, U.	171		
Rungger, M.	262		
Sabourin, C.	183		
Sagues, C.	252		
Salgueiro, M.	228		
Santana, P.	228		
Santos, C.	177		
Santos, V.	228		
Sauer, M.	69		
Scheiterer, R.	85		
Schilling, K.	53, 69		
Schulz, D.	29		
Serb, G.	202		
Simonin, O.	145		
Sitou, W.	262		
Solea, R.	37		
Sousa, A.	189		
Souza, G.	288		
Spanfelner, B.	262		
Steindl, G.	85		
Stursberg, O.	262		
Takanishi, A.	23		
Tanoto, A.	171		
Tsourveloudis, N.	77		
Tupureina, V.	234		
Verdasca, J.	311		
Vincze, M.	165		
Vitzilaios, N.	77		
Whitbrook, A.	5		
Witkowski, U.	171		
Woern, H.	214		
Wörn, H.	274		



Proceedings of ICINCO 2008
Fifth International Conference on Informatics in Control, Automation and Robotics
ISBN: 978-989-8111-31-9
<http://www.icinco.org>