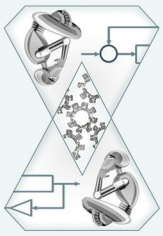


ICINCO 2008

FIFTH INTERNATIONAL CONFERENCE ON INFORMATICS IN CONTROL, AUTOMATION AND ROBOTICS

Proceedings

Robotics and Automation - Vol. 2

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

CO-ORGANIZED BY

INSTICC

UNIVERSIDADE da MADEIRA

IN COOPERATION WITH

aaai

CO-SPONSORED BY

IEEE

IEEE Systems, Man, and Cybernetics (SMC) Society

IFAC

INSTICC PRESS

ICINCO 2008

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

Volume RA-2

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

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

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 Superieure 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, Thecnical 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 Hagras, University of Essex, U.K.

Wolfgang Halang, Fernuniversitaet, 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 Kozlowski, 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 Nedevschi, 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. Nolazco-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, Universitá 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 Techology, Australia

Marek Zaremba, Universite du Quebec, 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

SPECIAL SESSION ON SERVICE ORIENTED ARCHITECTURES FOR SMEROBOTS AND PLUG-AND-PRODUCE

SPECIAL SESSION ON MULTI-AGENT ROBOTIC SYSTEMS

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 Estimaton 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 languages 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 (Torrisi 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 being 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 where 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:

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

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

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

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, $i(k) \in I = \{1,\ldots,s\}$ 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 *pdf*s 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:

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

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

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:

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

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 $\mathbf{i}_T = \mathbf{i}_T(k) \triangleq \{i(k),\ldots,i(k+T-1)\}$. 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:

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

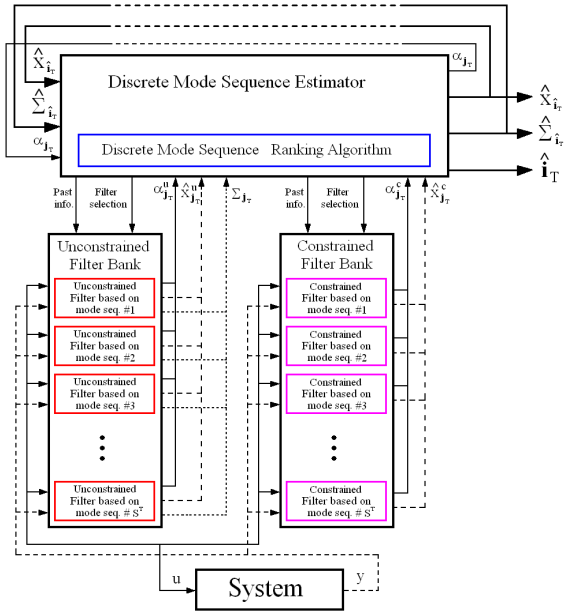


Figure 1: Interacting Multiple-Model Estimation Algorithm.

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 $Y_T(k) \triangleq [y(k)^{\mathrm{T}},\ldots,y(k+T-1)^{\mathrm{T}}]^{\mathrm{T}}$. 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 $\Omega_{\mathbf{i}_T}$:

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

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):

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

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

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 recomputed 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* $\mathbf{j}_T$:

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

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

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:

$$\mathbf{K}_{\mathbf{j}_T}(k|t-1) = \left(\Sigma^{-1}_{x_{\mathbf{j}_T}}(k|t-1)+\mathbf{C}^{\mathrm{T}}_{\mathbf{j}_T}\Sigma^{-1}_{Y_{\mathbf{j}_T}}\mathbf{C}_{\mathbf{j}_T}\right)^{-1}\mathbf{C}^{\mathrm{T}}_{\mathbf{j}_T}\Sigma^{-1}_{Y_{\mathbf{j}_T}} \tag{10}$$

$$\Sigma_{Y_{\mathbf{j}_T}} = \left[\mathbf{L}_{\mathbf{j}_T}\ I_{T.n_y}\right]\begin{bmatrix}\Sigma_{W_{\mathbf{j}_T}} & 0\\ 0 & \Sigma_{V_{\mathbf{j}_T}}\end{bmatrix}\left[\mathbf{L}_{\mathbf{j}_T}\ I_{T.n_y}\right]^{\mathrm{T}} \tag{11}$$

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

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

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

$$\Sigma_{x_{\mathbf{j}_T}}(k+1|t) = A_{j(k)}\Sigma_{x_{\mathbf{j}_T}}(k|t)A^{\mathrm{T}}_{j(k)} + L_{j(k)}\Sigma_{wj(k)}L^{\mathrm{T}}_{j(k)} \tag{13}$$

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

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:

$$\begin{bmatrix} x_{\mathbf{j}_T}(k) \\ W_{\mathbf{j}_T}(k) \\ V_{\mathbf{j}_T}(k)\end{bmatrix} \tag{14}$$

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:

$$\mathbf{H_e}\ .\begin{bmatrix} x_{\mathbf{j}_T}(k) \\ W_{\mathbf{j}_T}(k) \\ V_{\mathbf{j}_T}(k)\end{bmatrix} = \mathbf{h_e} \quad \Leftrightarrow \tag{15}$$

$$\Leftrightarrow\ \left[\mathbf{C}_{\mathbf{j}_T}\ \mathbf{L}_{\mathbf{j}_T}\ I_{n_Y}\right].\begin{bmatrix} x_{\mathbf{j}_T}(k) \\ W_{\mathbf{j}_T}(k) \\ V_{\mathbf{j}_T}(k)\end{bmatrix} = \left[Y_T(k)-\mathbf{D}_{\mathbf{j}_T}U_T(k)-\mathbf{g}_{\mathbf{j}_T}\right]$$

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:

$$\mathbf{H_i}\ .\begin{bmatrix} x_{\mathbf{j}_T}(k) \\ W_{\mathbf{j}_T}(k) \\ V_{\mathbf{j}_T}(k)\end{bmatrix} \leq \mathbf{h_i} \quad \Leftrightarrow \tag{16}$$

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

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

$$\begin{bmatrix} \hat{x}_{\mathbf{j}_T}(k|t) \\ \hat{W}_{\mathbf{j}_T}(k|t) \\ \hat{V}_{\mathbf{j}_T}(k|t)\end{bmatrix} = \begin{bmatrix} \hat{x}_{\mathbf{j}_T}(z,k|t-1) \\ \hat{W}_{\mathbf{j}_T}(k|t-1) \\ \hat{V}_{\mathbf{j}_T}(k|t-1)\end{bmatrix} + \mathbf{K}_{\mathbf{j}_T}(k|t)\left(\begin{bmatrix}\mathbf{h_e}\\ \mathbf{h_i}\end{bmatrix}-\begin{bmatrix}\mathbf{H_e}\\ \mathbf{H_i}\end{bmatrix}.\begin{bmatrix} \hat{x}_{\mathbf{j}_T}(k|t-1) \\ \hat{W}_{\mathbf{j}_T}(k|t-1) \\ \hat{V}_{\mathbf{j}_T}(k|t-1)\end{bmatrix}\right) \tag{17}$$

The constrained least-squares filter gain is defined as:

$$\mathbf{K}_{\mathbf{j}_T}(k|t) = \tag{18}$$

$$\left(\begin{bmatrix}\Sigma_{x_{\mathbf{j}_T}}(k|t-1) & 0 & 0\\ 0 & \Sigma_{W_{\mathbf{j}_T}} & 0\\ 0 & 0 & \Sigma_{V_{\mathbf{j}_T}}\end{bmatrix}^{-1}+\begin{bmatrix}\mathbf{H_e}\\ \mathbf{H_i}\end{bmatrix}^{\mathrm{T}}\mathbf{Z}_{\mathbf{j}_T}(k|t)\begin{bmatrix}\mathbf{H_e}\\ \mathbf{H_i}\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{H_e}\\ \mathbf{H_i}\end{bmatrix}^{\mathrm{T}}\mathbf{Z}_{\mathbf{j}_T}(k|t)$$

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

There are several methods, most of them iterative, for determining the matrix $\mathbf{Z}_{\mathbf{j}_T}(k|t)$, 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 $\Sigma^{-1}_{x_{\mathbf{j}_T}}(k|t-1)$ to 0. The corresponding constrained state estimate is referred to as $\hat{x}^{c*}_{\mathbf{j}_T}(k|t)$.

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, $\alpha^c_{\mathbf{j}_T}$:

$$\hat{\mathbf{i}}_T(k|t) = \arg\min_{\mathbf{j}_T} \alpha^c_{\mathbf{j}_T}(k|t) \tag{19}$$

The squared error associated with the *dms* $\mathbf{j}_T$ is given by:

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

where:

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

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

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, $\bar{\alpha}_{\mathbf{j}_T}$, 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 $\bar{\alpha}_{\mathbf{j}_T}$ should be defined as:

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

Equation (20) should be modified to:

$$\alpha_{\mathbf{j}_T}(k|t) = \bar{\alpha}_{\mathbf{j}_T} + \left\|\hat{Y}^*_{\mathbf{j}_T}(k|t) - Y_T(k)\right\|^2_{\Sigma^{-1}_{Y_{\mathbf{j}_T}}} \tag{23}$$

Equation (23) can be used to compute the squared errors of both the unconstrained estimates, $\alpha^u_{\mathbf{j}_T}(k|t)$, and the constrained estimates, $\alpha^c_{\mathbf{j}_T}(k|t)$, using $\hat{x}^{u*}_{\mathbf{j}}(k|t)$ and $\hat{x}^{c*}_{\mathbf{j}}(k|t)$, 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 $\hat{x}^{u*}_{\mathbf{j}_T}(k|t)$ can be determined by simple matrix sums and multiplications if the filter gain $\mathbf{K}_{\mathbf{j}_T}(k|t-1)$ is computed off-line, since there are no varying terms as can be seen in equation (9). The corresponding squared error $\alpha^u_{\mathbf{j}_T}(k|t)$, computed through equation (23), can also be determined using simple matrix sums and multiplications from $\hat{x}^{u*}_{\mathbf{j}_T}(k|t)$. The continuous state estimate $\hat{x}^u_{\mathbf{j}_T}(k|t)$ on the other hand, requires a matrix inversion to determine the corresponding filter gain using equation (10) since the matrix $\Sigma^{-1}_{x_{\mathbf{j}_T}}(k|t-1)$ 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 $\alpha^c_{\mathbf{j}_T}(k|t)$ can be determined using simple matrix sums and multiplications from $\hat{x}^{c*}_{\mathbf{j}}(k|t)$.

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:

$$\alpha^u_{\mathbf{j}_T}(k|t) \leq \alpha^c_{\mathbf{j}_T}(k|t) \tag{24}$$

An efficient reduction on the number of constrained estimates that have to be computed can be achieved by computing all unconstrained estimates $\hat{x}^{u*}_{\mathbf{j}}(k|t)$ and the corresponding squared errors $\alpha^u_{\mathbf{j}_T}(k|t)$ 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 $\alpha^c_{\mathbf{i}_T}$ under which $\alpha^c_{\mathbf{i}_T}$ is always smaller than $\alpha^c_{\mathbf{j}_T}$, 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):

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

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 $\alpha^c_{\mathbf{i}_T}(k|t)$ 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*, $\alpha^c_{\mathbf{j}_T}(k|t)$ and, the continuous state estimates $\hat{x}^{c*}_{\mathbf{j}_T}(k|t)$ 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:

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

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:

$$\begin{aligned} & \gamma_{\mathbf{j}_T,j}(t+1|t) = \\ & \max\Big(S_j\,\hat{x}^*_{\mathbf{j}_T}(t+1|t) + R_j u(t+1) + Q_j \hat{w}(t+1|t) - T_j\Big) \end{aligned} \tag{27}$$

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

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 $\mathbf{j}_T = \big\{j(k+1), \ldots, j(t), j(t+1)\big\}$ 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.

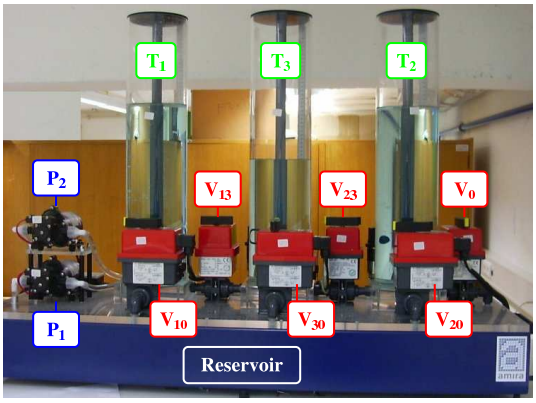


Figure 2: The three-tanks setup.

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" 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.

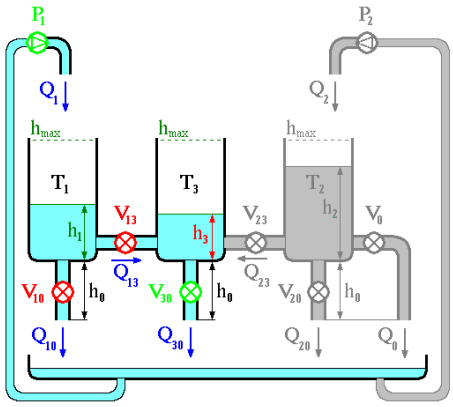


Figure 3: Final model of the three-tanks system.

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 $\mathbf{V_{10}}$

In this example, the estimation algorithm will have to estimate the discrete mode that indicates a fault on valve $\mathbf{V_{10}}$. As the analysis will focus on valve $\mathbf{V_{10}}$, the faults on valve $\mathbf{V_{13}}$ and sensor $\mathbf{h_3}$ 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 $\mathbf{V_{10}}$ 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
$\mathbf{V_{10}}$	"ok"	"faulty" *med*	"faulty" *max*	"faulty" *med*	"faulty" *max*	"ok"
$\mathbf{V_{13}}$	"ok"	"ok"	"ok"	"ok"	"ok"	"ok"
$\mathbf{h_3}$	"ok"	"ok"	"ok"	"ok"	"ok"	"ok"
$\mathbf{P_1}$	"on"	"on"	"on"	"on"	"on"	"on"
$\mathbf{V_{13}}$	"open"	"open"	"open"	"open"	"open"	"open"
$\mathbf{V_{30}}$	"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 $\mathbf{V_{10}}$ 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.

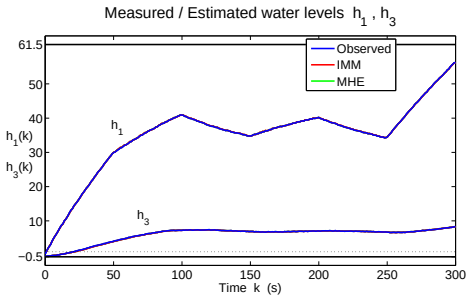


Figure 4: Water levels estimation using the IMM and MHE estimation algorithms.

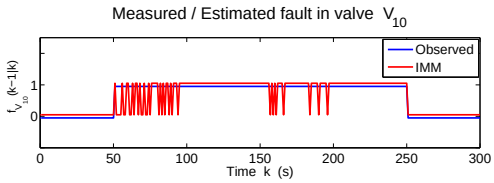


Figure 5: Estimation of the discrete mode sequence relative to the fault in valve $\mathbf{V_{10}}$.

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 $\frac{0.8^2}{12}$ cm^2 for all k, where 0.8 is the maximum water level change when the valve $\mathbf{V_{10}}$ is fully open.

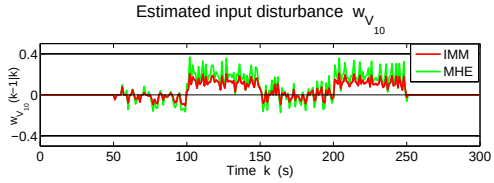


Figure 6: Estimation of the input disturbance $w_{V_{10}}(k-1|k)$ corresponding to the fault in valve $\mathbf{V_{10}}$.

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.

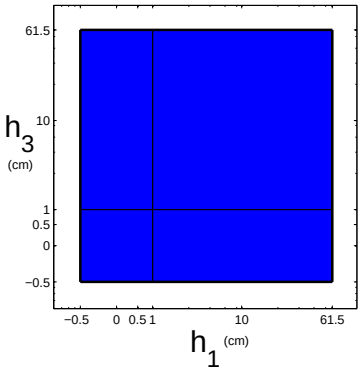


Figure 7: Map of probability of correct mode estimation, with 1s delay, when valve $\mathbf{V_{10}}$ is fully closed. (Red - probability of correct mode estimation 0, Blue - probability of correct mode estimation 1)

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

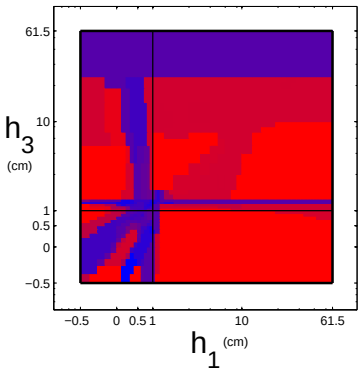


Figure 8: Map of probability of correct mode estimation, with 1s delay, when fault $f_{V_{10}}$ is active. (Red - probability of correct mode estimation 0, Blue - probability of correct mode estimation 1)

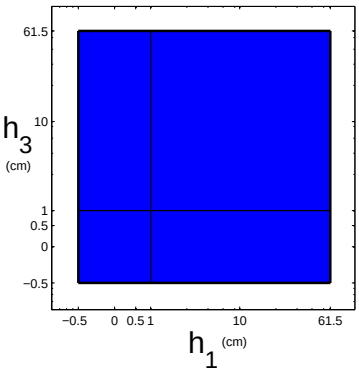


Figure 9: Map of probability of correct mode estimation, with 1s delay, considering that fault $f_{V_{10}}$ has maximum intensity, $w_{V_{10}} = 0.4$. (Red - probability of correct mode estimation 0, Blue - probability of correct mode estimation 1)

4 that the valve $\mathbf{V_{10}}$ 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 $\mathbf{V_{10}}$ 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 $\mathbf{V_{10}}$ 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.

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 tipically 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. - Lincon 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, 2^{nd} 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 Tcnico, Tecnhical 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 Tcnico, 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.

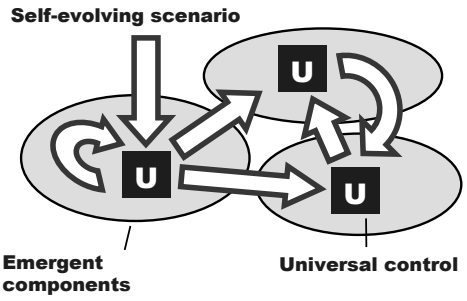


Figure 1: Runtime coverage of a distributed system.

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).

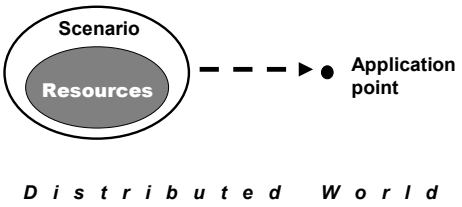


Figure 2: The initial state.

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.

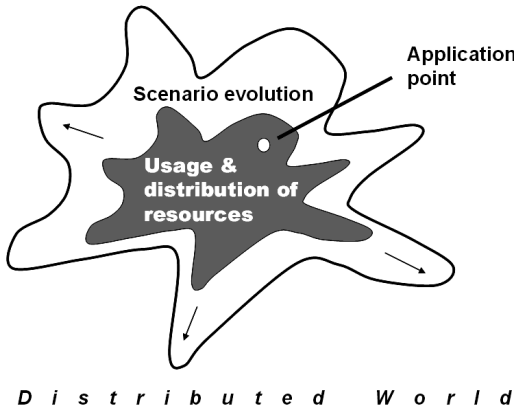


Figure3: Spreading operations and resources.

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:

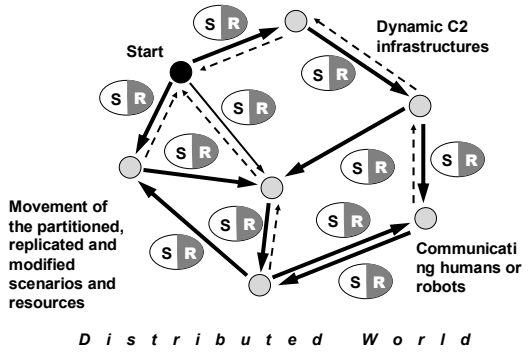


Figure 4: Resultant interactions between system parts.

- 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 that itself (here the content, or name, of each node is considered as its rank).

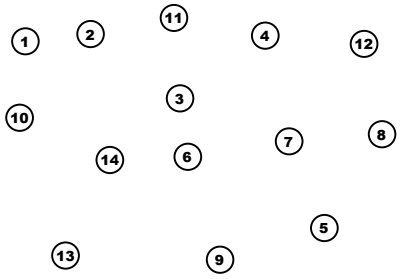


Figure 5: Distributed system nodes.

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.

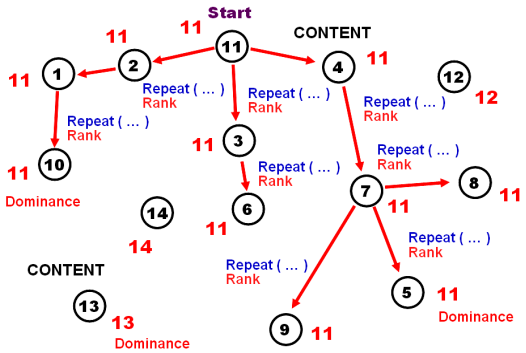


Figure 6: Resulting in partial dominance.

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.

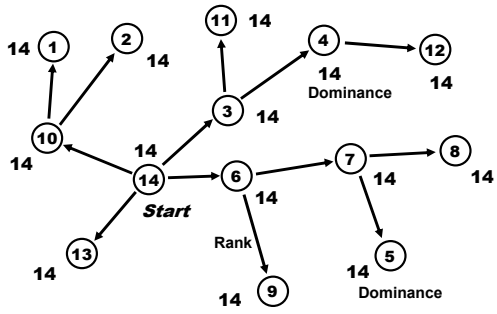


Figure 7: Setting absolute dominance of node 14.

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.

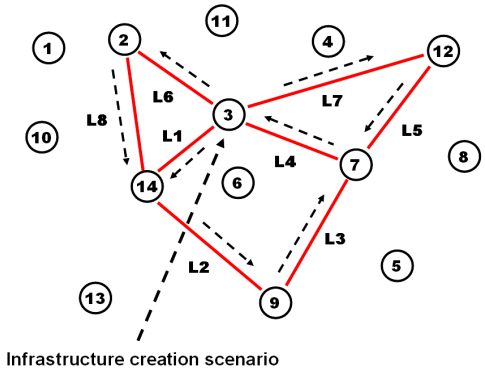


Figure 8: Creating a distributed infrastructure.

```
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.

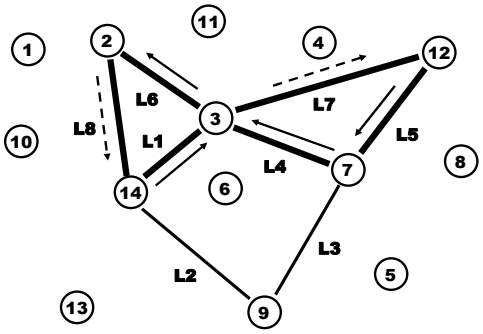


Figure 9: Finding all triangles in the infrastructure.

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").

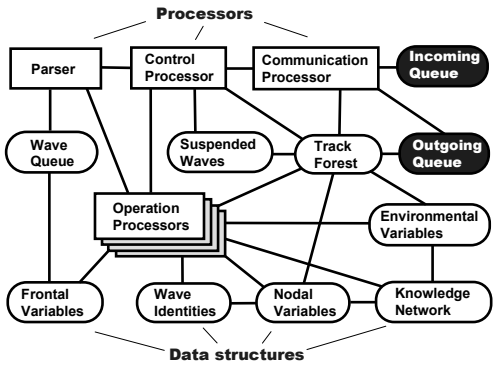


Figure 10: The WPL interpreter architecture.

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.

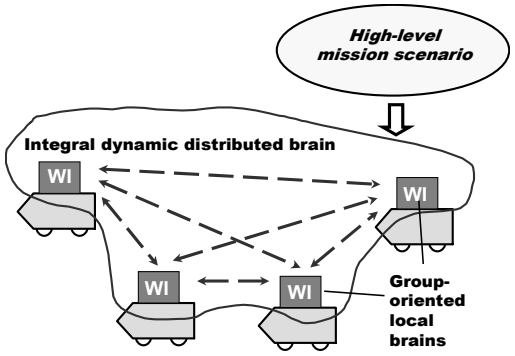


Figure 11: WPL interpreters (WI) forming distributed robotic brain.

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.

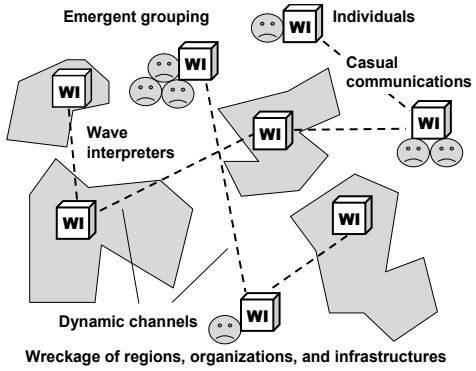


Figure 12: A disaster area with WPL interpreters embedded.

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)))))
```

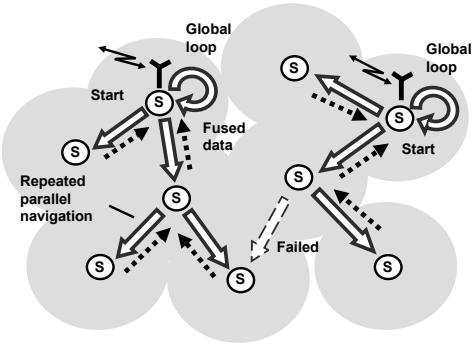


Figure 13: Collecting data by a sensor network.

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)
```

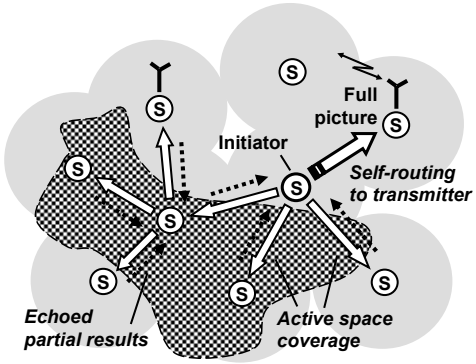


Figure 14: Space coverage with hierarchical assembling of a distributed phenomenon.

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.

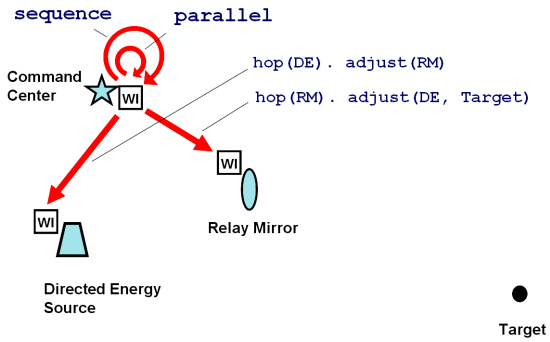


Figure 15: DE system operation, Snapshot 1.

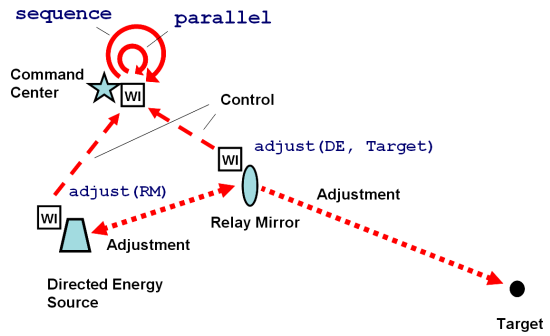


Figure 16: DE system operation, Snapshot 2.

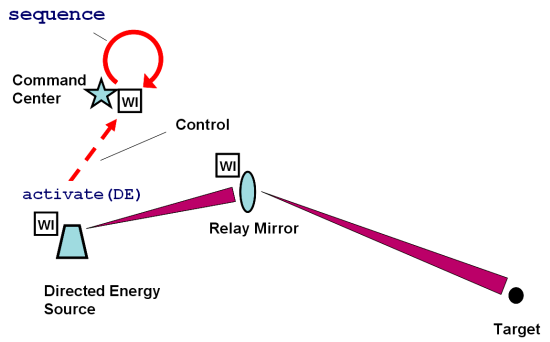


Figure 17: DE system operation, Snapshot 3.

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)
```

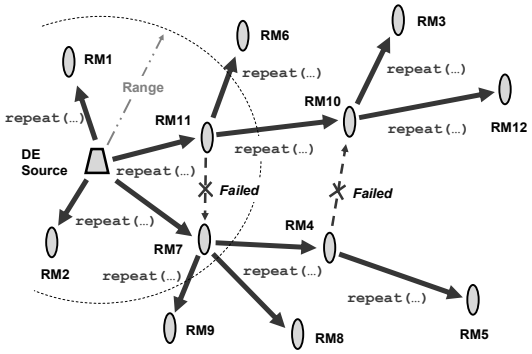


Figure 18: Dynamic shortest path tree through all RMs.

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)
```

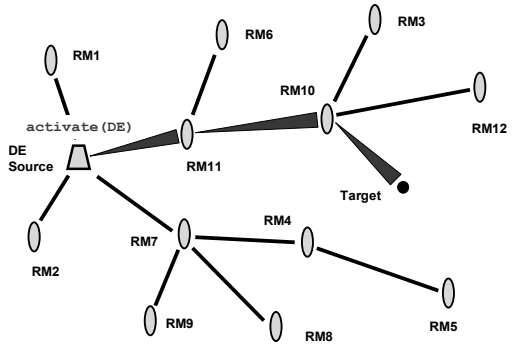


Figure 19: Energy delivery via the path found.

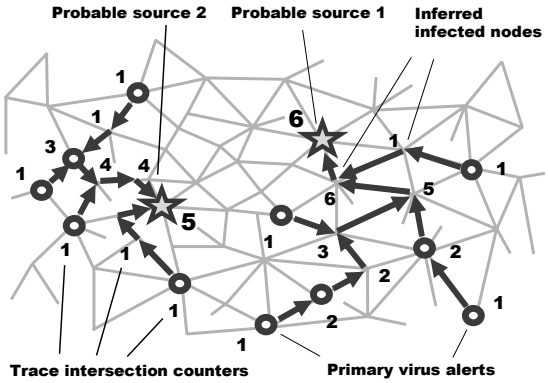


Figure 20: Finding virus sources in parallel.

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)))
```

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.

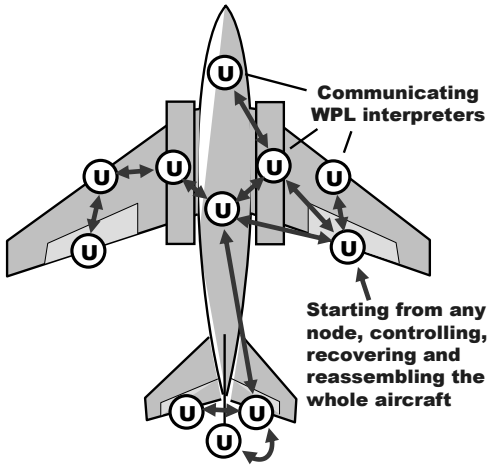


Figure 21: Aircraft self-analysis by the WPL network.

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).

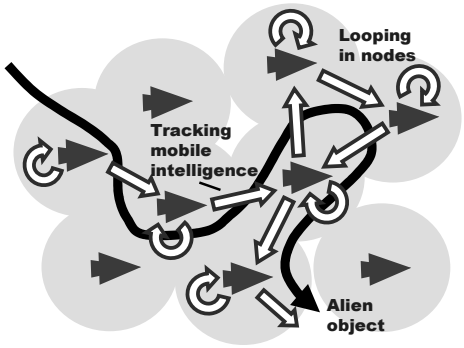


Figure 22: Collective tracking of a mobile object.

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.

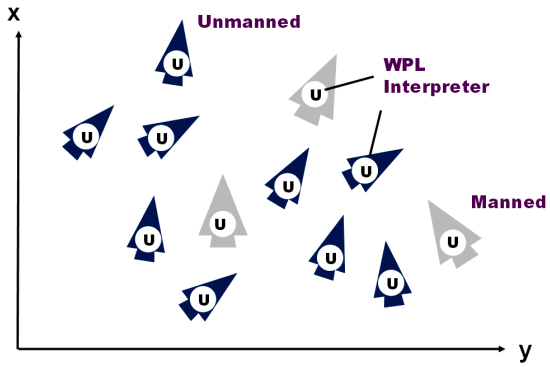


Figure 23: Initial distribution of units.

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.

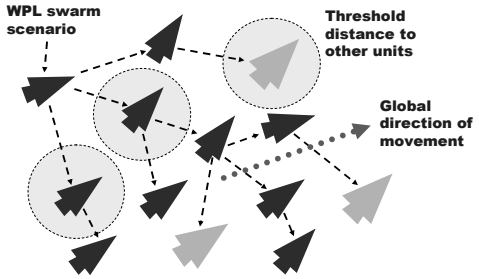


Figure 24: A swarm movement snapshot.

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)
```

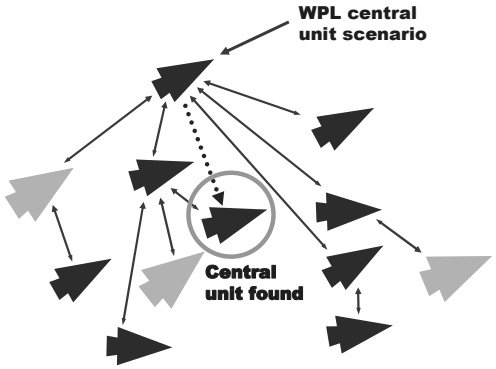


Figure 25: Finding topologically central unit.

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)))
```

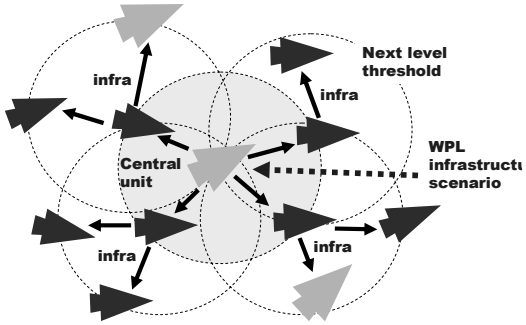


Figure 26: Hierarchical infrastructure built.

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))
```

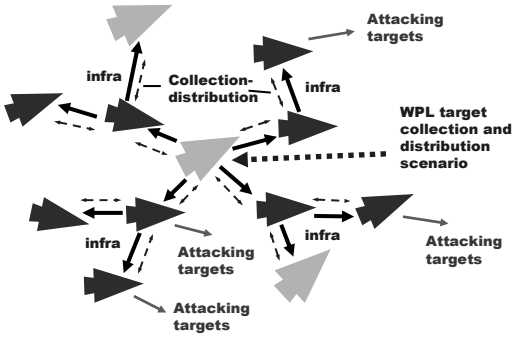


Figure 27: Hierarchical fusion and distribution of targets.

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 - 2

POSTERS

REAL TIME TRACKING OF AN OMNIDIRECTIONAL ROBOT
An Extended Kalman Filter Approach

José Gonçalves, José Lima
Department of Electrical Engineering, Polytechnic Institute of Bragança, Portugal
{*goncalves, jllima*}*@ipb.pt*

Paulo Costa
Deec, Faculty of Engineering of the University of Porto, Portugal
paco@fe.up.pt

Keywords: Probabilistic robotics, Kalman Filter, Omnidirectional robots.

Abstract: This paper describes a robust localization system, similar to the used by the teams participating in the Robocup Small size league (SLL). The system, developed in Object Pascal, allows real time localization and control of an autonomous omnidirectional mobile robot. The localization algorithm is done resorting to odometry and global vision data fusion, applying an extended Kalman filter, being this method a standard approach for reducing the error in a least squares sense, using measurements from different sources.

1 INTRODUCTION

Soccer was the original motivation for Robocup. Besides being a very popular sport worldwide, soccer brings up a set of challenges for researchers while attracting people to the event, promoting robotics among students, researchers and general public. RoboCup chose to use soccer game as a central topic of research, aiming at innovations to be applied for socially significant problems and industries (rob, 2008). As robotics soccer is a challenge in an highly dynamic environment, the robot and ball position information must be accessible as fast and accurate as possible (Sousa, 2003). As an example if the ball has a velocity of 2 ms^{-1} and if the lag time is 100 ms, the ball will travel a distance of 20 cm between two sampling instants, compromising the controller performance (Gonçalves et al., 2007). The presented localization algorithm is updated 25 times per second, fulfilling the proposed real time requisites.
Robots maintain a set of hypotheses with regard to their position and the position of different objects around them. The input for updating these beliefs come from poses belief and various sensors (Borestein et al., 1996). An optimal estimation can be applied in order to update their beliefs as accurately as possible. After one action the pose belief is updated based on data collected up to that point in time, by a process called filtering (Thrun et al., 2005).

2 RELATIVE POSITION ESTIMATION

Omnidirectional vehicles are widely used in robotics soccer, allowing movements in every direction, where the extra mobility is an important advantage. The fact that the robot is able to move from one place to another with independent linear and angular velocities contributes to minimize the time to react, the number of maneuvers is reduced and consequently the game strategy can be simplified (Ribeiro et al., 2004). The omnidirectional robots use special wheels, that allow movements in every direction. The movement of these robots does not have the restraints of the differential robots (Dudek and Jenkin, 2000), presenting the disadvantage of a more complex control. It is possible to conclude from the geometry of a three wheel omnidirectional robot, presented in Figure 1, that the velocities V_x, V_y and w vary with the linear velocities V_1, V_2 and V_3, as shown in equations system (1) (Kalmár-Nagy et al., 2002).

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} -sin(\theta) & cos(\theta) & L \\ -sin(\frac{\pi}{3}-\theta) & -cos(\frac{\pi}{3}-\theta) & L \\ sin(\frac{\pi}{3}+\theta) & -cos(\frac{\pi}{3}+\theta) & L \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ w \end{pmatrix} \quad (1)$$

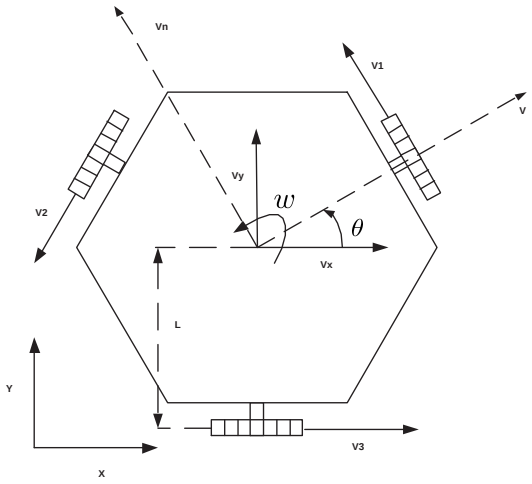


Figure 1: Geometry of a three wheel omnidirectional robot.

2.1 Odometry Calculation

The robot relative position estimation is based on the odometry calculation. The odometry calculation uses each wheel velocity in order to estimate the robot position, the disadvantage is that the position estimate error is cumulative and increases over time.

The robot kinematic equations can be represented by the equations system (2), in alternative to the equations system (1).

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & L \\ -sin(\frac{\pi}{3}) & -cos(\frac{\pi}{3}) & L \\ sin(\frac{\pi}{3}) & -cos(\frac{\pi}{3}) & L \end{pmatrix} \begin{pmatrix} V \\ V_n \\ w \end{pmatrix} \quad (2)$$

The linear and angular velocities V, V_n and w can be obtained rewriting equations system (2) as equations system (3),

$$\begin{pmatrix} V \\ V_n \\ w \end{pmatrix} = \begin{pmatrix} G \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} \quad (3)$$

where G is :

$$\begin{pmatrix} 0 & \frac{-1}{2\sin(\frac{\pi}{3})} & \frac{1}{2\sin(\frac{\pi}{3})} \\ \frac{1}{1+\cos(\frac{\pi}{3})} & \frac{-1}{2(1+\cos(\frac{\pi}{3}))} & \frac{-1}{2(1+\cos(\frac{\pi}{3}))} \\ \frac{\cos\frac{\pi}{3}}{L(1+\cos(\frac{\pi}{3}))} & \frac{1}{2L(1+\cos(\frac{\pi}{3}))} & \frac{1}{2L(1+\cos(\frac{\pi}{3}))} \end{pmatrix} \quad (4)$$

By this way θ can be found, applying an first order approximation, as shown in equation (5),

$$\theta(K) = \theta(K-1) + wT \quad (5)$$

where T is the sampling time.

After θ calculation an rotation matrix, presented in matrix (6), is applied in order to obtain V_x and V_y, as shown in equations system (7),

$$B = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

$$\begin{pmatrix} V_x \\ V_y \\ w \end{pmatrix} = BG \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} \quad (7)$$

x and y estimate is calculated applying an first order approximation, as shown in equations (8) and (9),

$$x(K) = x(K-1) + V_x T \quad (8)$$

$$y(K) = y(K-1) + V_y T \quad (9)$$

where T is the Sampling Time.

2.2 Odometry Error Study

With the objective of evaluate the odometry error a robot race was made, as shown in the flowchart presented in Figure 2. It is possible to observe that the robot moves across several locations, executing the trajectory presented in Figures 3, 4, 5 and 6.

The goal of the controller is to move the robot to a target position with controlled velocity. As input parameters we have as goal the robot displacement to a target position. Initially a position vector pointing to the target position is calculated, the position vector is normalized converting it into a velocity vector, becoming this the objective to accomplish. The equations (1) are used to calculate the velocity that each wheel must have in order to accomplish the objective. At each sampling time the estimated position changes, consequently the position vector changes, the velocity vector changes and the reference speed of each motor changes. The controller has also as objective to follow the trajectory with an angle near to zero. One important fact that needs to be enhanced from the graphics presented in Figures 3, 4 and 5, is that when it is expected the robot to pass by the position $x = 20$ cm and $y = -20$ cm, the robot starts to move to the next target position. This happens because the objective of reaching one position is accomplished if the error in x and in y is less than 2 cm, making the state machine evolve to the next state, changing the objective to $x = 20$ cm and $y = 20$ cm.

The presented graphics (Figures 2, 4, 5, 6 and 7), allow to obtain the odometry error model, relating the acceleration with the odometry error. If the robot accelerates there is an error increase because the acceleration forces the wheels to slip. This situation is

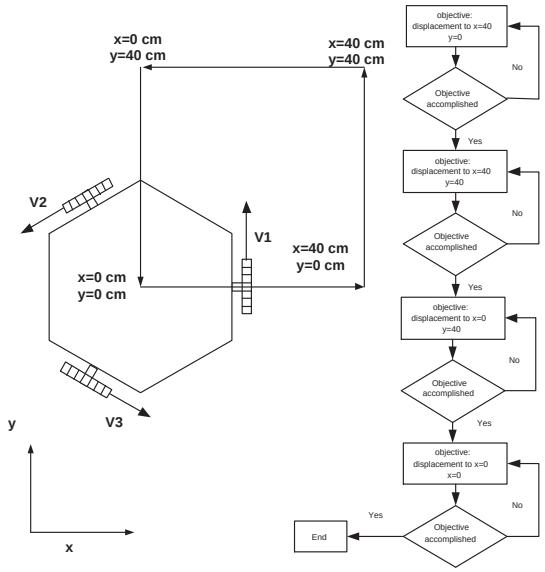


Figure 2: Flowchart of the robot race.

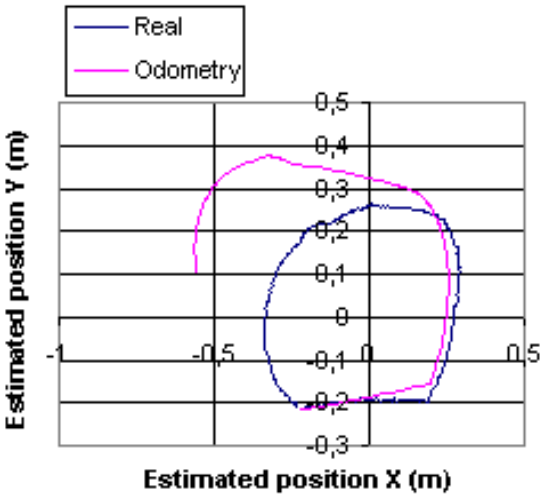


Figure 3: Estimated and real robot trajectory.

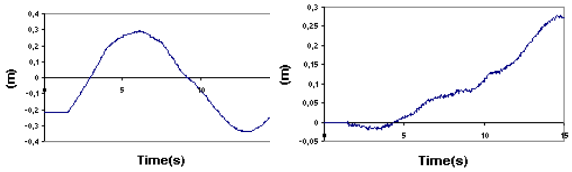


Figure 4: a) Real position x, and (b) Odometry x error.

more observable when the robot trajectory changes its direction, causing the robot to decelerate and to accelerate causing disturbances in the robot angle, increasing the angle error and consequently changing the rate

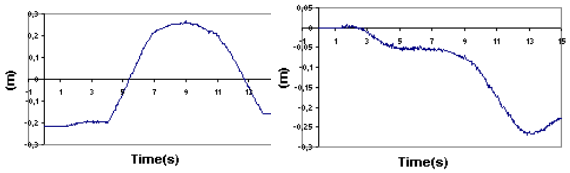


Figure 5: a) Real position y, and (b) Odometry y error.

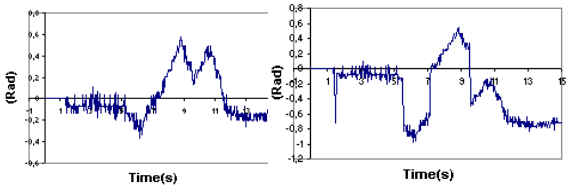


Figure 6: a) Real angle, and (b) Odometry angle error.

that the estimate position error evolves.

By this way, if the robot is not moving the variance for x and y is considered null and if the robot is moving the used odometry variance error model for x and y is:

$$Var_{xy} = K_1 + K_2 la(k-1)^2 \qquad (10)$$

- K_1 is the variance when the robot is moving in steady state.
- K_2 is a constant that relates the variance with the previous sample time acceleration $la(k-1)$.

The previous sample time acceleration is used instead of the present sample time because it is more representative to evaluate the odometry error noise, because the encoder transitions (necessary to calculate each well velocity estimation resorting to an first order approximation) are updated from the previous sample time up to the present sample time. The angle variance is modeled in a similar way.

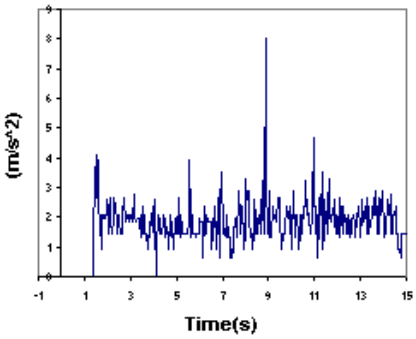


Figure 7: Linear acceleration modulus.

3 ABSOLUTE POSITION ESTIMATION

The Global vision system is required to detect and track a mobile robot in an area supervised by one camera. The camera is placed perpendicular to the ground, fixed to an metallic structure, allowing a maximal height of 3 meters, although in the presented case is placed only at 2 meters height. Placing the camera higher reduces the parallax error, reduces problems such as ball occlusion and the vision field increases, although the image quality decreases and the error due to the barrel distortion effect increases. The image quality concept, in this case study, is related with the number of pixels that are available at each frame to identify and localize an robot marker. The markers, placed on the robot top, have the goal to provide information about the robot localization. Their geometric shape is a circle, all with the same dimensions and with different colors. The number of observed pixels for each marker depends on the illumination conditions, color calibration and camera height. If the camera is placed higher the vision field is bigger, consequently the maximum number of observed pixels for each marker will be reduced.

3.1 Robot Localization

Knowing at first hand that are necessary to localize the robot two different markers, one to identify the center and another to provide information for the angle calculation. Being the field green, and the ball orange, the colors for the robot markers should be the most distant as possible in the RGB cube. The chosen colors were blue for the robot center and yellow for the angle, being the official Robocup colors to distinguish two teams in the SSL placing a colored marker in each robot center (rob, 2008). The ball localization is achieved the same way as the robot center, the only difference is that is a marker placed at a different height and associated to a different color. The blue and yellow markers are used for the robot detection and localization. The blue marker allows x and y calculation, and the yellow marker allows the angle calculation. The used robot is illustrated in Figure 8.

3.2 Global Vision Error Study

It was made for the global vision localization system an analysis of the error probability distributions (Thrun et al., 2005)(Choset et al., 2005). The position error probability distributions were approximated to Gaussian distributions (Ribeiro, 2004)(Negenborn, 2003), being the results presented in SI units. The

Figure 8: Omnidirectional robot prototype.

number of obtained pixels for the blue marker ($Q1$), affects the error variance in x and y, as shown in the next table:

Table 1.

$Q1$	x	y
5-10	1,5E-05	1,9E-05
10-20	9,25E-06	7,36E-06
20-30	4,84E-06	4,86E-06
30-40	4,15E-06	3,80E-06
≥ 40	1,96E-06	2,21E-06

On the other hand the variance of the angle error probability distribution is affected by the number of pixels obtained for both makers, for the blue ($Q1$) and for the yellow ($Q2$), as shown in the next tables:

Table 2.

$Q1$ / $Q2$	5-10	10-20	20-30	30-40
5-10	0,14	8E-02	1,2E-02	1E-02
10-20	1,6E-02	9,9E-03	1,3E-02	6,6E-03
20-30	1,5E-02	9,9E-03	7,2E-03	4,9E-03
30-40	1,4E-02	9,5E-03	5,9E-03	4,4E-03
≥ 40	1,4E-02	7,2E-03	5,77E-03	3E-03

Table 3.

$Q1$ / $Q2$	≥ 40
5-10	6,2E-03
10-20	4,6E-03
20-30	3,9E-03
30-40	2,9E-03
≥ 40	3E-03

4 ODOMETRY AND GLOBAL VISION DATA FUSION

Odometry and global vision data fusion was achieved applying an extended Kalman filter. This method was chosen because the robot motion equations are non-linear and also because the measurements error probability distributions can be approximated to Gaussian distributions (Choset et al., 2005).

4.1 Extended Kalman Filter Algorithm

With the dynamic model given by equations system (7) and considering that control signals change only at sampling instants, the state equation is:

$$\frac{dX(t)}{dt} = f(X(t), u(t_k), t), t\varepsilon[t_k, t_{k+1}] \qquad (11)$$

Where $u(t) = [V_1 V_2 V_3]^T$, that is, the odometry measurements are used as kinematic model inputs. This state should be linearized over $t = t_k$, $X(t) = X(t_k)$ and $u(t) = u(t_k)$, resulting in:

$$A^*k = \begin{pmatrix} 0 & 0 & \frac{-\sin(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\cos(\theta)}{2(1+\cos(\frac{\pi}{3}))} \\ 0 & 0 & \frac{\cos(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\sin(\theta)}{2(1+\cos(\frac{\pi}{3}))} \\ 0 & 0 & 0 \end{pmatrix} \qquad (12)$$

with state transition matrix:

$$\phi^*(k) = exp(A^*(k)(t_k - t_{k-1})) \qquad (13)$$

Resulting in:

$$\phi^*k = \begin{pmatrix} 1 & 0 & (\frac{-\sin(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\cos(\theta)}{2(1+\cos(\frac{\pi}{3}))})T \\ 0 & 1 & (\frac{\cos(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\sin(\theta)}{2(1+\cos(\frac{\pi}{3}))})T \\ 0 & 0 & 1 \end{pmatrix} \qquad (14)$$

Where T is the sampling time $(t_k - t_{k-1})$.

Thus the observations are obtained directly, H^* is the identity matrix.

The extended Kalman filter algorithm steps are as follows (Welch and Bishop, 2001):

1. State estimation at time $t = t_k, X(k^-)$, knowing the previous estimate at $t = t_{k-1}$, $X(k-1)$ and control $u(t_k)$, calculated by numerical integration as shown in equations (5), (8) and (9).
2. Propagation of the state covariance

$$P(k^-) = \phi^*(k)P(k-1)\phi^*(k)^T + Q(k) \qquad (15)$$

Where $Q(k)$ is the noise covariance (11) and also relates to the model accuracy. In order to achieve a more realistic model of the odometry error probability distribution it is necessary to have in account that for abrupt acceleration or deceleration the wheels can slip, consequently there is an significant position estimate error increase, mainly in the angle (Gonçalves et al., 2005).

As there is a measure, the follow also apply:

3. Kalman gain calculation

$$K(k) = P(k^-)H^*(k)^T(H^*(k)P(k^-)H^*(k)^T + R(k))^{-1} \qquad (16)$$

Where $R(k)$ is the covariance matrix of the measurements.

4. State covariation update

$$P(k) = (I - K(k)H^*(k))P(k^-) \qquad (17)$$

5. State update

$$X(k) = X(k^-) + K(k)(z(k) - h(X(k^-, 0))) \qquad (18)$$

Where z(k) is the measurement vector and $h(X(k^-, 0))$ is $X(k^-)$.

4.2 Kalman Filter Performance

With the objective of evaluating the Kalman filter performance another robot race was made, as shown in the flowchart presented in Figure 2. The robot trajectory is presented in Figures 9 and 10.

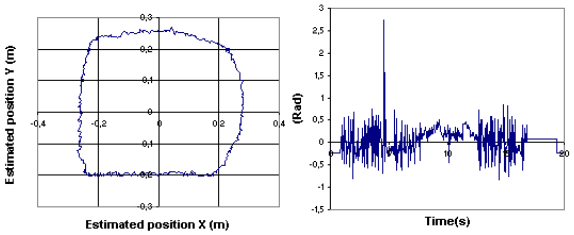


Figure 9: a) Robot trajectory, and (b) Estimated Angle.

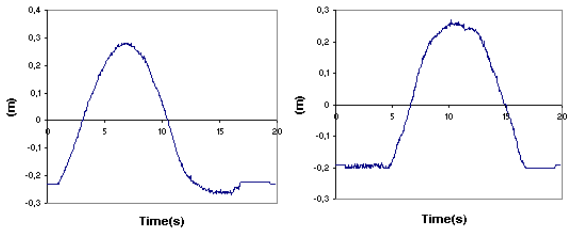


Figure 10: a)Estimated x, and (b) Estimated y.

The image quality of the robot markers for the presented robot race are presented in Figure 11 and the variance of the estimated robot position is presented in Figure 12. Whenever the image quality decreases the variance error of position estimate increases compromising the controller performance. On the other hand whenever the image quality increases the error variance is reduced and when the state update is done the position estimate error is reduced.

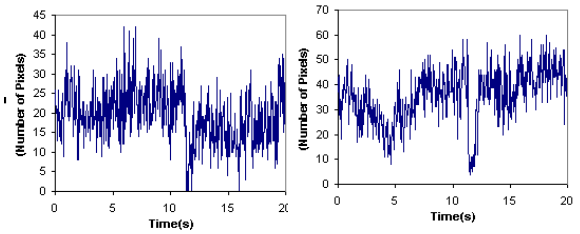


Figure 11: a) Image quality of the center marker Q1, b)Image quality of the angle marker Q2.

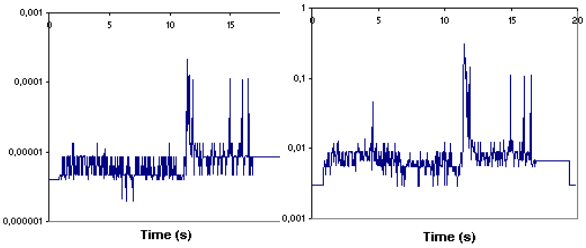


Figure 12: a) x and y variance, and (b) Angle variance.

5 CONCLUSIONS

Omnidirectional vehicles have many advantages in robotics soccer applications, allowing movements in every direction. The fact that the robot is able to move from one place to another with independent linear and angular velocities contributes to minimize the time to react, the number of maneuvers is reduced and consequently the game strategy can be simplified.

The robot relative position estimation is based on the odometry calculation. The odometry calculation uses each wheel velocity in order to estimate the robot position, the disadvantage is that the position estimate error is cumulative and increases over time.

It was made for the global vision localization system an analysis of the error probability distributions. The number of obtained pixels for the blue marker ($Q1$), affects the error variance in x and y. On the other hand the variance of the angle error probability distribution is affected by the number of pixels obtained for both makers, for the blue ($Q1$) and for the yellow ($Q2$).

Odometry and global vision real time data fusion was achieved applying an extended Kalman filter. This method was chosen because the robot motion equations are nonlinear and also because the measurements error probability distributions can be approximated to Gaussian distributions.

REFERENCES

(2008). Robocup. http://www.robocup.org/.

Borestein, Everett, and Feng (1996). *where am I, Sensores and Methods for Mobile Robot Positioning*. Prepared by the University of Michigan.

Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., and Thrun, S. (2005). *Principles of Robot Motion : Theory, Algorithms, and Implementations*. MIT Press.

Dudek, G. and Jenkin, M. (2000). *Computational Principles of Mobile Robotics*. Cambridge University Press.

Gonçalves, J., Costa, P., and Moreira, A. (2005). *Controlo e estimação do posicionamento absoluto de um robot omnidireccional de três rodas*. Revista Robótica, Nr 60, pp 18-24.

Gonçalves, J., Pinheiro, P., Lima, J., and Costa, P. (2007). Tutorial introdutório para as competições de futebol robótico. *IEEE RITA - Latin American Learning Technologies Journal*, 2(2):63–72.

Kalmár-Nagy, T., D'Andrea, R., and Ganguly, P. (2002). Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. In *Sibley School of Mechanical and Aerospace Engineering*.

Negenborn, R. (2003). *Robot Localization and Kalman Filters - On finding your position in a noisy world*. Master Thesis, Utrecht University.

Ribeiro, F., Moutinho, I., Silva, P., Fraga, C., and Pereira, N. (2004). Controlling omni-directional wheels of a robocup msl autonomous mobile robot. In *Proceedings of the Scientific Meeting of the Robotics Portuguese Open*.

Ribeiro, M. I. (2004). *Gaussian Probability Density Functions: Properties and Error Characterization*. Technical Report, IST.

Sousa, A. (2003). *Arquitecturas de Sistemas Robóticos e Localização em Tempo Real Através de Visão*. PHD Thesis, Faculty of Engineering of the University of Porto.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT Press.

Welch, G. and Bishop, G. (2001). *An introduction to the Kalman filter*. Technical Report, University of North Carolina at Chapel Hill.

A NEW APPROACH OF GRAY IMAGES BINARIZATION FOR ARTIFICIAL VISION SYSTEMS WITH THRESHOLD METHODS

Andrei Hossu and Daniela Hossu
University Politehnica of Bucharest, Faculty of Control and Computers
Dept. of Automatics and Industrial Informatics, 313 Spl. Independentei, sector 6, RO-77206, Bucharest, Romania
hossu@aii.pub.ro, dana@aii.pub.ro

Keywords: Vision systems, Gray level image binarization, gray level histogram, global optimum thresholding, dynamic optimum threshold, temporal histogram, temporal thresholding and moving scene in robotic automation.

Abstract: This paper presents some aspects of the (gray level) image binarization methods used in artificial vision systems. It is introduced a new approach of gray level image binarization for artificial vision systems dedicated to the specific class of applications for moving scene in industrial automation – temporal thresholding. In the first part of the paper are remarked some limitations of using the global optimum thresholding in gray level image binarization. In the second part of this paper are presented some aspects of the dynamic optimum thresholding method for gray level image binarization. In the third section are introduced the concepts of temporal histogram and temporal thresholding, starting from classic methods of global and dynamic optimal thresholding of the gray level images. In the final part are presented some practical aspects of the temporal thresholding method in artificial vision applications for the moving scene in robotic automation class; highlighting the influence of the acquisition frequency on the methods results.

1 IMAGE BINARIZATION WITH GLOBAL THRESHOLD

Threshold methods are defined as starting from the analyse of the values of a function T of the type:

$$T = T\,[x, y, p(x, y), f(x, y)] \tag{1}$$

Where:

$f(x, y)$ – represents the intensity value of the image element located on the co-ordinates (x, y);

$p(x,y)$ – represents the *local properties* of the specific point (like the average intensity of a region centred in the co-ordinates (x, y)).

T – is the *binarization threshold*

The goal is to obtain from an original gray level image, a binary image $g(x, y)$ defined by:

$$g(x, y) = \begin{cases} 1 & \text{for } f(x, y) > T \\ 0 & \text{for } f(x, y) \le T \end{cases} \tag{2}$$

For T a function only of $f(x, y)$, the obtained threshold is called *global threshold*.

In the case of T a function of both $f(x, y)$ and $p(x, y)$, the obtained threshold is named *local threshold*.

In the case of T a function of all $f(x, y)$, $p(x, y)$, x and y, the threshold is a *dynamic threshold*.

1.1 Intensity Level on Normal Distribution Assumption

Gray level histogram represents the probability density function of the intensity values of the image.

In order to simplify the explanations, we suppose the image histogram of the gray levels is composed from two values combined with additive Gaussian noise:

- The first segment of the image histogram corresponds to the background points – the intensity levels are closer to the lower limit of the range (the background is dark)

- The second segment of the image histogram corresponds to the object points – the intensity levels are closer to the upper limit of the intensity range (the objects are bright).

The problem is to estimate a value of the threshold T for which the image elements with an intensity value lower than T will contain background points and the pixels with the intensity value greater than T will contain object points, with a minimum error. For a real image, the partitioning between the

two brightness levels is not so simple and also not so accurate. The partitioning is fully accurate only if the two modes of the bimodal histogram are not overlapped. The classification is defined as the process of the distribution of the pixels in classes. The goal of the binarization process is the minimisation of the error of classification. The optimum binarization threshold is located in the intersection position of the two normal distributions.

The estimation of the error of classification is obtained from the area of the overlapped segments:

$$E = \frac{A + B}{image\ size} \tag{3}$$

Suppose the image contains two intensity level values affected with additive Gaussian noise. The mixture probability density function is:

$$p(x) = P_1 p_1(x) + P_2 p_2(x) \tag{4}$$

Where:

x – the random value representing the intensity level,

$p_1(x)$, $p_2(x)$ – are the probability density functions,

P_1, Pp_2- are the a priori probabilities of the two intensity levels ($P_1 + P_2 = 1$).

For the normal distribution case on the two brightness levels:

$$p(x) = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp \frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{P_2}{\sqrt{2\pi}\sigma_2} \exp \frac{(x-\mu_2)^2}{2\sigma_2^2} \tag{5}$$

Where:

$$p(x) = P_1 p_1(x) + P_2 p_2(x) \tag{6}$$

μ_1, μ_2 - are the mean values of the two brightness levels (the two modes),

σ_1, σ_2 - are the standard deviations of the two statistical populations.

Suppose the background is darker than the object. In this case $\mu_1 < \mu_2$ and defining a threshold T, so that all pixels with intensity level below T are considered belonging to the background and all pixels with level above T are considered object points. The probability of misclassification an object point (classifying an object point as a background point) is:

Similarly, E_2:

$$\begin{aligned} E_1(T) &= \int_{-\infty}^{T} p_2(x)dx \\ E_2(T) &= \int_{T}^{+\infty} p_1(x)dx \end{aligned} \tag{7}$$

The probability of error is given by:

$$E(T) = P_1 E_2(T) + P_2 E_1(T) \tag{8}$$

To find the threshold value for which the error is minimum, $E(T)$ is differentiate with respect to T:

$$P_1 p_1(t) = P_2 p_2(t) \tag{9}$$

Applying the result to the Gaussian density we obtain:

$$AT^2 + BT + C = 0 \tag{10}$$

Where:

$$\begin{aligned} A &= \sigma_1^2 - \sigma_2^2 \\ B &= 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2) \\ C &= \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + \sigma_1^2\sigma_2^2 \ln \frac{\sigma_1 P_1}{\sigma_2 P_2} \end{aligned} \tag{11}$$

If the standard deviations are equal, a single threshold is sufficient:

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln \frac{p_2}{p_1} \tag{12}$$

If the probabilities are equal $p_1 = p_2$ the threshold value is equal with the average of the means.

A way of checking the validity of the assumption of bimodal histogram is to estimate the mean-square error between the mixture density, $p(x)$ and the experimental histogram $h(x_i)$.

$$M = \frac{1}{N} \sum_{i=1}^{N} [p(x_i) - h(x_i)]^2 \tag{13}$$

Where: N – number of possible levels of the image (usually $N = 256$)

The image binarization is obtained changing the colour attribute of each pixel according to its intensity level relative to the binarization threshold. Characteristics of the global thresholding methods (Borangiu, et al., 1994)., (Haralick and Shapiro, 1992):

- The assumption that both classes have the same standard deviation is acceptable, but the assumption the classes (two levels) have the same a priori probabilities in many applications is not acceptable.

In the case of the artificial vision systems dedicated to object recognition for industrial applications there is a large amount of a priori information about the image that has to be processed. Better results of estimation of the distribution of the image elements of the scene (background image, without the objects) can be obtained. Usually, in robotic applications, the illumination environment is known and controlled

and also the object classes with a probability of apparition in the image are not known. In many robotic application an estimation of the ratio between the area of the objects to be analysed and the total area of the image scene, can be made with good results (a batter estimation than the assumption of $P_1 = P_2 = 0.5$).

2 IMAGE BINARIZATION WITH DYNAMIC THRESHOLD

There are some classes of scenes of artificial vision systems where using the global threshold methods is not acceptable:

- The case of the applications where the lighting system does not supply a uniform intensity all over the analysed surface.
- Segments of the image (or some times, image elements) do not have the same behaviour in the same lighting conditions.

For these types of images, for binarization of the image, the most often used are dynamic threshold methods. The methods are based on the local analyse of the image. The algorithm of the estimation of the dynamic threshold consist of:

- The original image is divided in regions of a prescribed size.
- For each region it is estimated the histogram
- For each histogram it is estimated the error induced from the assumption of bimodal histogram (a histogram built from two normal distributions)
- If the value of the error is less than an acceptable value, the global threshold for the region is estimated.
- If the value of the error is too big (the histogram is too far from a bimodal histogram) the threshold value for binarization is estimated from the interpolation of the neighbours region threshold values (for which the assumption of a bimodal histogram is considered acceptable).
- In the final stage, a second interpolation process is applied: for each image element is assigned a threshold value $T(x, y)$ from the interpolation of the values of the neighbour image elements.

The method is called dynamic thresholding because the value of the resulted threshold for each image element is dependent of the position of the element in the image - $T(x, y)$.

Characteristics of the dynamic thresholding methods:

- Lack of processing time consumption – each element of the image is used at least two times (the method requires multiple-pass of the image) in different steps of the algorithm (and the number of the elements is very large).
- Estimation of the acceptable error value (or the validation of the bimodal histogram assumption) is a complex process.
- To choose the size of the image regions we have to take into account:
- Large size of the region makes the method to loose the dynamic threshold characteristics and to fail into a global threshold method
- Small size of the region makes to loose the statistical characteristic of the population of the image elements contained by the analysed region (and the accuracy of the results is lost).

The last comment on the method is the fact that this method does not solve the problem of the non-uniformity of the illumination system or of the acquisition sensor.

3 TEMPORAL HISTOGRAM

For the class of artificial vision systems dedicated for moving scene (used very often in inspection and robotic applications) three types of image intensity level distortions can be identified (Croicu, et al., 1998), (Hossu, et al., 1998):

- Illumination non-uniformity (obtaining a uniform intensity of the light on the whole area of the scene where the image is analysed – usually 2 m – it is practical impossible).
- Sensor non-linearity – for linear cameras with a large number of pixels per row (2048 and more) can be identified areas of non-linear behaviour of the sensor (there are segments of the linear sensor with a different behaviour of the elements sensitivity at light intensity).
- Sensor cells non-uniformity – in cameras with CCD sensor, the cells presents a different response on sensitivity at light intensity related to their neighbours

In Figure 1 are presented the image intensity level distortions.

The main problem of the methods presented before represents the assumption that the image is a statistical population obtained from the addition of two ore more distributions (in the general accepted case normal distribution).

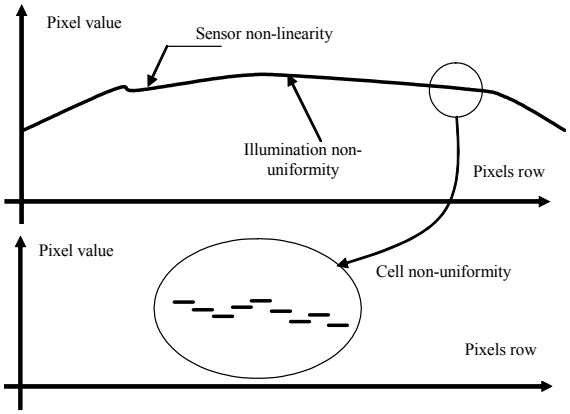


Figure 1: Image intensity level distortions for CCD linear camera acquisition.

In the general case (an *array image*) an image represents a data set of:

$$\{f(x, y) \mid x \in [0,N], y \in [0,M]\} \quad (14)$$

where:

N represents the number of image elements per row (number of image columns),

M represents the number of image elements per column (number of image rows),

In the *linear image* case, this data set become:

$$\{f(x) \mid x \in [0,N]\} \quad (15)$$

where:

N represents the number of image elements per row (number of image columns).

This assumption on the distribution of the intensity levels has the starting point the assumption that the insertion point of the noise is located on the transmission level of the information. In other words, the assumptions is that:

- The acquired image is an ideal image (with only two gray levels: the gray level of the scene pixels and the gray level of the pixels corresponding to the object)
- Then a global noise is applied, transforming the two levels in two normal distributions.

The assumption is false and using it we are analysing a histogram, which is far away of two normal distributions, and from here the results are distorted. In reality the noise on the intensity level has its insertion point on the acquisition level and not on image transmission level. Intensity source has the meaning of intensity signal on the acquisition element and not only the lighting system. This implies the fact that the noise on the intensity source represents the whole chain of: lighting source noise, reflective characteristics of the object surface and reflective characteristics of the scene surface and the sensitivity characteristics of the sensor. Moving the insertion point of the noise we obtain: In the general case (an *array image*) an image represents a data set of:

$$\{f_i(x, y) \mid i \in [0,L]\} \quad (16)$$

where:

L represents the number of the image frames (the size of the statistic population analysed),

$x \in [0,N]$, N representing the number of image elements per row (number of image columns),

$y \in [0,M]$, M representing the number of image elements per column (number of image rows).

In the *linear image* case, this data set become:

$$\{f_i(x) \mid i \in [0,L]\} \quad (17)$$

where:

L represents the number of the image frames (the size of the statistic population analysed),

$x \in [0,N]$, N representing the number of image elements per row (number of image columns).

In this way several temporal built statistical populations (from intensity levels of the same image element on a set of image frames acquired on different moments) replace the spatial built statistical population (made from image elements of the same image). The method of temporal histogram has the result the fact that each element of this set of histograms represents a bimodal histogram with two not overlapped modes (in case of a correct acquisition environment). It can be also introduce an estimation of the quality of the acquisition and binarization process using the estimation of the misclassification error analysing the parameters of the two normal distributions. The method offers also the capacity of identification of the areas where some modifications should be done (on the lighting system) in order to improve the quality of the acquisition and binarization process. The lack of the proposed method is the memory consumption (it has to be built N x M different histograms in array acquisition, or N – in linear acquisition case). This problem is not so restrictive because at the end only the threshold values have to be stored and not the whole histograms. Another restriction is the fact that the method requires a large number of image frames acquired for construction of the statistical populations (in application set-up time). In the case of the systems dedicated to industrial applications usually this does not represent a real problem. This type of applications does not require a system response in condition of a small number of image

	V [m/min]	Object Min+	Object Max	Object Min-	Scene Min+	Scene Max	Scene Min-	Threshold
1	20,7	221,532847	203,810219	168,364964	66,459854	44,306569	22,153285	110,766423
2	25,9	174,338624	160,391534	132,497354	52,301587	34,867725	17,433862	87,169312
3	31,1	147,510373	135,709544	112,107884	44,253112	29,502075	14,751037	73,755187
4	36,2	130,479452	120,041096	99,164384	39,143836	26,095890	13,047945	65,239726
5	41,3	118,513120	109,032070	90,069971	35,553936	23,702624	11,851312	59,256560
6	46,4	109,644670	100,873096	83,329949	32,893401	21,928934	10,964467	54,822335
7	51,4	102,927928	94,693694	78,225225	30,878378	20,585586	10,292793	51,463964
8	56,5	97,474747	89,676768	74,080808	29,242424	19,494949	9,747475	48,737374
9	61,6	93,040293	85,597070	70,710623	27,912088	18,608059	9,304029	46,520147
10	66,7	89,363484	82,214405	67,916248	26,809045	17,872697	8,936348	44,681742
11	72,5	85,877863	79,007634	65,267176	25,763359	17,175573	8,587786	42,938931

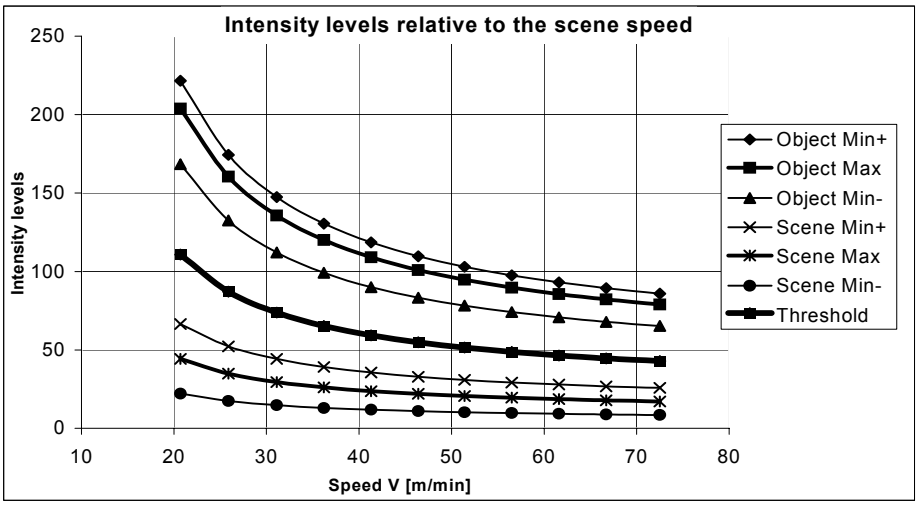


Figure 2: The influence on the intensity levels of the speed of the scene (acquisition frequency).

frames a priori acquired. The vision systems dedicated to industrial applications can take the advantage on the fact that the image environment does not change a lot in time. In this way it can be initially reserved a certain time for acquiring a large enough number of image frames in order to be able to identify the permanent characteristics of the environment. All the intensity level distortions present permanent characteristics. Using this method is a necessity for the artificial vision systems dedicated to applications where the errors on binarization are not acceptable. In the applications dedicated exclusively to shape recognition the errors are accepted in a predefined range.

4 BINARIZATION THRESHOLD VALUE AFFECTED BY THE ACQUISITION FREQUENCY ON THE

In moving scene applications, in order to maintain a constant resolution of the vision system along the direction of the scene movement, it is necessary the ratio between the acquisition frequency (the image lines rate – in the case of a line scan camera) and the scene speed to be constant. The acquisition frequency determines the exposure time of the CCD sensor cells. It can be notice an important influence of the speed (of the conveyor) on the intensity level of the same image element in the same lighting environment. In Figure 2 are presented the experimental results obtained analysing the influence on the intensity levels (for both: bright object and dark background) of the speed of the conveyor (acquisition frequency). The results were obtained on a statistical population from an image element on each measured speed. The second column represents the measured speed of the scene (conveyor) – V [m/min]. The 3rd to 8th columns represent image intensity levels estimated from the analysed statistical population (temporal histogram). The values from the Threshold column are the binarization threshold values obtained from a global optimum temporal thresholding method applied on the histogram built for each analysed level of the speed. In Figure 3 are presented graphical the explanations on the meanings of the data involved in the analysis of the influence of the speed

(acquisition frequency) on the intensity levels. The artificial vision system benefits from these results using a relation between the value of the binarization threshold and the speed V of the scene

$$T = T(x, V) \tag{18}$$

Because of the response time restrictions imposed to the artificial vision system, instead of using an explicit expression of the estimated function $T(x, V)$, a search method in an a priori filled table (at set-up time) is more appropriate. The size of the table is 256 (the number of the possible values of the binarization thresholds), containing floating-point values of the speed of the conveyor (acquisition frequency) for which the value of the binarization threshold has to be changed.

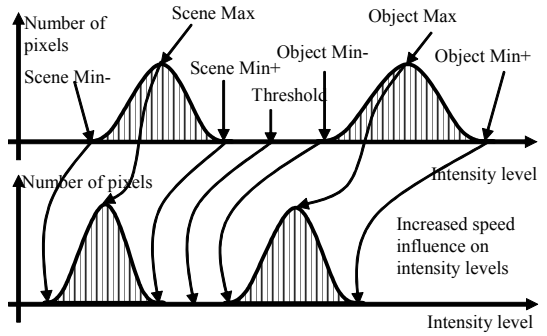


Figure 3: The influence of the speed on the intensity levels.

5 CONCLUSIONS

For the class of artificial vision systems dedicated for inspection and measurement industrial applications the error on binarization process is not acceptable. In this, case classic methods like global, local and dynamic threshold are not applicable. The paper introduces a new approach of gray level image binarization – temporal thresholding. For the class of artificial vision systems dedicated for moving scene the acquisition frequency is dependent on the speed of the transmission support (usually a conveyor). To solve this problem, the artificial vision system has to estimate the influence of the acquisition frequency on the histogram and on the binarization threshold values. The paper proposes a processing time efficient method to estimate the binarization threshold for the case of an *error free* vision system in the case of variation of the acquisition frequency.

REFERENCES

Haralick, R., Shapiro, L. (1992) *Computer and Robot Vision*, Addison-Wesley Publishing Company.

Borangiu, Th, Hossu A., Croicu, A. (1994) - *ROBOTVISIONPro, Users Manual,* ESHED ROBOTEC, Tel – Aviv.

Croicu, A., Hossu, A., Dothan, E., Ellenbogen, D., Livne, Y. (1998)- ISCAN-Virtual Class based Architecture for Float Glass Lines, *IsoCE'98*, Sinaia.

Hossu, A., Croicu, A., Dothan, E., Ellenbogen, D., Livne, Y.(1998) - *ISCAN Cold-Side Glass Inspection System for Continuous Float Lines*, User Manual, Rosh-Haayn.

ACTIVE SECURITY SYSTEM FOR AN INDUSTRIAL ROBOT BASED ON ARTIFICIAL VISION AND FUZZY LOGIC PRINCIPLES

B. Fevery, B. Wyns, L. Boullart
Department of Electrical Energy, Systems and Automation, Ghent University, Ghent, Belgium
brecht.fevery@ugent.be, bart.wyns@ugent.be, luc.boullart@ugent.be

J. R. Llata García, C. Torre Ferrero
Control Engineering Group, Electronic Technology and Automatic Systems Department
University of Cantabria, Santander, Spain
llata@teisa.unican.es, carlos@teisa.unican.es

Keywords: Active security, application, artificial vision, fuzzy logic, real-time, robot control.

Abstract: An active security system assures that interacting robots don't collide or that a robot operating independently doesn't hit any obstacle that is encountered in the robots workspace. In this paper, an active security system for a FANUC industrial robot is introduced. The active security problem where one robot needs to avoid a moving obstacle in its workspace is considered. An obstacle detection and localization mechanism based on stereoscopic vision methods was successfully developed. To connect the vision system, an operator's pc and the robot environment a real-time communication is set up over Ethernet using socket messaging. We used fuzzy logic for intelligent trajectory planning. A multitask oriented robot application in the KAREL programming language of FANUC Robotics was implemented and tested.

1 INTRODUCTION

In industrial settings, robots often work on valuable products and with expensive tools. When more robots are working together on one assignment, a collision free interaction of the robot arms needs to be guaranteed at all time. Systems that establish collision free robot interaction are identified as Active Security Systems (ASSYS). ASSYSs can also be situated in the domain of interaction between industrial manipulators and human operators, where the physical safety of the operator, rather than an economical concern, constitutes the necessity for the design of an appropriate security system.

The key principle of ASSYSs is the vigilance of the work area of cooperating robots and the streaming of information about events that are unexpected for each robot. This contrasts with a strategy where every robot is seperately programmed to execute its task and where interaction signals are sent between robots over rigid communication media.

Safe robot motion is typically guaranteed by the use of a sensor system. A camera network based human-robot coexistence system was already proposed in (A.J. Baerveldt, 1992) and a safety system also using a network of cameras and with path replanning in an on-line manner was presented in (D. Ebert et al., 2005). In this paper, we present the setup of a basic vision system to detect and positionally reconstruct obstacles in the robot's workspace. The stereoscopic vision techniques applied in the design of the vision system will be presented in section 2.

In literature, some researchers focus on the direct kinematics of robot manipulators. Using the differences between actual and goal angular configuration of every axis, output actions for every axis's motor are produced taking into account an obstacle's configuration in the two or three dimensional space. In this paper, we will present a security system that controls the motional actions of an industrial FANUC Robot Arc Mate 100*i*B with six degrees of freedom and a circular range of 1800 millimeters. Instead of giving commands to every axis's motor, positional and rotational configuration of the robot arm will be calculated along the nodes of an alternative path around one detected obstacle. Intelligent path planning is done by using a fuzzy logic control system. A rule base composed of linguistic if-then implications is used to simulate human reasoning in decision taking. The fuzzy system produces a set of alternative positions and rotational configurations that assure collision free motion continuation towards a final location. Fuzzy logic

is popular due to its simplicity and hands-on, intuitive design of the control strategy and was successfully applied by preceding authors, e. g. in (Bischoff, 1999), to build active safety systems for robots. In this paper, a three dimensional obstacle avoidance strategy will be introduced that is founded on the idea of repelling and attracting forces (P. Zavlangas et al., 2000). The design of a fuzzy logic controller will be highlighted in section 3.

Although specialised solutions exist for each component of the proposed ASSYS, the goal in this paper was to build such a system using only basic components communicating over an Ethernet network. No multiple robot interaction was assumed and due to the early stage of the investigation, the vision system was only designed to make the robot avoid collision with a single, however dynamically moving, obstacle. Attention was also given to the time performance of the vision system.

To make the industrial FANUC robot move along an alternative path in an on-line manner, a robot application needed to be programmed in the proper programming language KAREL of FANUC Robotics. A multitask oriented design in the KAREL language assures that alternative positions can be read in by the robot's system and subsequently moved to by the robot arm. The architecture of the robot application, as well as details on the real-time communication system established over Ethernet, will be commented in section 4. In section 5 results and drawbacks of the designed ASSYS are commented.

2 ARTIFICIAL VISION

2.1 3D Object Reconstruction

Stereoscopic vision applications intent to reconstruct the 3D location of characteristic object points. From (Torre Ferrero, 2002) an analytical method was taken that allows for a unique 3D reconstruction of an object point P, knowing the pixel sets (u_1, v_1) and (u_2, v_2) of P's projection into two different image planes I_1 and I_2. The camera's projection matrices, that are composed of the camera's extrinsic and intrinsic parameters (Gonzaléz Jiménez, 1999), are also needed for reconstruction. These parameters were obtained for every camera by applying a camera calibration method based on (J. Heikkilä et al., 1997). For more details on camera projection principles and reconstruction methods, please consult (Gonzaléz Jiménez, 1999) and (Torre Ferrero, 2002).

2.2 Camera Setup of the Vision System

A triplet of network cameras was installed to watch the robot's workspace. Camera images can be obtained by sending an image request signal to their IP address over a Local Area Network (LAN). For every camera, a video stream of images using ActiveX components is activated. Images are taken out of the video stream and saved as image matrices of dimension 480x640x3 in the Red Green Blue (RGB) image space. A pc is used to perform image processing operations. The cameras were collocated in a triangular pattern and mounted on the ceiling above the robot's workspace.

2.3 Object Detection and Reconstruction Methods

In industrial settings, image processing times need to be small. Preliminary knowledge about the object's color and shape is therefore often used to detect obstacles in the robot's workspace as quickly as possible. For the experimental setup of our vision system, we worked with a foam obstacle of parallelepiped structure. The motion of the obstacle is achieved by simply dragging the foam into the robot's workspace with a rope. Because it is not within the scope of this paper, no attention was given to the detection of the robot's arm, nor to the detection of humans or objects of other form than a parallelepiped. In the next sections, we will introduce the vision techniques that were used for the detection of a moving obstacle and for the reconstruction of its 3D position. The reconstruction method is based on the technique of epipolar lines, which form a useful geometric restriction in vision applications.

2.3.1 Obstacle Observation

The obstacle of parallelepiped form is detected in an image by converting this image to binary form and subsequently check for the presence of contours of squared form, using a simple criterion that relates a square's perimeter to its area. The presence of the obstacle is checked by drawing the image of one camera out of the activated video stream every 50 milliseconds and by applying the square detection criterion.

When a moving obstacle is detected for the first time in the workspace, the ASSYS halts all robot motion. Only if the obstacle stops moving within a certain number of time frames after it had first been detected, the robot will resume its motion, now moving around the obstacle. By taking subsequent images out of the video stream of the same camera and resting

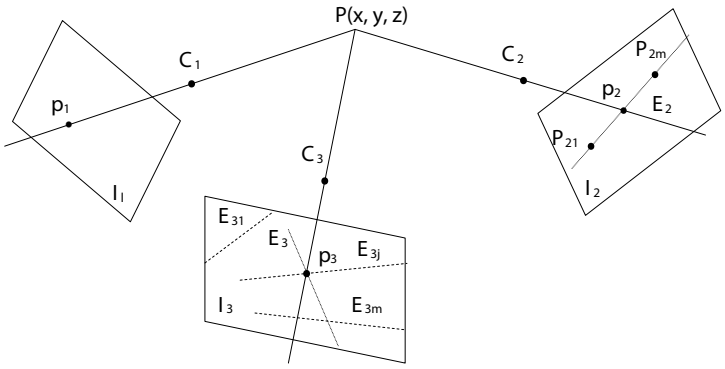


Figure 1: Pixel correspondence algorithm.

two subsequent image matrices, the system checks if the object has stopped moving. As soon as this condition holds, images are drawn out of the video stream of the two other cameras and the 3D reconstruction of the obstacle is initiated. We typically reconstruct the 3D location of the obstacle's four upper corners. These characteristic object points are determined by applying a *Canny* edge detector (Gonzaléz Jiménez, 1999) and a corner detection operator on the three images.

Once characteristic points -true and also false object corners due to image noise or nearby objects- are detected, an algorithm is applied to determine corresponding corners. This problem boils down to finding for every upper corner of the obstacle in a first image, the location of the same corner in the second and third image.

2.3.2 Calculation of Pixel Correspondences

The applied algorithm is based on the geometric restriction of the epipolar line: an image point P that is projected onto a pixel in a first image can only be projected onto one line of pixels in a second image. We aim to find three pairs of pixel coordinates of one point in space that is projected into three images I_1, I_2 and I_3. This problem is identified as the search for pixel correspondences. The algorithm can briefly be explained as follows, according to the notations of figure 1. The image point p_2 in the second image I_2 that corresponds to the point p_1 in the first image I_1 has to be situated on the epipolar line E_2 associated to p_2. Characteristic pixels of I_2 that are located sufficiently close to this epipolar line are selected as correspondence candidates P_{2i} in I_2. When the epipolar lines E_{3j} associated to all correspondence candidates P_{2i} are constructed in a third image I_3, this results in a number of intersections p_{3k} in I_3 between the epipolar lines associated to the set P_{2i} and the epipolar line E_3 associated to p_1. The intersection that coincides or is located sufficiently close to one of the characteristic points in I_3, results in the unique corresponding pixel triplet $\{p_1, p_2, p_3\}$. Epipolar line pixel equations from (Torre Ferrero, 2002) were used to construct epipolar lines.

As soon as the corresponding corner pixels are found in the three camera images, the pixel correspondences are used to calculate the obstacle's 3D location in space, as described in section 2.1. False pixel correspondences, that originated from detected corners not belonging to the object, can be discarded because the resulting 3D positions don't fall within the range in which the obstacle is expected to be encountered. The obstacle's 3D location in space is used as an input of the fuzzy logic controller that calculates an alternative trajectory.

3 FUZZY LOGIC CONTROL

3.1 Introduction

A Fuzzy Logic Controller (FLC) is a useful tool to transform linguistic control strategies based on expertise into an automatic control strategy (O. Cordon et al., 2001). The basic idea is to assign linguistic labels to physical properties. The process that converts a numerical value into a linguistic description is the fuzzification process. Using a rule base that simulates human reasoning in decision taking, a number of linguistic control actions is computed and subsequently defuzzified or converted to numerical control actions. For more information and a detailed description on FLCs, please consult (O. Cordon et al., 2001).

A pneumatically controlled tool was mounted on

the robot arm. The term End Effector is used to indicate this tool, that is depicted in figure 2 for different configurations. In robot terminology, the central point of the End Effector is called the Tool Center Point (TCP). The dots between the ends of the End Effector in figure 2 represent this TCP.

3.2 Fuzzy Avoidance Strategy

A fuzzy rule containing two types of actuating forces was designed. An attracting force proportional to the 1D distance differences between actual TCP coordinates and final location coordinates causes the FLC to output distance increments towards the final location. A repelling force describing the distance to the obstacle's sides deactivates the attracting force and invokes specific avoidance actions that have to be undertaken by the robot's End Effector to avoid collision with the obstacle.

Further on, it will be explained how safety zones are constructed around the obstacle, based on the distance differences between the TCP and the obstacle's sides. When the robot's TCP enters one of these safety zones around the obstacle, two types of avoidance actions are undertaken. Rotational actions guarantee the End Effector's orthogonal position to the obstacle's sides and translational actions assure an accurate avoidance in position. This idea is depicted in figure 2.

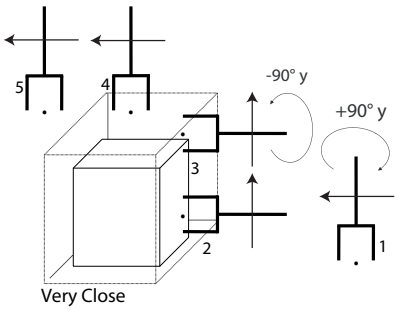


Figure 2: Fuzzy avoidance strategy.

3.3 Inputs to the Fuzzy Logic Controller

The inputs of the FLC consist of two types. A first type describes 1D distance differences between actual TCP coordinates and final location coordinates, while the second input indicates if the TCP is near to one of the obstacle's sides. The first input is related to the attracting influence and the second one to the repelling influence. It will now be explained how both types of inputs to the FLC are composed.

The distance between final location and TCP can be described in linguistic terms as e.g. *Close* or *Far*.

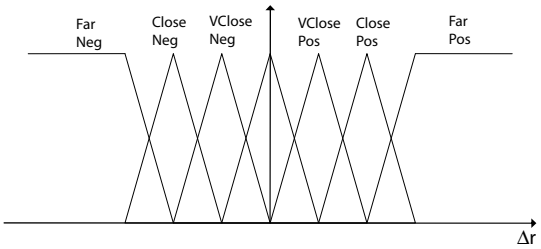


Figure 3: MFs for fuzzy sets of attracting influence.

For a given numeric value of distance to the final location, each of the linguistic labels will be true with a certain value in the range [0, 1]. This value will be determined by the Membership Function (MF) of the specified linguistic distance label. Figure 3 illustrates the MFs of the labels that describe the distance difference Δr in x, y and z direction between the coordinates of the Tool Center Point and the final location coordinates. MFs of triangular and open trapezoidal form were chosen because they are easy to implement in programming applications and require small evaluation times. The central triangular represents the MF for *Contact* with the obstacle. Table 1 indicates the 1D distance descriptions in coordinates $r = x$, y and z towards the final desired configuration.

Table 1: Labels for attracting influence.

Linguistic label	Short notation
Goal Far Negative	GFar Neg *r*
Goal Close Negative	GCl Neg *r*
Goal Very Close Negative	GVCl Neg *r*
Goal Reached	Goal *r*
Goal Very Close Positive	GVCl Pos *r*
Goal Close Positive	GCl Pos *r*
Goal Far Positive	GFar Pos *r*

The second FLC input is related to the repelling force. To understand how these FLC inputs originate, we make the following consideration. If the robot's TCP is *Very Close* to the positive x side of the obstacle, this means it is close to the positive x bound of the obstacle *AND*: within the y and z range *OR* within the y range and very close to the positive z bound *OR* ... . Figure 4 illustrates this idea for the construction of the label *Very Close Positive x*.

The distance differences Δs ($s = x$, y and z-dimension) represent the distances of the TCP to the closest obstacle bound in the considered coordinate. For the example of figure 4, the considered label needs to be evaluated using *AND* and *OR* logical operators. Table 2 represents distance difference descriptions towards the sides of the obstacle.

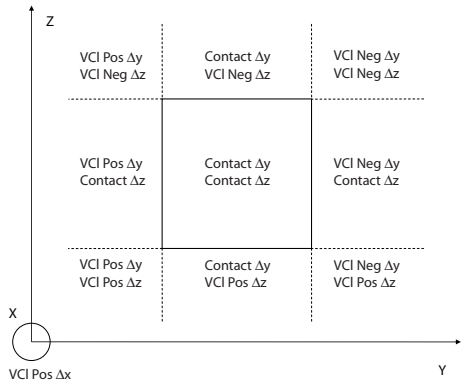


Figure 4: Construction of label *Very Close Positive x.*

Table 2: Labels for repelling influence.

Linguistic label	Short notation
Far Negative	Far Neg Δs
Not Close Negative	NCl Neg Δs
Close Negative	Cl Neg Δs
Very Close Negative	VCl Neg Δs
Contact	Contact Δs
Very Close Positive	VCl Pos Δs
Close Positive	Cl Pos Δs
Not Close Positive	NCl Pos Δs
Far Positive	Far Pos Δs

Note that, according to the idea of figure 4, the labels of table 2 have three-variable MFs, because they are all composed of one dimensional MFs for coordinate differences Δr ($r = x$, y and z) towards the obstacle's boundary coordinates. These 1D MFs are similar to the ones for the attracting forces depicted in figure 3. Next step is to construct safety zones around the obstacle, as shown in figure 2 for the label *Very Close*. Analogously, zones *Close* and *Not Close*, and an outer region *Far*, complementary to the inner zones, are constructed.

To fuzzify the entrances of the Fuzzy Inference System, we used a singleton fuzzificator (J. R. Llata García et al., 2003).

3.4 Design of a Rule Base

The basic principle of the rule base is the deactivation of the attracting influence -determined by the distance to the final location- when the repelling influence is triggered. Taking this into account, logical rules for closing up to the final location can be constructed.

For the rules related to the repelling influence, we can state that the designer of the rule base is free to choose the direction and sense of the avoidance actions. We decided to undertake an avoidance action in positive z direction when the TCP closes up to the (*Very*) *Close* x or y, *Negative* or *Positive* side of the obstacle.

The avoidance action for the (*Very*) *Close* z, *Positive* or *Negative* side, is decided upon by a criterion that checks the distance difference in x and y coordinate of the TCP's current position and the final location coordinates. If the distance difference is bigger in the x direction, then an avoidance action in x is undertaken, otherwise in y direction.

As soon as the TCP enters the safety zone *Not Close*, a rotation of -90° or +90° around the appropriate axis of a fixed coordinate system needs to be undertaken, to prevent the End Effector from hitting the obstacle (see figure 2).

To resolve the fuzzy intersection operator used for the composition of rule premises and for the implication on rule consequents, we used a *T-norm* of the product type. In the aggregation of rule consequents an *S-norm* for the fuzzy union operator was chosen. We implemented a *maximum* operator as this *S-norm* to save in processing time.

Given an initial and final position and an obstacle's location supplied by the vision system, the FLC outputs a set of positional and rotational configurations that guarantee collision free motion towards the final location.

3.5 Outputs of the FLC

Fuzzy outputs of the *Sugeno* singleton type (J. R. Llata García et al., 2003) were used for defuzzification. Depending on the output of a rule, a specific value can be assigned to the considered system output. The designer of the FLC is free to determine the size of the output actions.

Upon detection of an obstacle and halting of robot motion, the TCP's current position is sent by the robot's operational system over a socket connection to the artificial intelligence system as a start point for the calculation of an alternative path. Alternative positions and rotational configurations are then sent back over the socket in data packages that contain the desired coordinates of the TCP and the desired rotational configuration of the End Effector with respect to the fixed coordinate system.

4 EXPERIMENTAL SETUP

4.1 Real-time Communication

Robotic control applications often have cycle times of typically hundreds of microseconds. When operational data needs to be exchanged between a robot and an operator's pc, the fastness and the guarantee of data transmission is of utmost importance. For many years, Ethernet was banned as a communication medium from the industrial work floor, for data packages that are sent over the Ethernet by devices connected to a same Local Area Network can collide and be lost, due to the network's media access control protocol CSMA/CD (Van Moergestel, 2007). Nowadays, Fast Ethernet switches can be used to isolate network devices into their own collision domain, hereby totally eliminating the chance for collision and loss of data packages. Ethernet switches together with the development of Fast Ethernet (100Mbps) and Gigabit Ethernet (1Gbps) have made Ethernet popular as a real-time communication medium in industrial settings (Decotignie, 2005).

To establish the Ethernet communications in the ASSYS, we used so called Ethernet *sockets*. Sockets are software entities that are assigned to a combination of communication port and IP address, so that they can be used by a client and a server device to communicate over a LAN. In our setup, this LAN was created by a Fast Ethernet switch. The exchange of all data packages between the industrial FANUC Robot Arc Mate 100*i*B and a pc running the vision and fuzzy logic applications is performed by socket messaging.

4.2 Multitask Robot Application

A multitask oriented active security application was developed and tested in KAREL, the programming language of FANUC Robotics for advanced user applications. A motion task executes a normal operation trajectory until a condition handler is triggered by the detection signal that was received through an Ethernet socket by a concurrently running communication task. When this condition handler is triggered, robot motion is halted within a time that is acceptably small and the TCP's current position is sent by the communication task to the operator's pc, where the FLC calculates the first alternative positions and sends them back over the opened socket connection to the communication task. An interrupt routine for motion along the alternative path is then invoked in the motion task. The robot axes's motors start accelerating immediately. Meanwhile, the communication task completes the reading in of subsequent alternative positions and rotational configurations. Motion continues until the original final location is reached. The KAREL application is written in a non-cyclic way. Upon reaching the final position, program execution is aborted. Coordination between the motion task and the communication task was realised by the use of semaphores. The artificial vision system, the FLC and the robot control application in KAREL were tested in an integrated way.

5 RESULTS AND DISCUSSION

The obstacle is dragged into the robot's workspace when the robot arm is close to the leftmost or rightmost point of its regular trajectory. Absence of the robot's arm in the central zone of the workspace is necessary for correct obstacle detection because the robot arm would deform the binary image of the obstacle's squared contour. Further development of the vision system is therefore needed to distinguish the robot arm from the obstacle in order to be able to signal obstacle presence in all operational situations. In an advanced stadium, detection criteria for human operators can also be elaborated.

However, if the robot arm is occulting one of the obstacle's upper corners in one of the three images, performing an accurate reconstruction of the obstacle's 3D location is still possible, since a free view on three of the four upper corners in all images is sufficient for the reconstruction. The parallelepiped in figure 5 depicts the result of the vision system and fuzzy logic path planning.

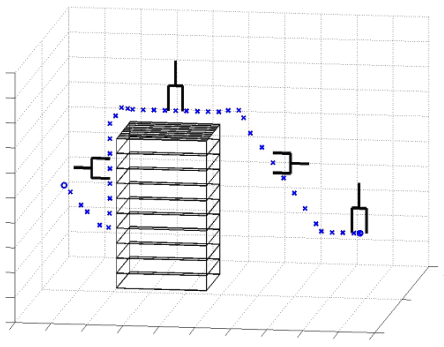


Figure 5: Alternative trajectory around reconstructed obstacle.

With distance increments of 50 millimeter in the FLC we typically obtained a number of 40 alternative positions. The designer of the FLC is free to choose the size of the translational increments larger or smaller and in a rather intuitive way. However, a

thorough study can be performed making a trade-off between small increments and thus larger calculation times and larger robot processing times or large distance increments and thus smaller calculation times and robot processing times. This last option implicates however that the safety zones around the obstacle need to be bigger and that longer trajectories have to be completed by the robot tool before it reaches the final location. For industrial settings, where small robot motion execution times are of utmost importance, this trade-off study is an interesting topic for future research. More specifically, a time efficient and distance optimal path construction algorithm can be designed.

The FLC only takes the TCP's position as an input. Collision of the robot's arm is prevented by rotating the End Effector +90° or -90° when it enters the first safety zone *Not Close*. For the majority of practically executable robot trajectories, this preventive action has proven to be sufficient. In future research however, the distance to the obstacle of extra points on the robot's arm will have to be monitored to guarantee safer motion.

In this design, a parameter to take into account is the processing time needed by the robot's system to handle new motion instructions. The robot system is able to continue program execution after launching a motion instruction. Moreover, a continuous transition between two separate motion instructions is possible using the appropriate clauses in the motion commands. Nevertheless, we chose to keep the number of motion commands as limited as possible and decided to only send every fourth alternative position as an effective motion instruction to the robot. Given the fact that alternative positions are situated close to each other (see figure 5), this strategy still results in accurate obstacle avoidance and in a smooth, continuous robot motion.

The time needed to draw images out of the video stream and save them as pixel matrices for further image processing could be restricted to 15 milliseconds. The computational time to identify pixel correspondences and make a 3D reconstruction is also very small. Regarding processing time, the bottleneck of the vision system, and thus of the entire ASSYS, has proven to be the identification of characteristic object pixels, in our case corner pixels of the parallelepiped. Improvements have to be made. Remark that during 3D reconstruction of the obstacle the robot is motionless, thus no unsafe situation is created due to the high processing times of the vision system.

6 CONCLUSIONS

An active security system for an industrial FANUC robot was designed. With special attention for real-time performance of the constituting subsystems, satisfying experimental results were obtained. The setup of the Ethernet communication through sockets, the fuzzy obstacle avoidance mechanism and the vision system open wide perspectives for future investigation on active security. More attention can be given to distinguishing the robot arm from foreign objects, to optimizing image processing times, to searching cost optimized alternative paths and to automating the robot application in a cyclic way.

REFERENCES

A. J. Baerveldt (1992). A safety system for close interaction between man and robot. *Proceedings of IFAC Conference on Safety Security Reliability SAFECOMP 1992*.

Bischoff, A. (1999). Echtzeit Kollisionsvermeidung für einen Industrieroboter durch 3d-Sensorüberwachung.

D. Ebert et al. (2005). Safe human-robot-coexistence: Emergency-stop using a high-speed vision-chip. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1821–1826.

Decotignie, J.-D. (2005). Ethernet-based real-time and industrial communications. *IEEE*, 93:1102–1117.

Gonzaléz Jiménez, J. (1999). *Visión por computador*. Paraninfo.

J. Heikkilä et al. (1997). A four-step camera calibration procedure with implicit image correction. *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

J. R. Llata García et al. (2003). *Introducción a las Técnicas de Inteligencia Artificial*. Grupo de Ingeniería de Control, Departamento de Tecnología Electrónica e Ingeniería de Sistemas y Automática, Universidad de Cantabria.

O. Cordon et al. (2001). Genetic fuzzy systems: Evolutionary tuning and learning of fuzzy knowledge bases. *World Scientific*.

P. Zavlangas et al. (2000). Industrial robot navigation and obstacle avoidance employing fuzzy logic. *Journal of Intelligent and Robotic Systems*, 27:85–97.

Torre Ferrero, C. (2002). Reconstrucción de piezas en 3d mediante técnicas basadas en visión estereoscópica.

Van Moergestel, L. J. M. (2007). *Computersystemen en Embedded Systemen, 2nd reviewed print*. Academic Service, Den Haag.

CONTRIBUTION CONCERNING ROBOT ACCURACY USING NUMERICAL MODELING

Daniela Ghelase[1], Luiza Daschievici[1] and Irina Ghelase[2]
[1] *"Dunarea de Jos" University, Str. Domneasca, Nr. 47, Galati, Romania*
daniela.ghelase@ugal.ro, luiza.tomulescu@ugal.ro
[2] *Politehnica University, Bucharest, Romania*
Irina_Ghelase@yahoo.com

Keywords: Robot accuracy, numerical modelling, computer simulation, rigidity of gearing.

Abstract: The kinematical accuracy of robot is very important. It is induced by the rigidity of each mechanism of the robot. The paper presents a numerical method to evaluate the rigidity of worm-gearing teeth. The software, including setting-up and graphic display, could be adopted of any kind of cylindrical worm-gear drive or for spur gear drives and bevel gear drives, mechanisms which are in the robot structure. Besides, we can determine geometrical parameters of the gear drives which influence the increase of accuracy of robot linkages.

1 INTRODUCTION

Into the kinematical chain there are worm-gear drives, screw-nut mechanisms and pinion-rack drives. During the working, these gear drives and mechanisms of the robot deform under the load, leading to the motion errors. The errors can not be entirely eliminated, but their maximum values must be limited. The theoretical advantage of the conjugate action in involute gears is lost due to the deflection of the teeth under load and due to the manufacturing and assembling errors. These factors produce instantaneous variations in the gear ratio.

As it is well-known, the rigidity of the meshing teeth changes as the contact point moves from the initial point of contact to the final point of contact. During the meshing the normal force is mobile on the tooth flank, it changes continuously the position with respect to the fixing zone of the teeth. The load is unevenly distributed, depending on the contact ratio. Consequently, all these factors causes rotative speed variations of the driven shaft, vibrations, shocks, noise, power loss, low durability of gears. The purpose of the present work is to develop a methodology to evaluate the rigidity of the worm-gearing tooth. By means of this methodology the performances of the robot mechanisms may be improved.

2 GEOMETRY OF THE WORM-GEARING TOOTH

In order to analyze the rigidity of the worm-gearing tooth we assume that the spatial gearing consists of more plane-gearings (pinion-rack drives), that in fact are cross sections perpendicular to worm-gear axis (Figure 1). The analytic solving of the problem, even for a ruled worm-gearing, is very difficult due to the complexity of the equations of the plane-gearing profiles that are involved in the enveloping. Consequently, we use the "minimum distance method" applied in the case of the "discrete representation" of the enveloping profiles. Thus, the enveloping profile of the elementary worm-gear (plane-gear) can be determined numerically by knowing "discretely" a matrix having as elements the coordinates of the worm axial section and by using the theorem of the "minimum distance method".

The minimum distance theorem in "discrete way" states (Ghelase, D., Daschievici, L., 2006):

The envelope to the family of curves, represented in "discrete way" as massive of the coordinates of the points belonging to the family curves, consists of the all points there are on these curves, for which, at a certain size of the increment φ_1, the distance at the meshing pole is minimum.

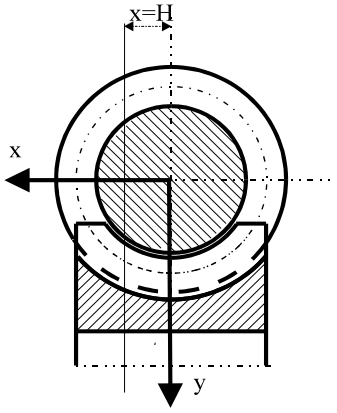


Figure1: Worm-gear drive.

2.1 Worm-Geometry

In order to determine the coordinates of the worm axial section, we focus on the case of a worm-gearing with modified profile could ensure, as well as possible, the generalization of the model from the geometrical viewpoint. Hence, let the axial section (x=0) of the worm (Figure 2) with constant pitch, having a circular arch profile with the centre in O_1 for the right flank and in O_2 for the left flank. The coordinates of the centre O_1, respectively O_2, are given by the following relations:

$$\begin{cases} Y_{O1}=R_e-u\cdot\cos\alpha-a\cdot\sin\alpha \\ Z_{O1}=b+u\cdot\sin\alpha-a\cdot\cos\alpha \end{cases}$$
$$\begin{cases} Y_{O2}=R_e-u\cdot\cos\alpha-a\cdot\sin\alpha \\ Z_{O2}=-b-u\cdot\sin\alpha-a\cdot\cos\alpha \end{cases} \quad (1)$$

where: *a* is a constant parameter;

$b=\pi\cdot m/4-1.25\cdot m\cdot tg\,\alpha$;

$p=m/2$;

$u=1.25\cdot m/\cos\alpha$;

$R=\sqrt{a^2+u^2}$ is the radius of the circular arc profile;

R_e is the tip radius of the worm tooth, all measured in mm (see Figure 2).

2.1.1 Equations of the Worm Flanks

In accordance with Figure 2, a point of the worm flank has the following coordinates:

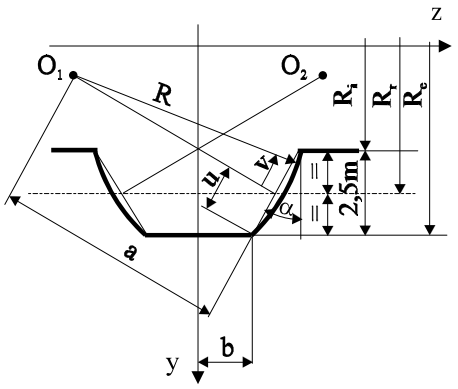


Figure 2: Worm flank geometry.

- For the right flank:

$$\begin{cases} X=0; \\ Y=Y_{O1}+R\cdot\cos(\frac{\pi}{2}-\alpha+v_1); \\ Z=Z_{O1}+R\cdot\sin(\frac{\pi}{2}-\alpha+v_1); \end{cases} \quad (2)$$

- For the left flank:

$$\begin{cases} X=0; \\ Y=Y_{O2}+R\cdot\cos(\frac{\pi}{2}-\alpha+v_2); \\ Z=Z_{O2}-R\cdot\sin(\frac{\pi}{2}-\alpha+v_2). \end{cases} \quad (3)$$

In the above relations, v_1 and v_2 are variable parameters of the right flank and left flank, respectively. Generally, the helical motion can be written by means of two coordinate transformations corresponding to simple motions, components of the helical motion: rotation about Oz axis, having parameter φ, and translation on the same axis, proportional to the rotation angle $p\cdot\varphi$, p being helical parameter. In this way, the helical motion of the movable coordinate system XYZ is described by the matrix equation:

$$x=\omega_3^T(\varphi)\cdot X+a \quad (4)$$

or

$$\begin{Vmatrix} x \\ y \\ z \end{Vmatrix}=\begin{Vmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{Vmatrix}\cdot\begin{Vmatrix} X \\ Y \\ Z \end{Vmatrix}+\begin{Vmatrix} 0 \\ 0 \\ p\cdot\varphi \end{Vmatrix} \quad (5)$$

where x is the matrix of a point coordinates with respect to the coordinate system xyz fixed to the frame, X is the matrix of the same point coordinates

with respect to the movable coordinate system, a is the matrix of the point O coordinates (the origin of the movable coordinate system) with respect to the point O_1 (see Figure 3), and $\omega_3(\varphi)$ is the matrix of the rotation transformation.

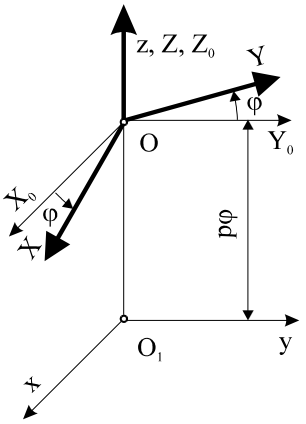


Figure 3: Coordinate system applied for the helical motion.

Substituting (1), (2) and (3) in (4), we obtain the parametric equations of the right flank surface and left flank surface.

Then, crossing these surfaces with the plane x=H, the curve representing the worm profile corresponding to the sectional plane takes the form (for example, the right flank):

$$\Sigma_{DH}\begin{cases}\sin\varphi_1=\dfrac{H}{-[Y_{O1}+R\cdot\sin(\alpha-v_1)]};\\ y=[Y_{O1}+R\cdot(\alpha-v_1)]\cdot\cos\varphi_1;\\ z=Z_{O1}+R\cdot\cos(\alpha-v_1)+p\cdot\varphi_1;\end{cases}\qquad(6)$$

2.2 Determination of the Worm-Gear Flank Profile

The worm-gear tooth surface is generated by the rolling.

We apply the "minimum distance method" on the algorithm of the discretization in the case of generation with the rack-bar tool.

First of all, we get the discretization of the generating curve C_Σ, which in this case is the worm profile, represented by the vector (7), where:

y_i and z_i are the coordinates of the profile from the "H" plane, which were determined by (5).

The gear flank generation of the elementary gear drive is made with the rack-bar tool (see Figure 4).

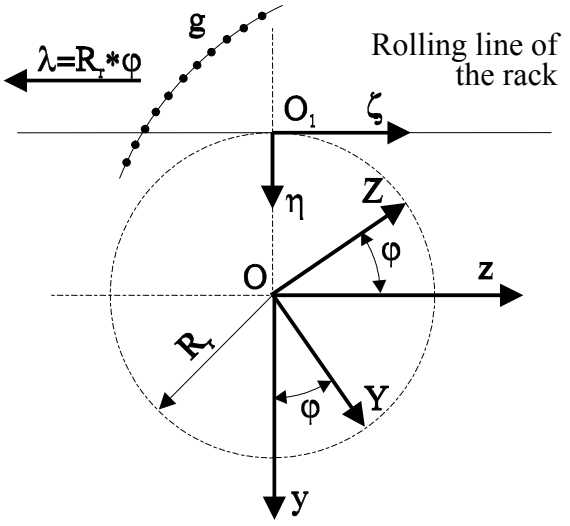


Figure 4: Worm-gear flank generation.

$$g=\begin{Vmatrix}y_1 & z_1\\ y_2 & z_2\\ . & .\\ . & .\\ . & .\\ y_n & z_n\end{Vmatrix}\qquad(7)$$

The rolling condition interpreted in "discrete" way is the following:

$$K\cdot\Delta\lambda=R_r\cdot\Delta\varphi\cdot j\qquad(8)$$

Where $\Delta\varphi$ is the angular increment of the rolling. It is then obvious that for generating a profile with high accuracy from the technical viewpoint, this increment has to be enough small.

2.2.1 Generation Motion

The generation motions of the worm-gear flank are:

1) Rotation of centroid, associated to the gear of the elementary gear drive, with respect to the fixed coordinate system xyz, described by the matrix equation

$$x=\omega_1^T(j\cdot\Delta\varphi)\,X.\qquad(9)$$

In this relation, x is the matrix of the point coordinates with respect to the fixed coordinate system, X is the coordinates matrix of the same point with respect to movable coordinate system XYZ and $\omega_1(\varphi)$ is matrix of the rotation transformation about O_x axis;

2) Translation of the movable coordinate system $\xi\eta\zeta$ associated to the rack, with respect to the fixed

coordinate system, described by the equation (a is the coordinates matrix of the point O_1, the origin of the movable coordinate system, with respect to the point O):

$$x=\xi+a \tag{10}$$

with

$$a=\begin{Vmatrix} 0 \\ -R_r \\ -R_r\cdot(j\cdot\Delta\varphi) \end{Vmatrix}. \tag{11}$$

3) Relative motions

Combining (9) and (10) we obtain that the motion equation of a point on the generating curve "g" (Figure 4) from the coordinate system XYZ with respect to the coordinate system $\xi\eta\zeta$ is as follows:

$$\xi=\omega_1{}^T(j\cdot\Delta\varphi)\ X-a \tag{12}$$

$$X=\omega_1{}^T(j\cdot\Delta\varphi)[\xi+a]. \tag{13}$$

From the last equation, we infer that

$$\begin{cases} X=\xi; \\ Y=(\eta-R_r)\cdot\cos(j\cdot\Delta\varphi)+[\zeta-R_r\cdot(j\cdot\Delta\varphi)]\cdot \\ \sin(j\cdot\Delta\varphi); \\ Z=-(\eta-R_r)\cdot\sin(j\cdot\Delta\varphi)+[\zeta-R_r\cdot(j\cdot\Delta\varphi)]\cdot \\ \cos(j\cdot\Delta\varphi). \end{cases} \tag{14}$$

The system of equations (14) represents the family of generating curves "g" with respect to the coordinate system of the worm-gear, η and ζ being the coordinates of the points that are on the generating curve (Figure 5).

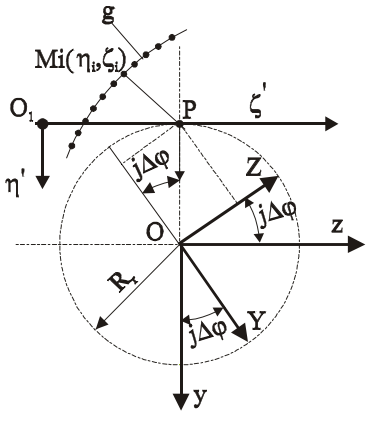


Figure 5: Coordinates of the meshing pole P.

The envelope to the family (14) is what we have to determine, more precisely, the gear profile (see Figure 6, flank of gear).

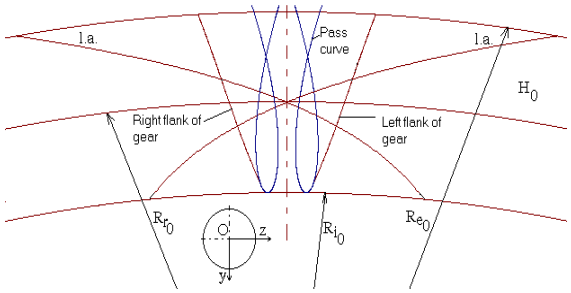


Figure 6: Line of contact (l.a.) in the plane H_0.

The enveloping condition is given by the minimum of distance

$$d=\sqrt{(Y-Y_P)^2+(Z-Z_P)^2}\ , \tag{15}$$

where Y_P and Z_P (coordinates of the meshing pole) are:

$$Y_P=-R_r\cdot\cos(j\cdot\Delta\varphi);\ Z_P=R_r\cdot\sin(j\cdot\Delta\varphi). \tag{16}$$

2.3 Surface of Contact

The surface of contact is defined as locus of the contact points of the two conjugated surfaces (which are in enveloping) in the fixed coordinate system xyz (Figure 4). The parametric equations of the surface of contact are obtained associating the enveloping condition to the absolute motion equation of the worm-gear flank profile. In the sectional plane x=H, the line of contact is given by:

$$\begin{cases} y=Y\cdot\cos(j\cdot\Delta\varphi)-Z\cdot\sin(j\cdot\Delta\varphi); \\ z=Y\cdot\sin(j\cdot\Delta\varphi)+Z\cdot\cos(j\cdot\Delta\varphi). \end{cases} \tag{17}$$

3 WORM-GEARING TOOTH RIGIDITY

Once the algorithm for the determination of the contact points, both on the flank height and along the line of contact, is performed, then it is possible to evaluate the rigidity of the worm-gearing tooth.

3.1 Bases of Design

The mathematical model is based on the following assumptions:

- The worm-gearing is errors free and the gears are rigid except the teeth;
- Taken into consideration only the bending produced by the meshing normal force;

- Consider that the worm-gearing consists of more plane-gear drives (pinion-rack drives), named "elementary gear drives", that in fact are cross sections perpendicular to axis of rotation of the worm-gear (Figure 1);
- The tooth of the elementary gear drive is supposed to be a beam fixed at one end in the body of gear;
- The assembly of the plane-gear drives into the worm-gear drive was made provided that the teeth of the elementary gear drives to deform together and not separately under the same load.

3.2 Computer Program

Our algorithm to evaluate the rigidity of the worm-gearing tooth is the following (Ghelase, D., 2005):

1) Computation of the rigidity for an elementary tooth;
2) Computation of the rigidity for a pair of elementary teeth;
3) Computation of the rigidity for an elementary gearing tooth (pinion-rack drive);
4) Computation of the rigidity for the worm-gearing tooth.

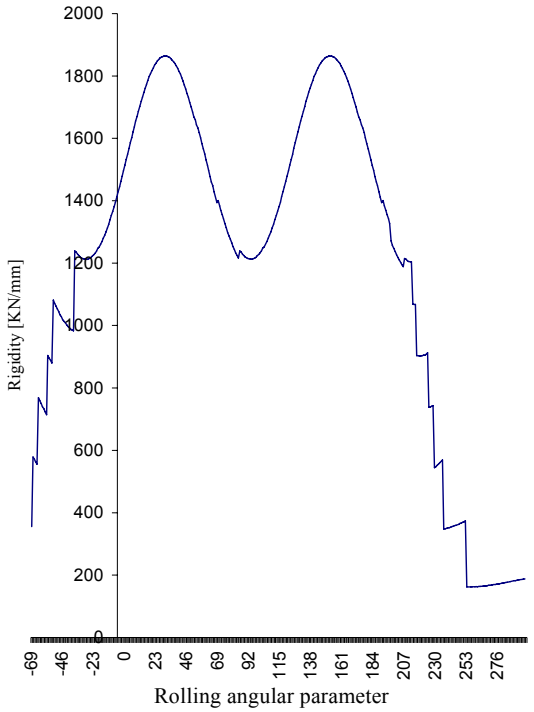


Figure 7: Rigidity of the worm-gearing tooth.

By means of the numerical modelling, these steps will be added to the computer program used for the study of the worm-gearing tooth geometry, finally providing the instrument for the determination of the worm-gearing tooth rigidity. The computation diagram of rigidity of worm-gearing tooth can be seen in Figure 7.

The cvasisinusoidal zone of the curve from Figure 7 repeats periodically, because it represents the rigidity during the meshing when the all plane-gear drives are involved in the meshing. Thus, if the input and output rigidities are eliminated, being less importing for our study, we get the elasticity characteristic of the worm-gearing tooth.

3.3 Elasticity Characteristic

The elasticity characteristic represents the variation of rigidity of the worm-gearing tooth depending on the rolling angle ($j \cdot \Delta\varphi$), where "j" is the rolling angular parameter (Ghelase, D., Tomulescu, L., 2003). It is cvasisinusoidal curve with the high jumps when a tooth binds or recesses (Figure 8).

The investigation of the elasticity characteristic is very important for the study of an elastic system, such as: gearing, linkage. Hence, the introduction of this concept contributes to the completion of the used gearing study and it leads to increase of the gearing tooth rigidity.

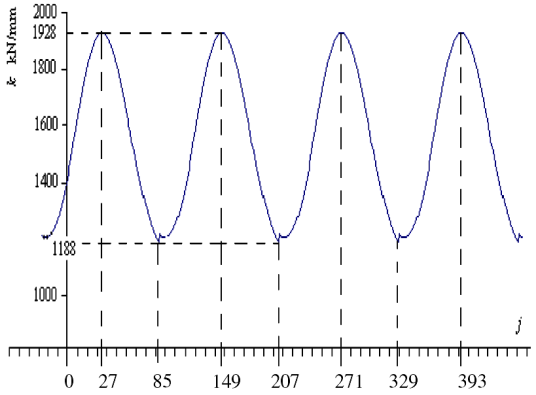


Figure 8: Elasticity characteristic of the worm-gearing tooth.

3.4 Influence of Geometrical Parameters

The influence of the geometrical parameters on the rigidity was obtained by means of the computerized simulation (Ghelase, D., 2003). It was applied to 150 worm-gear drives and we can present the following conclusions:

1. The rigidity of worm-gearing tooth increases if diametral quotient q increases and radius of profile curvature R increases (Ghelase, D., 2003).

2. The rigidity of worm-gearing tooth reduces if profile angle α increases (Ghelase, D., 2003) and number of the gear teeth z_2 increases (Figure 9, Table 1).

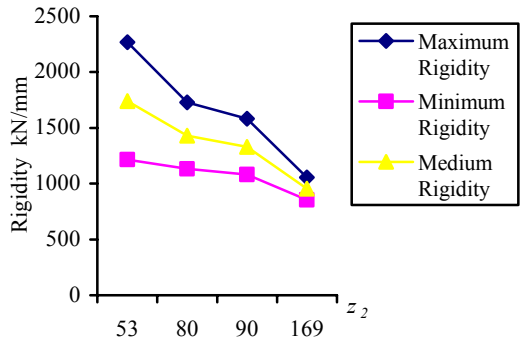


Figure 9: Rigidity depending on number of gear teeth z_2.

Table 1: Influence of number of gear teeth on rigidity.

z_2	Maximum Rigidity [kN/mm]	Minimum Rigidity [kN/mm]	Medium Rigidity [kN/mm]
53	2267.385	1215.140	1741.262
80	1727.633	1132.201	1429.917
90	1581.896	1079.696	1330.796
169	1055.990	853.826	954.908

4 CONCLUSIONS

Finally, we can draw the following conclusions:

1) A method to evaluate the rigidity of worm-gearing tooth was developed;

2) The proposed approach may be applied for any types of cylindrical worm-gearing and for spur gearing and bevel gearing. These mechanisms are in the structure of robot and by them rigidity depends the kinematical accuracy of robot;

3) The introduction of "elasticity characteristic" concept contributes to the completion of study for the used mechanisms;

4) The developed computer program enables to obtain numerical solutions and graphic illustration;

5) The numerical method, proposed and analyzed in this paper, affords the geometry optimization and the study of the meshing for various geometrical parameters of the worm-gearing, being in fact a simulation of meshing;

6) Moreover, we can determine the parameters which influence the improvement of rigidity for worm-gearing tooth and the increase of accuracy of robot linkages.

REFERENCES

Ghelase, D., Daschievici, L., 2006. Computerized Design-Generation of the Worm-Gear Flank. *The Archive of Mechanical Engineering– Polish Academy of Sciences Committee of Machine Design , 2,* 165-178.

Ghelase, D., Gratie, L., 2005. Numerical Computation Rigidity of Worm-Gearing Tooth with Circulare Profile. In *IEEE ICIT2005, International Conference on Industrial Techology.* City University of Hong Kong.

Ghelase, D., Tomulescu, L., 2003. Computerized Determination of the Elasticity of the Worm-Gearing Tooth for Machine-Tools and Robots. In *Machine-Building and Technosphere of the XXI Century, International Science and Engineering Conference.* Donetsk National Technical University.

Ghelase, D., 2003. Influence of the Geometrical Parameters on Rigidity of the Worm-Gearing Tooth. *The Annals of "Dunarea de Jos" University of Galati, Fascicle XIV,* 45-48.

DRIVER'S DROWSINESS DETECTION BASED ON VISUAL INFORMATION

Marco Javier Flores, José María Armingol and Arturo de la Escalera
Intelligent System Laboratory, Universidad Carlos III de Madrid, Leganés 28911, Madrid, Spain
{mjflores, armigol, escalera}@ing.uc3m.es

Keywords: Drowsiness, driver assistance system, object detection, support vector machine, intelligent transportation technology.

Abstract: In this paper, a new Driver Assistance System (DAS) for automatic driver's drowsiness detection based on visual information and image processing is presented. This algorithm works on several stages using Viola and Jones (VJ) object detector, expectation maximization algorithm, the Condensation algorithm and support vector machine to compute a drowsiness index. The goal of the system is to help in the reduction of traffic accidents caused by human errors. Examples of different driver's images taken over a real vehicle are shown to validate the algorithm.

1 INTRODUCTION

Active Security, whose objective is to endow vehicles with intelligent systems that predicts and avoids accidents, has acquired a growing interest and it has become one of the most important research fields in the transport security. Indeed, DAS objective is to contribute in traffic accident reduction by using new technologies; this is, increasing the vehicles security, and at the same time, decreasing the danger situations that may be generated during driving process.

Current research is interested in the study of driver's state behavior; in this ambitious research, it has taken relevance the driver's drowsiness study, also denominated fatigue and related closely with distraction. Drowsiness is presented in stress and fatigue situations in an unexpected and inopportune way. The dream sensation generates the decrease vigilance level state, and this factor produces danger situations and increases the probability of causing some accident. Drowsiness may also be produced by dream's illnesses, certain type of medications, and even, bored situations, such as driving for a long time. It has been estimated that drowsiness produces among 10% and 20% of traffic accidents with dead drivers (Tian and Qin, 2005) and hurt drivers (Dong and Wu, 2005). Whereas trucking industry produces 57% of fatal truck accidents for this fatality (Ji and Yang, 2002; Bergasa et al., 2004). Fletcher (Fletcher et al., 2003) goes further on and has mentioned that 30% of total traffic accidents have been produced by drowsiness. For these reasons, it is important to design systems that allow monitoring the drivers and measuring their level of attention during whole driving process. Fortunately, people in drowsiness produce several typical visual cues that are detected on the human face: yawn frequency, eye-blinking frequency, eye-gaze movement, head movement and facial expressions. Taking advantage of these visual characteristics; computer vision is the feasible and appropriate technology to treat this problem.

The organization of the paper is as follows. Section 2 presents an extended state of the art. Section 3 introduces the proposed method for face location and eye detection in detail. Finally, in section 4 results and conclusions are shown.

2 PREVIOUS WORK

Ji and Yang (2002) has presented a detection drowsiness system based on infrared light illumination and stereo vision. This system localizes the eye position using image differences based on the bright pupil effect. Afterwards, this system computes the blind eyelid frequency and eye gaze to build two drowsiness indices: PERCLOS and AECS. Bergasa and his colleagues (Bergasa et al., 2004) has developed a non-intrusive system that also uses infrared light illumination, this system computes driver vigilance level using a finite state automata

with six eye states that computes several indices, among them, PERCLOS; on the other hand, the system is able to detect inattention through face pose. Horng et al. (2004) has shown a system that uses a skin color model over HSI space for face detection, edge information for eye localization and dynamical template matching for eye tracking. Using color information of eyeballs, it identifies the eye state and computes the driver's state. Brandt et al. (2004) has shown a system that monitors the driver fatigue and inattention. For this task, he has used VJ method to detect the driver's face. Using the optical flow algorithm over eyes and head this system is able to compute the driver state. Tian and Qin (2005) have built a system for verifying the driver's eye state. Their system uses Cb and Cr components of the YCbCr color space; with vertical projection function this system localizes the face region and with horizontal projection function it localizes the eye region. Once the eyes are localized the system computes eye state using a complexity function. Dong and Wu (2005) have presented a system for driver fatigue detection, which uses a skin color model based on bivariate Normal distribution and Cb and Cr components of the YCbCr color space. After localizing the eyes, it computes the fatigue index utilizing the eyelid distance to classify between open eyes and closed eyes.

3 PROPOSED SYSTEM

In this paper, a system to detect the driver's drowsiness is presented; it works on grayscale images taken with the camera inside the IvvI (Intelligent Vehicle based on Visual Information) vehicle, Figure 1 (b). IvvI is an experimental platform used to develop driver assistance systems in real driver conditions. IvvI is a Renault Twingo vehicle, Figure 1 (a), equipped with a processing system which processes the information comes from the cameras. This system consists of several parts that will be described throughout this section.

3.1 Face Detection

To localize the face, this system uses VJ object detector which is a machine learning approach for visual object detection. It uses three important aspects to make an efficient object detector based on the integral image, AdaBoost technique and cascade classifier (Viola and Jones, 2001). Each one of these elements is important to process the images efficiently

(a) (b)

Figure 1: (a) IvvI vehicle, (b) Driver's camera.

and near real-time with 90% of correct detection. A further important aspect of this method is its robustness under changing light conditions. However, in spite of the above-mentioned, its principal disadvantage is that it can not extrapolate and does not work appropriately when the face is not in front of the camera axis. Such would be the case when the driver moves his/her head; however, this shortcoming will be analyzed later on.

(a) (b)

(c) (d)

Figure 2: Face and eye detection on different drivers.

Continuing with the algorithm description, when driver's face is detected, it is enclosed with a rectangle R which is addresses by left-top corner coordinates $P_0 = (x_0, y_0)$ and right-bottom corner coordinates $P_1 = (x_1, y_1)$, as can be observed in Figure 2 (a), (c). Indeed, rectangle size comes from experimental analysis developed on the face database that has been created for this task.

3.2 Face Tracking

The principal problem of VJ method is that it is only able to localize the human face when it is in frontal position at camera. This drawback induces to have an unreliable system to driver's analysis during all driving process. Much effort has been put to correct

this problem; so, using a dual active contour (Gun and Nixon, 1994; Dokladal et al., 2004) is able to solve this disadvantage and to track the face in the driving process appropriately.
This face tracker needs to be initialized to extract an approximation of the head boundary. This is done once the face has been located through the rectangle R of the previous section, the system automatically generates an internal and external ring around the face based on gradient information, continuing with the calculation of the energy of the two active contours, after that, it corrects the position that corresponds to the face contour model.

3.3 Eye Detection

Once the face has been located through the rectangle R in previous section, using the face anthropometric properties (Gejgus and Sparka, 2003) which come from face database analysis, two rectangles containing the eyes are obtained. Preliminary, we use R_L for left eye rectangle and R_R for right eye rectangle. The rectangles coordinates are presented in Table 1 and Figure 3 (a) shows some examples.

Table 1: Preliminary rectangles that contain the eyes.

	Left eye
Left top corner	$(u_{0L}, v_{0L}) = (x_0 + w/6, y_0 + h/4)$
Right bottom corner	$(u_{1L}, v_{1L}) = (x_0 + w/2, y_0 + h/2)$
	Right eye
Left top corner	$(u_{0R}, v_{0R}) = (x_0 + w/2, y_0 + h/4)$
Right bottom corner	$(u_{1R}, v_{1R}) = (x_1 - w/6, y_1 - h/2)$

where $w = x_1 - x_0$ and $h = y_1 - y_0$.
After the previous step; the exact position of each eye will be localized, incorporating information from grey-level pixels through the following algorithm:

- Generate the image J by means of the following equation:

$$J(x,y) = \frac{I(x,y) - m}{\sigma} \quad (1)$$

where m and σ are the mean and the standard deviation, respectively. They are computed over the eye rectangles described in Table 1, and $I(x,y)$ is the pixel value in the position (x,y).

- Generate the image K using the equation:

$$K(x,y) = \begin{cases} J(x,y) - 256\delta_1 & if \quad J(x,y) \geq 0 \\ 256\delta_2 + J(x,y) & if \quad J(x,y) < 0 \end{cases} \quad (2)$$

where $\delta_1 = \max(0, ceil(J(x,y)/256) - 1)$, $\delta_2 = \max(1, ceil(|J(x,y)|/256))$ and $ceil(x)$ is the function that returns the smallest integer larger than x.

- Obtain the binary image , B , from image K through the equation (3), namely,

$$B(x,y) = \begin{cases} 255 & if \quad K(x,y) \geq \kappa \\ 0 & other \quad case \end{cases} \quad (3)$$

where κ is computed by Ostu's method (Otsu, 1979), Figure 3 (b).

- Compute the gradient image, G, using the Sobel horizontal and vertical edge operator followed by an image contrast enhancement (Jafar and Ying, 2007), Figure 3 (d).
- Compute the logarithm image, L, with the objective to enhance the iris pixels that are the central part of the eye (Wu et al., 2004), Figure 3 (e).

All previous information produces a random sample that comes from a distribution function that it has an elliptic shape; i.e., the pixels coming from each eye through the images B, G and L can be viewed as a realization of a random variable. Having specified all the data describing the model, to obtain the parameters of this function the expectation maximization algorithm (EM) has been used. Special attention has received the ellipse center, because, it allows to obtain the exact position of the eye center. The ellipse axes determine the width and height of the eyes. The result is shown in Figure 3 (c), (f), while in Figure 2 (b), (d) the eye position generated for this procedure is depicted. The expectation maximization algorithm computes the mean, variance and the correlation of X and Y coordinates that belong to the eye. The initial parameters to run EM are obtained from a regression model adjusted with the last square method. These parameters will be used in the eye state analysis below.

3.4 Eye Tracking

There are a number of reasons for tracking. One is the VJ's problems mentioned above. Another is the necessity to track the eyes continuously from frame to frame. A third reason is to satisfy the real-time conditions reducing the eye search space. For this task; the Condensation algorithm that was proposed

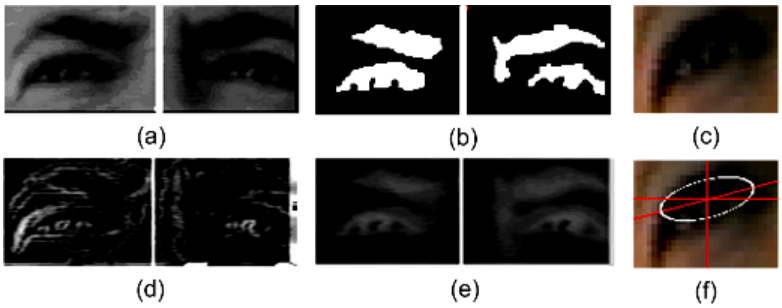


Figure 3: Eye location through R_L and R_R and Expectation Maximization algorithm over a spatial distribution of the eye pixels: (a) grayscale image, (b) binary image B, (d) gradient image G, (e) logarithm image L, (c) right eye image, and (f) ellipse parameters: center position, axes and inclination angle.

by Isard and Blake (1998) for tracking active contours using a stochastic approach has been used. The Condensation algorithm combines factored sampling with a dynamical model that is governed through the state equation:

$$X_t = f(X_{t-1}, \xi_{t-1}) \tag{4}$$

where X_t is the state at instant t, $f(\cdot)$ is an nonlinear equation and depends on a previous state plus a white noise. The goal is to estimate the state vector X_t with the help of systems observation which are realization of the stochastic process Z_t governed by the measurement equation:

$$Z_t = h(X_t, \eta_t) \tag{5}$$

where Z_t is the measure system at time t, $h(\cdot)$ is another nonlinear equation that links the present state plus a white noise. The processes ξ_t and η_t are each one white noise and independent among them. It must be pointed out that X_t is an unobservable underlying stochastic process, for this problem, it is eye position over the image and its velocity:

$$X_t = (x, y, \dot{x}, \dot{y})^T \tag{6}$$

The Condensation is initialized when the eyes are localized with the method of previously explained. Table 1 shows the eye tracking results that has been developed in two sequences of images.

3.5 Eye State Detection

To identify drowsiness through eye analysis is necessary to know its state (open or closed) and develop an analysis over the time. The classification among the open and closed state is complex due to the changing shape of the eye, among other factors, changing position and face rotating, twinkling and illumination variations. All this makes difficult to use only color cues to analyze eye in a reliable manner. For the problems that have been exposed a supervised classification method has been used for this challenging task, in this case, support vector machine (SVM) classification (Cristianini and Shawe-Taylor, 2000; Chang and Lin, 2001) which is rooted in statistical learning theory. SVM uses a training set, $S = \{(x_i, y_i) : i = 1, \cdots, m\}$, where x_i is the characteristic vector in R^n, $y_i \in \{1,2\}$ represents the class, in this case 1 for open eyes and 2 for closed eyes, and m is the number of elements of S. To do this work a training set has been built that consists of images of open eyes and images of closed eyes. The images come from diverse sources, under several illumination conditions and different races. A further important aspect of this eye database is that contains images of different eye colors, i.e., blue, black, green. Previous to SVM training, it is indispensable to process each image that consists on histogram equalization, filter with the median filter, followed by the sharpen filter and to normalize in the [0,1] interval. The median filter is used to reduce the image noise, whereas the sharpen filter is used to enhance the borders. The main objective of training SVM is to find the best parameters and the best kernel that minimizes the optimization problem (Chang and Lin, 2001), so, after several training experiments of the SVM algorithm, it has decided to use the RBF kernel, i.e., $K(x_i, x_j)$ is $\exp(-\gamma \|x_i - x_j\|^2)$, $C = 35$ and $\gamma = 0.0128$; these parameters reach high training classification rate that is about 94%.

Table 2: Result of eye tracking and eye state analysis.

	Total frames	Eye tracking		Eye state analysis		
		Tracking failure	Correct rate (%)	Eyes Open	Eyes Closed	Correct rate (%)
Video1	960	20	97.91	690/700	258/260	98.90
Video2	900	30	96.60	520/560	339/340	96.27

Figure 4: Different stage of the proposed algorithm on several instants of time and driving conditions.

3.6 Drowsiness Index

The eye-blinking frequency is an indicator that allows to measure driver's drowsiness (fatigue) level. As in the works of Horng et. al. (2004) and Dong and Wu (2005), if five consecutive frames or during 0.25 seconds are identified as eye-closed the system is able to issue an alarm cue. Table 2 also presents the result of eye state analysis over two sequences of images.

4 CONCLUSIONS

A non-intrusive driver's drowsiness system based on computer vision has been presented in this paper. This system uses visual information to analyze and to monitor driver's eye state at near real-time and real-driving conditions, i.e., external illuminations interference, vibrations, changing background and facial orientations changing. Experiments were carried out in the IvvI vehicle with different drivers. This guarantees and confirms that these experiments have proven robustness and efficiency in real traffic scenes. Another drowsiness indexes will be implemented as future works and they will be compared. Figure 4 shows an example that validates this system.

REFERENCES

Viola P. and Jones M., 2001: Rapid Object Detection using a Boosted Cascade of Simple Features. *Accepted Conference on Computer Vision and Pattern Recognition.*

Horng W., Chen C. and Chang Y., 2004: Driver Fatigue Detection Based on Eye Tracking and Dynamic Template Matching. *Proceedings of the IEEE International Conference on Networking, Sensing & Control.*

Tian Z. and Qin. H., 2005: Real-time Driver's Eye State Detection. *IEEE International Conference on Vehicular Electronics and Safety, Pg. 285-289.*

Ji Q. and Yang. X., 2002: Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance. *Real Time Imaging, Nr. 8, Pg. 357-377, Elsevier Science Ltd.*

Bergasa L., Nuevo J., Sotelo M. and Vazquez M., 2004: Real Time System for Monitoring Driver Vigilance. *IEEE Intelligent Vehicles Symposium.*

Isard M. and Blake A., 1998: Condensation: conditional density propagation for visual tracking. *International Journal on Computer Vision, 29(1), pp. 5-28.*

Cristianini N. and Shawe-Taylor J., 2000: An introduction to Support Vector Machines and other kernel-based learning methods. *Cambridge University Press.*

Chang C. and Lin C., 2001: LIBSVM: a library for support vector machine, URL: www.csie.ntu.edu.tw /~cjlin/libsvm

Otsu N., 1979: A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man and Cybernetics, Vol. 9, pp. 62-66.*

Gejgus P. and Sparka M., 2003: Face Tracking in Color Video Sequences. *The Association for Computing Machinery Inc.*

Brandt T., Stemmer R., Mertsching B., and Rakotomirainy A., 2004: Affordable Visual Driver Monitoring System for Fatigue and Monotony. *IEEE International Conference on Systems, Man and Cybernetics. Vol. 7, pp. 6451-6456*.

Fletcher L., Petersson L. and Zelinsky A., 2003: Driver Assistance Systems based on Vision In and Out of Vehicles. *IEEE Proceedings of Intelligent Vehicles Symposium, pp. 322-327.*

Wu Y., Liu H. and Zha H., 2004: A New Method of Detecting Human Eyelids Based on Deformable Templates. *IEEE International Conference on Systems, Man and Cybernetics, pp. 604-609*.

Jafar I., Ying H., 2007: A new method for Image Contrast Enhancement Based on Automatic Specification of Local Histograms. *IJCSNS International Journal of Computer Science and Network Security, Vol.7 No.7, July*.

Dong W. Wu X., 2005: Driver Fatigue Detection Based on the Distance of Eyelid. *IEEE Int. Workshop VLSI Design & Video Tech. Suzhou-China*.

Gunn S. R. and Nixon M. S., 1994: A Dual Active Contour for Head Boundary Extraction. *IEEE Colloquium on Image Processing for Biometric Measurement, pp. 6/1 - 6/4, London.*

Dokladal P., Enficiaud R. and Dejnozkova E., 2004: Contour-Based Object Tracking with Gradient-Based Contour Attraction Field. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'04), vol. 3, pp. iii-17-20.*

CALIBRATION ASPECTS OF MULTIPLE LINE-SCAN VISION SYSTEM APPLICATION FOR PLANAR OBJECTS INSPECTION

Andrei Hossu and Daniela Hossu
University Politehnica of Bucharest, Faculty of Control and Computers
Dept. of Automatics and Industrial Informatics, 313 Spl. Independentei, sector 6, RO-77206, Bucharest, Romania
hossu@aii.pub.ro, dana@aii.pub.ro

Keywords: Industrial Vision System, Line-Scan Camera, Dual-Camera Vision System, and Moving Scene in Robotic Automation.

Abstract: The System Set-up Time it is one of the characteristics of an Industrial Vision System, besides the accuracy performances and response time. Minimizing the set-up time while keeping the performances in accuracy and in response time is one of the goals of any advanced Vision System. Starting from the purpose and the required performances of the proposed Industrial Vision System, in the paper is presented a calibration method developed for a multiple line-scan camera Vision System (in particular for a dual line-scan camera system). The calibration method presented is based on analyzing the image of a calibration tool exposed to the Vision System. There are presented the type of dimensional distortions identified from the experimental results. The second part of the paper presents the calibration method. The Industrial Vision System described in the paper is designed for silhouette inspection of planar objects located on a moving scene (transport conveyor), in a robotic handling application (it is a pure 2D Vision System, the volumetric characteristics of the analyzed objects being not relevant for the application). However the height of the object is varying in time (from one set of objects to another). Due to the fact the distance between the cameras and the objects is changing, the measuring results are affected. The proposed calibration method allows the Vision System to self adjust the calibration parameters for a known change in height of the objects, without affecting the accuracy system performances. In the final section of the paper are presented some practical aspects of the proposed calibration method, and the balance between the off-line and the on-line required computational efforts from the Vision System.

1 MULTI LINE-SCAN CAMERA VISION SYSTEMS CHARACTERISTICS

The class of the Artificial Vision Systems dedicated for analyzing objects located on a moving scenes (conveyor) presents some specific characteristics relative to the Artificial Vision Systems dedicated for static scenes. These characteristics are identified also on the Image Calibration Process (Borangiu, et al., 1994)., (Haralick and Shapiro, 1992).

Figure 1 presents the model of the image obtained from a dual line-scan camera Vision System.

For this class of the Artificial Vision Systems we could identify as relevant for the calibration process the following characteristics:

- The system is using line-scan cameras for the image acquisition.
- The system is a dual-camera.
- The obtained image has significant distortions on (and only on) the image sensors direction.
- There is an overlapped image area between the two cameras. The end of the acquisition line of the 1st camera is overlapping the beginning of the acquisition line of the 2nd camera. This overlapping area is significant in dimension and is a constant parameter resulted during the artificial vision system installation process.
-There is a lengthwise conveyor distance between the acquisition lines of the two cameras. This distance is also a constant parameter and its value is fixed during the system installation process.

2 THE PATTERN BASED CALIBRATION TOOL

For the calibration process we adopted the method of using a Pattern based Calibration Tool.
This Pattern based Calibration Tool represent a set of blobs with a priori known dimensions and locations for the real world (millimeters and not image pixels) (Croicu, et al., 1998).

The outcome of using this type of calibration technique was to obtain the following:

- Estimation with the highest accuracy of the scene model parameters on the distortions direction
- Estimation of the size of the overlapped image area for both cameras
- The parallelism of the two acquisition lines if obtain during the installation process, using the support of the Calibration Tool
- The accuracy of installing the cameras in such a way to obtain the perpendicularity of the acquisition lines on the moving direction of the scene (of the conveyor).
- Obtain a high accuracy on the distance lengthwise the conveyor of the acquisition lines. The shape and the dimensions of the pattern adopted for the Calibration Tool force this characteristic.

3 CALIBRATION TOOL DESCRIPTION

In Figure 2 it is presented the pattern adopted for the Calibration Tool used for the dual line-scan camera Vision System (the dimensions are presented in millimeters) (Croicu, et al., 1998)., (Hossu, et al., 1998).

The characteristics of the adopted Pattern:

- The pattern contains dark blebs (marks) placed on a bright background (with a high level of light intensity for the image)
- The pattern is symmetrical on the vertical direction (lengthwise the conveyor). The two cameras have the acquisition lines parallel one each other but located on different position on the conveyor (due to the lighting system adopted – built from two fluorescent tubes used for obtaining the image from the reflection on the object surface). 1st Camera will have the acquisition line located on the top edge of the lower section of the pattern, and the 2nd Camera will locate its acquisition line on the bottom edge of the upper section of the pattern.
- The pattern is partially homogenous on the horizontal axis (the direction crosswise the conveyor, the direction of the distortions)
- The pattern contains a characteristic of a small difference (1 mm.) between the even and the odd marks. This will force the installation process to be very accurate in obtaining

the parallelism of the acquisition lines of the cameras and also the perpendicularity on the conveyor direction.

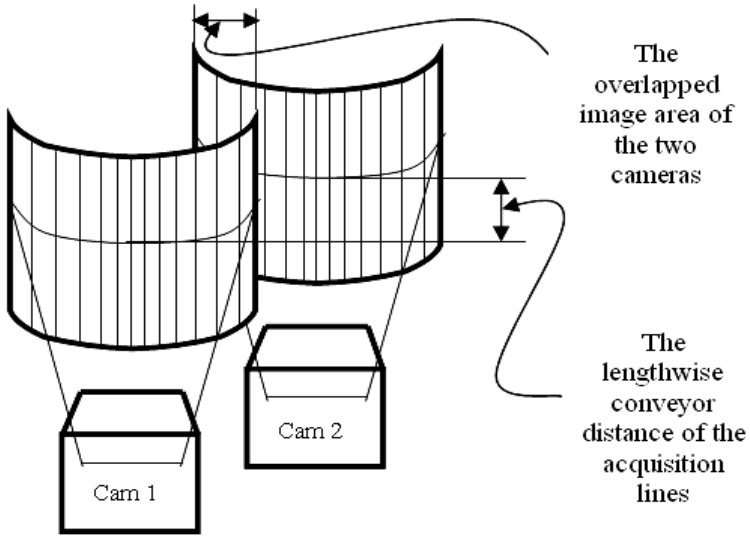


Figure 1: The image obtained from a dual line-scan camera Vision System.

4 EXPERIMENTAL RESULTS OF THE CALIBRATION PROCESS

In Figure 3 are presented the results obtained from the Calibration process performed on the 1st Camera.

The Excel Cell used as support for representing the results of the Calibration on the 1st Camera contains the following:

- The 1st column (called Mark) contains the number of the corresponding Mark existing in the pattern
- The 2nd column (called Calib. Cam) contains the values of the coordinates of the marks on the Calibration Tool.

These values are obtained from the "real world" from direct measuring of the Pattern applied on the Calibration Tool (represented in millimeters).

- The 4th column (called Pixel(x)) represents the coordinates of the existing Marks on the image. These coordinates are represented in pixel number.
- For both cameras we chose a Polynomial Trend of order 3 for the approximation of the conversion function of the image coordinates values (pixels) into the conveyor coordinates values (millimeters). Using this type of trend the calibration method will provide results with a maximum approximation error inside the accuracy requirements of the particular Vision System
- On the top and right side of the Excel Cell (the first two rows and the last four columns) are stored the parameters of the 3rd Order Polynomial estimated as trend.
- The last column (called Estimated (y)) contains the estimated of the marks coordinates values (on the conveyor), obtained from applying the Polynomial trend of order 3.
- In the bottom right side of the Excel Cell is presented the graphical chart of the trend of the coordinates values of the marks on the conveyor related to the pixel coordinates.

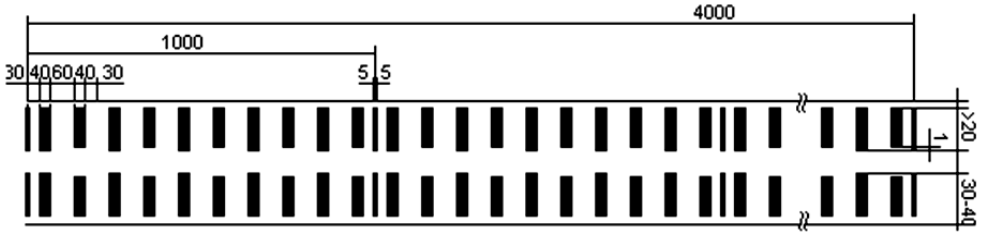


Figure 2: The pattern of the Calibration Tool used for the dual line-scan camera Vision System.

	a1	b1	c1	d1
	1.448567E-09	-9.128585E-06	0.5088785	338.9459

Mark	Calib Cam1	Real Cam1	Pixel (x)	Estimated (y)
1	436	429.08037	175	427.7278
2	535.5	529.18545	380	531.0810
3	635	629.31390	576	629.3081
4	734.5	729.46574	777	729.5128
5	834	829.64098	979	829.7479
6	933.5	929.83962	1181	929.5853
7	1036	1033.08381	1390	1032.5400
8	1135.5	1133.32999	1595	1133.2616
9	1235	1233.59960	1799	1233.3085
10	1334.5	1333.89265	2004	1333.7361
11	1434.5	1434.71331	2210	1434.6181
12	1534	1535.05337	2415	1535.0504
13	1633.5	1635.41690	2621	1636.0883
14	1733	1735.80390	2825	1736.3342
15	1832.5	1836.21437	3028	1836.3487
16	1932	1936.64834	3231	1936.6953
17	2036.5	2042.15453	3443	2041.9242
18	2136	2142.63667	3644	2142.1760
19	2235.5	2243.14232	3844	2242.4668
20	2335	2343.67150	4046	2344.3760

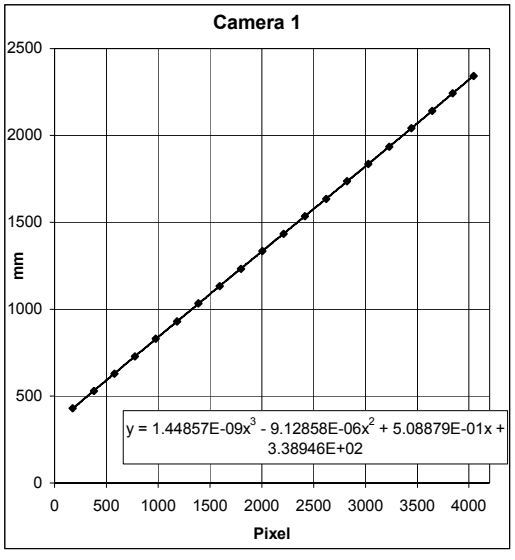


Figure 3: Experimental results of the calibration process on the 1st camera.

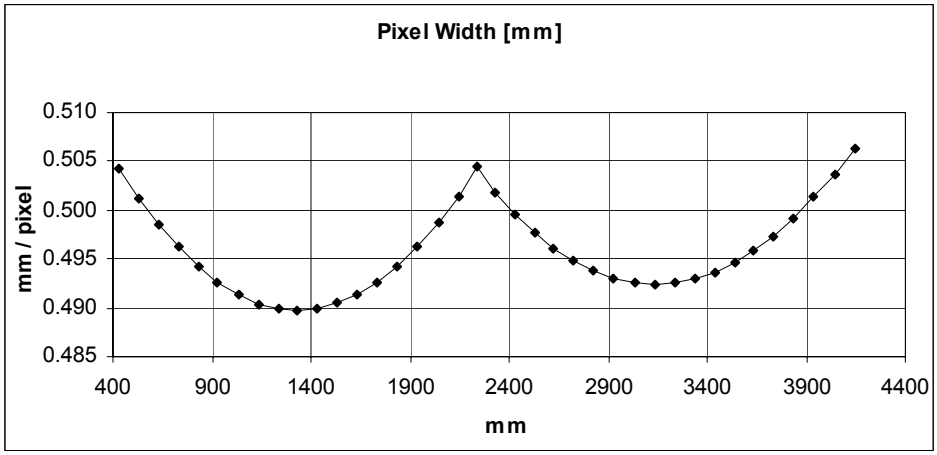


Figure 4: The distortions estimated on the acquisition lines direction.

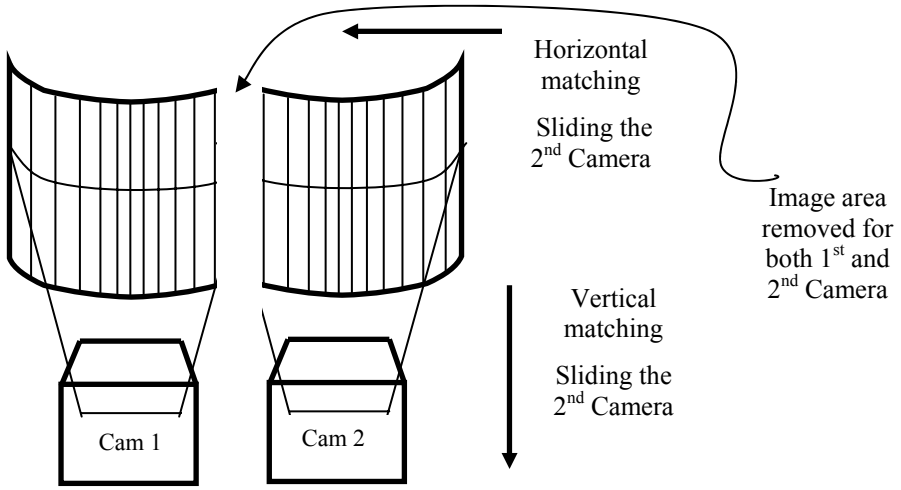


Figure 5: Using the calibration results for a dual line-scan camera system.

In the figure we are using the following notations:

- x coordinate value in image – **(pixels)**
- y coordinate value on the objects scene (on conveyor) – **(mm)**
- $y = a_1 x^3 + b_1 x^2 + c_1 x + d_1$ for the 1st Camera and
- $y = a_2 x^3 + b_2 x^2 + c_2 x + d_2$ for the 2nd Camera

In Figure 4 is presented the graphical chart of the evolution of the distortions estimated on the acquisition lines direction. The two cameras are covering around 4 meters wide view. The figure represents the behavior of the two acquisition cameras: in the are from 400 mm to 2250 mm it is represented the behavior of the 1st camera and in the area from 2250 mm to 4400 mm it is represented the behavior of the 2nd camera.

We can notice the distortions are effecting the pixel width of the image form 0.49 mm/pixel to 0.507 mm/pixel.

Ignoring this variation of the pixel width along the image acquisition line, would lead to accumulation of very high errors in some areas of the image, due to the fact the amount of pixels contained in an acquisition line is high (8096).

5 CONCLUSIONS

The most important thing obtained from estimating the scene parameters, is the estimation of the image distortions.

This end of the process of distortion estimation leads to obtaining a table for conversion the image coordinates (pixels) in scene coordinates (millimeters) for each of all the 8096 pixels of the acquisition lines. This lookup table will contain 8096 values (double representation) representing the real values of the scene (conveyor) coordinates of each pixel.

The CPU effort of the image processing algorithm will be minimal on converting the image coordinates into conveyor coordinates, using the pixel coordinate as the index of the offline built lookup table.

In order to minimize the processing time of the image, the polynomial estimation is not used on-line. The polynomial estimation is used for building the pixel to millimeters lookup table, in the offline stage of the presented method. In fact using a polynomial trend of 3rd order or other type of trend is not relevant for the response time performances of the vision system but only for obtaining the required approximation error. This stage, being offline and using a relative small amount of data, another more sophisticated approximation method could be used.

In Figure 5 are presented the ways the last two steps of the calibration process are performed.

REFERENCES

Borangiu, Th., Hossu, A., Croicu, A., 1994. ROBOT VISIONPro, *Users Manual, ESHED ROBOTEC*, Tel - Aviv.

Croicu, A., Hossu, A., Dothan, E., Ellenbogen, D., Livne, Y., 1998. ISCAN-Virtual Class based Architecture for Float Glass Lines, *IsoCE'98*, Sinaia.

Hossu, A., Croicu, A., Dothan, E., Ellenbogen, D., Livne, Y., 1998. ISCAN Cold-Side Glass Inspection System for Continuous Float Lines, *User Manual, Rosh-Haayn.*

Haralick, R., Shapiro, L., 1992. *Computer and Robot Vision*, Addison-Wesley Publishing Company.

TWO LAYERS ACTION INTEGRATION FOR HRI
Action Integration with Attention Focusing for Interactive Robots

Yasser Mohammad and Toyoaki Nishida
Graduate School of Informatics, Kyoto University, Kyoto, Japan
yasser@ii.ist.i.kyoto-u.ac.jp, nishida@i.kyoto-u.ac.jp

Keywords: HRI, Action Selection, Natural Listening.

Abstract: Behavior architectures are widely used to program interactive robots. In these architectures multiple *behaviors* are usually running concurrently so a mechanism for integrating the resulting actuation commands from these *behaviors* into actual actuation commands sent to the robot's motor system must be faced. Different architectures proposed different action integration mechanisms that range from distributed to central integration. In this paper an analysis of the special requirements that HRI imposes on the action integration system is given. Based on this analysis a novelle hybrid action integration mechanism that combines distributed attention focusing with a fast central integration algorithm is presented in the framework of the EICA architecture. The proposed system was tested in a simulation of a listener robot that aimed to achieve human-like nonverbal listening behavior in real world interactions. The analysis of the system showed that the proposed mechanism can generate coherent human-like behavior while being robust against signal correlated noise.

1 INTRODUCTION

Behavioral architectures are widely used to program interactive robots (Ishiguro et al., 1999), (S Karim, 2006). Behavioral architectures employs a vertical decomposition of the software design into a set of corunning *behaviors* that continuously map the current state of the environment into commands to the actuators of the robot. A basic problem that must be solved in any such architecture is how to integrate the results of multiple running processes into a final command sent to the actuators of the robot. The proposed solution to this problem (referred to as the action integration problem hereafter) can be divided into selective solutions that selects the action of a single behavior at any point of time to control the robot and combinative solutions that generates the final commands based on the proposed responses of multiple behaviors and hybrid solutions that tries to combine cooperation with coordination. Another dimension of comparison between action integration solutions is whether or not a central integrator is employed. According to this dimension available solutions are central or distributed. The action integration mechanism proposed in this paper can be classified as a hybrid two layered architecture with distributive attention focusing behavior level selection followed by a fast central combinative integrator.

The next section formalizes the requirements for an action integration mechanism suitable for HRI applications. Section 3 introduces the L_0EICA architecture for which this action integration mechanism was designed. Section 4 gives the details of the action integration mechanism proposed in this paper and section 5 reports an experiment with a humanoid robot that used the proposed action integration mechanism.The paper is then concluded.

2 REQUIREMENTS

One goal of HRI research is to realize robots that can use human-like verbal and nonverbal interaction modalities. To achieve this goal researchers usually employ behavioral architectures because of their robustness and fast response. A special property of HRI applications is that the details of the external behavior of the robot will be interpreted by human users as intentional signals (Breazeal, 2005), and usually combining nonverbal messages results on absolutely different messages. This is why combinative architectures are not common in HRI research and most of the HRI specific architectures employ a selective

integration strategy. For example (Ishiguro et al., 1999) proposed a robotic architecture based on situated modules and reactive modules in which reactive modules represent the purely reactive part of the system, and situated modules are higher levels modules programmed in a high-level language to provide specific behaviors to the robot. The situated modules are evaluated serially in an order controlled by the module controller. Research in nonverbal communication in humans reveals a different picture in which multiple different processes do collaborate to realize the natural action. For example (Argyle, 2001) showed that human spatial behavior in close encounters can be modeled with two interacting processes. It is possible in the selective framework to implement these two processes as a single behavior but this goes against the spirit of behavioral architectures that emphasizes modularity of behavior (Perez, 2003). This leads to the first requirement for HRI aware action integration: *The action integration mechanism should allow a continuous range from pure selective to pure combinative strategies*. In other words the system should use a *hybrid* integration strategy. The need to manage the degree of *combinativity* based on the current situation entails the second requirement: *The action integration mechanism should adapt to the environmental state using timely sensor information as well as the internal state of the robot*. In current systems this requirement is usually implemented by using a higher level deliberative layer but in many cases the interaction between simple reactive within the action integrator can achieve the same result as will be shown in the example implementation of this paper.

The Hybrid Coordination approach presented in (Perez, 2003) is the nearest approach to achieve this first requirement. In this system every two behaviors are combined using a *Hierarchical Hybrid Coordination Node* that has two inputs. The output of the HHCN is calculated as a nonlinear combination of its two inputs controlled by the activation levels of the source behaviors and an integer parameter k that determines how combinative the HHCN is, where larger values of k makes the node more selective. The HHCNs are then arranged in a hierarchical structure to generate the final command for every DoF of the robot (Perez, 2003). Although experiments with the navigation of an autonomous underwater robot have shown that the hybrid coordination architecture can outperform traditional combinative and selective architectures, it still has some limitations in the HRI domain. One major limitation of the hybrid coordination system is its reliance on binary HHCNs which makes it unsuitable for large numbers of behaviors due to the exponential growth in the number of HHCNs needed. Another problem is the choice of the parameter k for every HHCN. Yet the most difficult problem for this system is figuring out the correct arrangement of the behaviors into the HHCN inputs. This leads to the third requirement: *The action integration mechanism should not depend on global relationships between behaviors*. One of the major problems with this architecture is that every behavior must calculate its own activation level. Although this is easy for behaviors like *avoid-obstacles* or *go-to*, it is very difficult for interactive processes like *attend-to-human* because the achievement of such interactive processes is not manifested in an easily measurable specific goal state that must be achieved or maintained but in the exact way the overall behavior of the robot is changing over time. This leads to the fourth requirement: *The action integration mechanism should separate the calculation of behavior's influence from the behavior computation*.

The number of behaviors needed in interactive robots usually is very high compared with autonomously navigating robots if the complexity of each behavior is kept acceptably low, but most of those behaviors are usually passive in any specific moment based on the interaction situation. This property leads to the fifth requirement: *The system should have a built-in attention focusing mechanism*. HRI systems usually work in the real world with high levels of noise but it is required that the robot shows a form of goal directed behavior. This leads to the sixth requirement: *The action integration system should be robust against noise and data loss to provide a goal-directed behavior*.

In summary the six requirements HRI imposes on the action integration system are:

R1 It should allow a continuous range from pure selective to pure combinative strategies

R2 It should adapt to the environmental state utilizing timely sensor information.

R3 It should not depend on global relationships between behaviors

R4 It should separate the calculation of behavior's influence from the behavior's computation

R5 It should have built-in attention focusing

R6 It should be robust against noise and data loss.

Table 2 compares the action integration scheme of some well used behavioral architectures with the proposed system in terms of the six requirements

In this paper an action integration mechanism that has the potential of meeting these requirements is presented.

Table 1: Comparison of the Action Integration Capabilities of Some Behavioral Architectures in Terms of the Six Requirements in section 2.

Architecture	Integration	Implement.	R1	R2	R3	R4	R5	R6
Subsumption (Brooks, 1986)	Selective	Distributed			√		√	√
ASD (Maes, 1989)	Selective	Distributed				√		√
PDL (Steels, 1993)	Combinative	Central			√			
Motor Schemas (Arkin, 1993)	Combinative	Distributed			√		√	
Hybrid Coordination (Perez, 2003)	Hybrid	Distributed	√					√
AVB (Nicolescu and Matarić, 2002)	Selective	Distributed		√	√			√
Situated Modules (Ishiguro et al., 1999)	Selective	Central		√	√		√	√
Proposed System	Hybrid	Hybrid	√	√	√	√	√	√

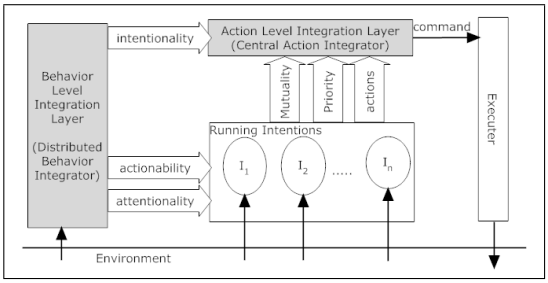


Figure 1: The Proposed Action Integration Mechanism.

3 L_0EICA

EICA is a behavioral hybrid architecture designed for HRI applications (Mohammad and Nishida, 2007). The basic behavioral element of EICA is the *intention*. Every intention implements a simple well defined reactive capability of the robot. Intentions are more like motor schema of (Arkin, 1993) than complete behaviors. Every intention has three attributes:

Attentionality a real number (0 1) that specifies the relative speed at which the intention is running.

Actionability a real number (0 ∞) specifying the activation level of the intention. A zero or negative activation level prevents the intention from execution. This attribute controls the influence of this intention on other running components.

Intentionality a real number (0 ∞) set by the distributed action integration layer and used by the central integrator to combine the actions of various intentions.

4 THE PROPOSED SYSTEM

Fig. 1 shows the block diagram of this hybrid system. The following subsections will detail the two layers of the system.

4.1 Behavior Level Integration

The goal of this layer of the action integration mechanism is to provide timely values for the *actionability*, *attentionality*, and *intentionality* for various intentions in the system. The first two of those parameters will determine how frequent every intention will be allowed to send actions to the central action integrator, and the third will be used by the action integrator, along with other parameters, to integrate the actions proposed by all the running intentions as will be detailed in the next section.

This layer is implemented as a set of *processes* connected together and to the intentions through *effect channels*.

Every process runs with a speed directly proportional to its *attentionality* level as long as its *actionability* is positive. The *processes* are connected together through *effect channels* in a network. Every *effect channel* has a set of n inputs that use continuous signaling and a single output that is continuously calculated from those inputs according to the *operation* attribute of the *effect channel*. The example presented in this paper uses the *Avg* operation defined as: $y = \frac{\sum_{i=1}^{n}(a_i x_i)}{\sum_{i=1}^{n}(a_i)}$, where x_i is an input port, a_i is the *actionability* of the object connected to port i and y is the output of the effect channel. Every *process* or *intention* has a single effect channel connected to its *attentionality* attribute and another effect channel connected to its *actionability* attribute, and effect channels can be arranged into hierarchical structures like the HHCNs in (Perez, 2003) but rather than carrying action and influence information, the effect channels carry only the influence information between different processes of the system.

4.2 Action Level Integration

This layer consists of a central fast action integrator that fuses the actions generated from various intentions based on the intentionality of their sources, the mutuality and priority assigned to them, and general parameters of the robot. The *actionability* of the action integrator is set to a fixed positive value to ensure that it is always running. The action integrator is an *active* entity and this means that its *attentionality* is changeable at run time to adjust the responsiveness of the robot. The action integrator periodically checks its internal master *command* object and whenever finds some action stored in it, the *executer* is invoked to execute this action on the physical actuators of the robot.

Algorithm 1 shows the *register-action* function responsible of updating the internal *command* object based on actions sent by various intentions of the robot. The algorithm first ignores any actions generated from intentions below a specific system wide threshold τ_{act}. The function then calculates the *total priority* of the action based on the intentionality of its source, and its source assigned priority. Based on the mutuality assigned to every degree of freedom (DoF) of the action, difference between the total priority of the proposed action and the currently assigned priority of the internal command, the system decides whether to combine the action with the internal command, override the stored command, or ignore the proposed action. Intentions can use the *immediate* attribute of the *action* object to force the action integrator to issue a command to the executer immediately after combining the current action.

Algorithm 1 : Register Action Algorithm.

function REGISTER-ACTION(Action a, Intention s)
  if $s.actionability < \tau_{act} \vee s.intentionality < \tau_{int}$ **then**
    exit
  end if
  $c \leftarrow$ current combined command
  $p \leftarrow a.priority + max\_priority \times s.intentionality$
  for every DoF i in the a **do**
    $combined \leftarrow true$
    if $p < c.priority \wedge s \neq c.source$ **then**
      $c.source \leftarrow s$
      $c.dof(i) \leftarrow a.dof(i)$
      $c.priority(i) \leftarrow p$
      $c.hasAction(i) \leftarrow true$
    end if
    if $c.source \neq null$ **then**
      $c.source \leftarrow null$
      $c.priority(i) \leftarrow \max(p, c.priority(i))$
    else
      $c.source \leftarrow s$
      $c.priority(i) \leftarrow p$
    end if
    if $a.mutual = true \vee c.source = s$ **then**
      $c.source \leftarrow s$
      $c.dof(i) \leftarrow a.dof(i)$
      $c.priority(i) \leftarrow p$
    else
      $c.dof(i) \leftarrow \frac{p \times a.dof(i) + c.priority \times c.dof(i)}{p + c.priority}$
    end if
    if $combined = true$ **then**
      return false
    end if
    $c.actionable \leftarrow true$
    if $a.notCombinableWithLower$ **then**
      $c.stopCombiningLower \leftarrow true$
    end if
    if $a.immediate$ **then**
      execute a
    end if
  end for
end function

4.3 Requirement Achievement

1. The system can achieve a continuous range of combinative to selective behavior based on the values of the actionability and attentionality of the intentions and the mutuality assigned to the actions. On one extreme τ_{act} can be set to the same value of the intention with the highest intention which will lead to a purely selective behavior. On the other hand τ_{act} and action mutuality can be set to zero while keeping the intentionality of all intentions equal which will lead to a purely combinative behavior.
2. The distributed layer of the system can use the sensor information and internal state stored in the processes to manipulate the tau_{act} parameter as well as the intentionality of various intentions to control the level of combinative behavior based on timely information from the environment.
3. The proposed action integration mechanism does not depend on any global relation between the intentions or the processes and adding new processes or intentions will only involve deciding the local effect, and data channels related to this new active component.

4. The influence of any intention on the final behavior of the robot is controlled based on its *intentionality* which is in turn controlled by the processes of the behavior integration layer rather than the intention itself. This provides the required separation between the calculation of influence and the computation of the basic behaviors.
5. Attention focusing is implemented in the lowest level of implementation in the proposed system and is controlled by the *attentionality* of various processes and intentions. This calculation is separated from the influence calculation through the *actionability* parameters. Most existing robotic architectures do not separate these two aspects of behavior control. By separating the actionability from the attentionality and allowing actionability to have a continuous range, EICA enables a form of attention focusing that is usually unavailable to behavioral systems. This separation allows the robot to select the active processes depending on the general context (by setting the actionability value) while still being able to assign the computation power according to the exact environmental and internal condition (by setting the attentionality value). The fact that the actionability is variable allows the system to use it to change the possible influence of various processes (through the operators of the effect channels) based on the current situation.
6. The central action integrator acts as a low pass filter that reduces the effects of noise by combining the actions sent by various intentions and accumulating them for a period that is controlled by the *attentionality* of the action integrator itself. This provides a simple means of noise rejection. A more subtle advantage of the proposed system in relation to goal-directed behavior is the distributive nature of the behavior integration layer which allows, for well designed robots, the emergence of complex external behavior from simple internal processes. The other option to achieve this complex external behavior was to map this complexity directly to the intentions themselves which would have complicated the design too much.

4.4 Limitations

Although the previous subsection has shown that the proposed architecture can theoretically achieve the six requirement of section 2 the system still has some limitations. One of the major problems with the proposed system is that it is sometimes difficult design the behavior level integration layer because of its asynchronous distributed nature and it is also difficult to learn the parameters needed to control the timing of the operation of various processes in it. Although careful division of the task into a set of intentions and task-specific integration processes can alleviate this problem, a general guideline to this process is needed to simplify the design of EICA robots. This limitation can be overcame by restricting the number of processes active at any moment to only one process but this will lead to over-complication in the design of the intentions. A better solution to this problem is a direction for future research.

5 EXPERIMENT

The ability to use human-like nonverbal listening behavior is an advantage for humanoid robots that coexist with humans in the same social space and is complex enough to test some of the proposed system's features. (Kanda et al., 2007) implemented a robot that tries to use natural human like body language while listening to a human giving it road directions based on the situated modules architecture. The road guidance task is simplified by the fact that there are no other objects of interest in the scene except the human. In this work we try to build a general listener robot that can generate natural nonverbal behavior in an explanation scenario involving unknown number of objects that can also be moving. As a minimal design, only the head of the robot was controlled during this experiment. This decision was based on the hypothesis accepted by many researchers in the nonverbal human interaction community that gaze direction is one of the most important nonverbal behaviors involved in realizing natural listening in human-human close encounters (Argyle, 2001). This example is intended as a guide for designing systems that can utilize the proposed integration strategy and a proof of its applicability for HRI.

5.1 Procedure

The evaluation data was collected as follows:

1. Six different explanation scenarios were collected in which a person is explaining the procedure of operating a hypothetical machine that involves pressing three different buttons, rotating a knob, and noticing results in an LCD screen in front of a Robovie II robot while pretending that the robot is listening to the explanation. The data was collected using the PhaseSpace Motion Digitizer system (PhaseSpace, 2007) by utilizing 18 LED markers attached to various parts of the speaker's body. The data was logged 460 times per second.

2. The logged data were used as the input to a robot simulator that implemented the proposed system. The behavior of the robot's head was compared with known human-human behavior in terms of mutual gaze and gaze toward instructor.
3. For every scenario 20 new synthetic scenarios were generated by utilizing 20 different levels of noise. The error level is defined as the percentage of the mean value of the noise term to the mean of the raw signal. The behavior of the simulator was analyzed for every one of the resulting 120 scenarios and compared to the original performance.

5.2 Design

Four reactive intentions were designed that encapsulate the possible interaction actions that the robot can generate, namely, looking around , following the human face , following the salient object in the environment, and looking at the same place the human is looking at. Each one of those intention is implemented as a simple state machine (Mohammad et al., 2007). The sufficiency of those intentions was based on the fact that in the current scenario the robot simply has no other place to look, and the necessity was confirmed empirically by the fact that the three behavioral processes needed to adjust the intentionality of all of these intentions.

The analysis of natural listening requirements showed the need of three behavioral processes. Two processes to generate an approach-escape mechanism controlling looking toward the human operator which is inspired by the *Approach-Avoidance* mechanism suggested in (Argyle, 2001) in managing spatial distance in natural human-human situations. These processes were named Look-At-Human, and Be-Polite. A third process was needed to control the realization of the mutual attention behavior. This process was called Mutual-Intention. The details refer to (Mohammad et al., 2007). A brief description of them is given here:

1. *Look-At-Human*: This process is responsible of generating an attractive virtual force that pulls the robot's head to the location of the human face.
2. *Be-Polite*: This process works against the *Look-At-Human* process generating a repulsive virtual force that pulls the robot's head away from the location of the human face depending on the period of attending to the human.
3. *Mutual-Attention*: This process tries to pull the robot's head toward direction to which the human is looking.

Table 2: Comparison between the Simulated and Natural Behavior.

Item	Statistic	Simulation	H-H value
Mutual Gaze	Mean Std.Dev.	31.5% 1.94%	30% –
Gaze Toward Instructor	Mean Std.Dev.	77.87% 3.04%	75% –
Mutual Attention	Mean Std.Dev.	53.12% 4.66%	unknown –

Five perception processes were needed to implement the aforementioned behavioral processes and intentions. The details can be found in (Mohammad et al., 2007).

5.3 Results

Some of the results of numerical simulations of the listening behavior of the robot are given in Table 2. The table shows the average time of performing two basic interactive behaviors obtained from the simulated robot in comparison to the known average values measured in human-human interaction situations. The average times in the human-human case are reported from (Argyle, 2001). As the table shows the behavior of the robot is similar to the known average behavior in the human-human case for both mutual gaze and gaze toward instructor behaviors and the standard deviation in both cases is less than 7% of the mean value which predicts robust operation in real world situations.

Although the average time of showing the three evaluation behaviors are similar to the human-human values as shown in Table 2, this similarity is not enough for completely judging human like natural behavior and the dynamical aspects of the interaction must be taken into account before drawing final conclusions. This method of analysis, although limited, was selected because of two reasons. First the behavior of the speaker in this experiment is not totally natural because the robot did not respond at real time, which affects the dynamics of the interaction but it was hypothesized that the effect on the averages used for evaluation is much less. Second there is no available data about the detailed dynamics of the behavior of the listener and the speaker in the human-human case during explanation scenarios. In near future a wide scale human-human experiment will be conducted by the authors to collect such data for more accurate evaluation.

Fig. 2 shows the effect of increasing the error level on the percentage of time mutual gaze and gaze to-

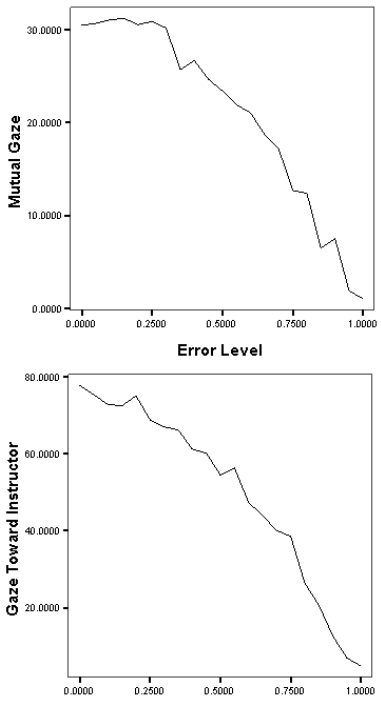


Figure 2: Effect on the error level on the behavior of the robot.

ward instructor behaviors were recognized in the simulation. As expected the amount of time spent on these interactive behaviors decreases with increased error level although this decrease is not linear but can be well approximated with a quadratic function as regression analysis revealed. This means that the performance degrades *gracefully* with the increased noise level even for signal correlated noise.

5.4 Discussion

One of the main purposes of having robotic architectures is to make it easier for the programmer to divide the required task into smaller computational components that can be implemented directly. The proposed action integration mechanism helps in achieving a *natural* division of the problem by the following simple procedure. First the task is analyzed to find the basic competencies that the robot must possess in order to achieve this task. Those competencies are not complex behaviors like *attend-to-human* but finer behaviors like *look-right*, *follow-face*, etc. Those competencies are then mapped to the *intentions* of the system. Each one of these intentions should be carefully engineered and tested before adding any more components to the system

The next step in the design process is to design the behavior level integration part of the action integration. To do that, the task is analyzed to find the underlying processes that control the required behavior. Those processes are then implemented. The most difficult part of the whole process is to find the correct parameters of those processes to achieve the required external behavior. Currently this parameter choice is done using trial-and-error but it will be more effective to use machine learning techniques to learn those parameters from the interactions either offline or online. The current architecture supports run-time adaptation of those parameters, and this feature will be exploited in the future to implement learning of the behavioral integration layer. Those behavioral steps are added incrementally and the relative timing between them is adjusted according to the required behavior.

This simple design procedure is made possible because of the separation between the basic behavioral components (intentions) and the behavior level integration layer (processes).

It is informative to compare this procedure with the procedure suggested in (Ishiguro et al., 1999) for the situated modules architecture. Every situated module should have a list of *preconditions* that is always checked by the module controller which chooses that module that is most suitable to the situation. The problem with this arrangement for HRI applications is that the evaluation of the preconditions can be very time consuming or even very difficult to decide in the first place. Let's consider the situated module *look-at-human* that should be activated during the interaction enough to make the speaker feel comfortable but not too much. How can the designer find all the rule to select all the occasions in which the behavior is to be invoked? and how can this list be updated for reuse in other applications? The main problem is that it is too difficult to find the preconditions for a behavior as simple as *look-at-human* and the only solutions are either to complicate the behaviors used (increase the granularity) or to complicate the module controller (may be by using deliberation). In the proposed system this problem does not exist because the behavior itself (the intention) is coded without any need to think about its preconditions. The behavior level integration processes are then designed based on the global view of the task and not the requirements of each intention which means that this set of preconditions need not be built at any point in the design process. This allows intentions to be thinner than the situated modules without the need of a higher deliberative layer.

A widely accepted definition of intention is: "*a choice with commitment*" (Cohen and Levesque,

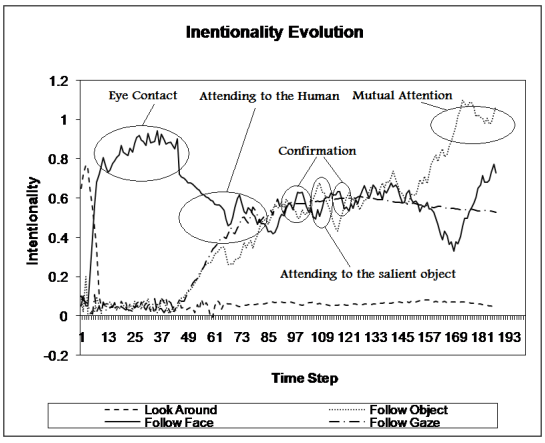


Figure 3: The evolution of intentionality of the four basic intention used in the listener robot.

1990). This definition emphasizes that for a behavior to be perceived as intentional it must possess a form of behavioral *inertia* that makes the robot stick with its behavior until its goal is achieved or the situational changes result on the adoption of a new goal. Careful design of the behavior level integration processes can achieve this goal in the proposed architecture. For example Fig. 3 shows the evolution of intentionality of the aforementioned four basic reactive intentions in one case annotated with the external behavior as perceived by the interacting human. As shown in the figure the effects of the fluctuations of the input signals ,although mapped to the intentionality of various intentions, do not affect the final behavior directly because of the existence of the central action level integrator that averages those fluctuations and provides this needed *behavioral inertia* to make the final behavior look intentional for the human.

6 CONCLUSIONS

This paper analyzed the required properties for action integration mechanisms suitable for interactive robots, and presented the design of a new two layers hybrid action integration mechanism that utilizes both distributive integration in the behavior integration layer and a central action level integrator that can achieve the aforementioned requirement. The paper also presented a simple example of a listener robot implemented based on the proposed mechanism.

REFERENCES

Argyle, M. (2001). *Bodily Communication*. Routledge; New Ed edition.

Arkin, R. C. (1993). *Biological neural networks in invertebrate neuroethology and robotics*, chapter Modeling neural function at the schema level: Implications and results for robotic control, page 17. Academic Press.

Breazeal, C. Kidd, C. T. A. H. G. B. M. (2005). Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)*, pages 708–713.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.

Cohen, P. and Levesque, H. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42:213–261.

Ishiguro, H., Kanda, T., Kimoto, K., and Ishida, T. (1999). A robot architecture based on situated modules. In *IEEE/RSJ Conference on Intelligent Robots and Systems 1999 (IROS 1999)*, volume 3, pages 1617–1624. IEEE.

Kanda, T., Kamasima, M., Imai, M., Ono, T., Sakamoto, D., Ishiguro, H., and Anzai, Y. (2007). A humanoid robot that pretends to listen to route guidance from a human. *Autonomous Robots*, 22(1):87–100.

Maes, P. (1989). How to do the right thing. *Connection Science*, 1(3):291–323.

Mohammad, Y. F. O. and Nishida, T. (2007). A new, hri inspired, view of intention. In *AAAI-07 Workshop on Human Implications of Human-Robot Interactions*, pages 21–27.

Mohammad, Y. F. O., Ohya, T., Hiramatsu, T., Sumi, Y., and Nishida, T. (2007). Embodiment of knowledge into the interaction and physical domains using robots. In *International Conference on Control, Automation and Systems*, pages 737–744.

Nicolescu, M. N. and Matarić, M. J. (2002). A hierarchical architecture for behavior-based robots. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 227–233, New York, NY, USA. ACM.

Perez, M. C. (2003). *A Proposal of a Behavior-based Control Architecture with Reinforcement Learning for an Autonomous Underwater Robot*. PhD thesis, University of Girona.

PhaseSpace (2007). http://www.phasespace.com/.

S Karim, L Sonenberg, A. T. (2006). A hybrid architecture combining reactive plan execution and reactive learning. In *9th Biennial Pacific Rim International Conference on Artificial Intelligence (PRICAI)*.

Steels, L. (1993). *The artifcial route to artifcial intelligence. Building situated embodied agents*, chapter Building agents with autonomous behaviour systems. Lawrence Erlbaum Associates, New Haven.

ROBOT LOCALIZATION BASED ON VISUAL LANDMARKS

Hala Mousher Ebied, Ulf Witkowski, Ulrich Rückert
System and Circuit Technology, Institute of Heinz Nixdorf, Paderborn Universit
Fürstenallee 11, D-33102 Paderborn, Germany
hala@hni.upb.de, witkowski@hni.upb.de, rueckert@hni.upb.de

Mohamed Saied Abdel-Wahab
Department of Scientific Computing, Faculty of Computer and Information Sciences
Ain Shames University, Cairo, Egypt
mswahab@gmail.com

Keywords: Position system, Vision-based localization, Triangulation, Minirobot, Visual landmarks.

Abstract: In this paper, we will consider the localization problem of the autonomous minirobot Khepera II in a known environment. Mobile robots must be able to determine their own position to operate successfully in any environments. Our system combines odometry and a 2-D vision sensor to determine the position of the robot based on a new triangulation algorithm. The new system uses different colored cylinder landmarks which are positioned at the corners of the environment. The main aim is to analyze the accuracy and the robustness in case of noisy data and to obtain an accurate method to estimate the robot's position.

1 INTRODUCTION

Vision-based localization is the process to recognize the landmarks reliably and to calculate the robot's position (Borenstein et al., 1996). Researchers have developed a variety of systems sensors, and techniques for mobile robot positioning. (Borenstein et al., 1997) define seven categories for positioning system: Odometry, Inertial Navigation, Magnetic Compasses, Active Beacons (Trilateration method and Triangulation method), Global Positioning Systems, Landmark Navigation, and Model Matching. Some experimental systems that work on localization are using mobile robots equipped with sensors that provide range and bearing measurements to beacons (Witkowski and Rückert, 2002) and some other work with vision sensors (Chinapirom et al, 2004).

Because of the lack of a single good method, developers of mobile robots usually combine two or more methods (Borenstein et al., 1997). To predict the robot's location, some systems combine odometric measurements, landmark matching, and triangulation with observations of the environment from a camera sensor (DeSouza and C.Kak, 2002; Yuen et al., 2005). (Martinelli et al, 2003) combine the odometric measurements, uses encoder readings as inputs, and the readings from a laser range finder as observations to localize the robot.

Many solutions to the localization problem included geometric calculations which do not consider uncertainty and statistical solutions (Kose et al., 2006). Triangulation is a widely used to the localization problem. It uses geometry data to compute the robot's position in indoor environment (Shoval et al., 1998).

Triangulation is based on the measurement of the bearings of the robot relatively to the landmarks placed in known positions. Three landmarks are required at least for solving the triangulation. The robot's position estimated by the triangulation method is based on find the intersection of the two circles which passes through the robot and two landmarks, show figure 1.

Cohen and Koss, 1993 present four methods for triangulation from three fixed landmarks in known locations in an environment: iterative search (IS), geometric triangulation (GT), iterative Newton Raphson (NR) and geometric circle intersection (GCI). NR and GCI methods are found to be the most efficient ones. IS and GT are not practical.

The geometric triangulation method is based on the law of sins and works consistently only when the robot is within the triangle formed by the three

landmarks (Casanova et al., 2002, Esteves et al., 2003). The geometric circle intersection method is widely used in literature. It fails when the three landmarks and the robot lie on a same circle. Betke and Gurvis, 1997 are using more than three beacons.

The landmark is written as a complex number in the robot-centered coordinate system. The processing time depends linearly on the number of the landmarks. Casanova et al (Casanova et al., 2002; Casanova et al., 2005) address the localization problem of moving objects using laser and radiofrequency technology through a geometric circle triangulation algorithm. They show that the localization error varies depending on the angles between the bacons.

In this paper, a system for global positioning of a mobile robot is presented. Our system combines odometry and 2-D vision data to determine the position of the robot by a new Alternative Triangulation Algorithm (ATA). ATA is obtained from a set of equations from the triangles between the robot and the landmarks. We present simulation results for noisy input data to estimate the robot's position with respect to the environment.

In the next section, the robot platform is described. Section 3 describes the triangulation method based on the law of cosine. Section 4 contains the description of the new Alternative Triangulation Algorithm. Results are given in section 5.

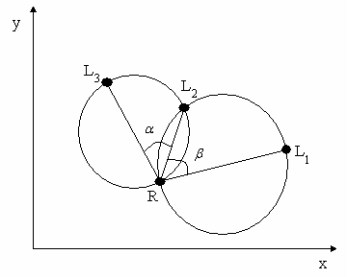


Figure 1: The robot's position is the intersection of two circles. Each one of the two circle passes through the robot and two landmarks.

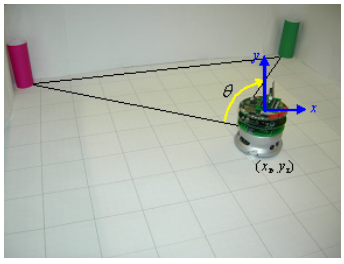


Figure 2: Khepera with the camera module in the test environment.

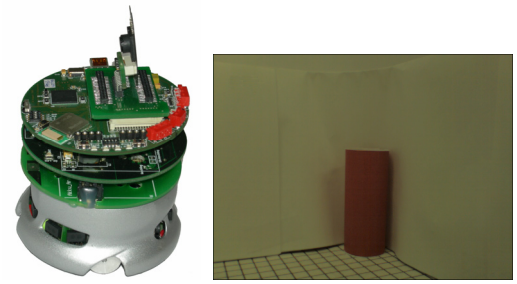


Figure 3: Khepera minirobot II equipped with FPGA module and 2D color CMOS camera module, and a typical image with 640 x 480 pixels from camera module.

2 ROBOT PLATFORM

The main idea is the detection of landmarks of different color by using a CMOS color sensor. For this, the robot gets pictures of the environment in different orientations and/or from different positions. An image processing algorithm will be used to extract the centre of each landmark; details are given in (Ebied et al., 2007). Also the different angles between different landmarks as viewed from the mobile robot will be calculated using the odometry method as show in figure 2. Then we develop a new alternative triangulation algorithm to calculate the robot's position based on parameter set (angles) and the knowledge of the positions of the landmarks.

The positioning system has been implemented on the minirobot Khepera (K-Team, 2002) that uses an additional camera module, as show in figure 2. The camera is a 2D color CMOS camera from Transchip, model TC5740MB24B. To control the camera we use an FPGA module that is equipped with USB 2.0 port. Via the USB port the programmer is able to see on a computer screen what the robot is capturing. The received images have a resolution of 640 x 480 pixels in 8 bit RGB color, see figure 3.

3 THE TRIANGULATION ALGORITHM

This section shows how to apply well-known triangulation algorithm (Betke and Gurvis, 1997) to the robot localization problem. If the positions of the landmarks are known and also the angles between the landmarks relative to the robot's position are known, then we can use the law of cosine to calculate the distance between the robot and the three landmarks. Let (x_i, y_i), i=1…3 be the positions of the landmarks L_1 … L_3 in the Cartesian

coordinate of the environment. From the triangle between the robot and the landmarks L_1 and L_2 , we get the following equation by applying the law of cosine

$$\left(x_1-x_2\right)^2=d_1^2+d_2^2-2d_1d_2\cos\left(\theta_1*\frac{\pi}{180}\right) \quad (1)$$

where d_i, i=1…3 are the unknown distance between the landmarks and the robot's position, and θ_1 is the angle between landmarks L_1 and L_2 relative to the robot' position, as shows in figure 4.

With three landmarks, there are three pairs of landmarks (i.e. landmarks 1 and 2, 2 and 3, and 1 and 3). So we can get system of equations to determine the distance d_i between the robot's position and the landmarks L_i, by applying the law of cosine to all pairs of landmarks

$$\left(x_1-x_3\right)^2=d_1^2+d_3^2-2d_1d_3\cos\left(\theta_2*\frac{\pi}{180}\right) \quad (2)$$

$$\left(x_2-x_3\right)^2=d_2^2+d_3^2-2d_2d_3\cos\left(\theta_{12}*\frac{\pi}{180}\right) \quad (3)$$

Where θ_2 is the angle between landmarks L_2 and L_3 relative to the robot' position, as shown in figure 4. The system of nonlinear equations (1), (2), and (3) can be solved using a least square method. Once the distances d_i are found, the robot's position (x_r, y_r) can be estimated from solving the following equations:

$$d_1^2=\left(x_1-x_r\right)^2+\left(y_1-y_r\right)^2 \quad (4)$$

$$d_2^2=\left(x_2-x_r\right)^2+\left(y_2-y_r\right)^2 \quad (5)$$

$$d_3^2=\left(x_3-x_r\right)^2+\left(y_3-y_r\right)^2 \quad (6)$$

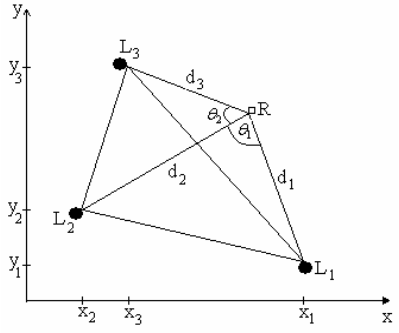


Figure 4: Estimated robot's position R using law of cosine.

4 AN ALTERNATIVE TRIANGULATION ALGORITHM

In the alternative triangulation algorithm, also at least three landmarks are required to estimate the robot's position. We can use the triangles between the robot and the three landmarks to estimate the robot's position. From the triangle between the robot and the landmarks L_1 and L_2, we get the following equations

$$\tan\left(\theta_{1a}*\frac{\pi}{180}\right)=\frac{x_1-x_r}{y_r-y_1} \quad (7)$$

$$\tan\left(\theta_{1b}*\frac{\pi}{180}\right)=\frac{x_r-x_2}{y_r-y_2} \quad (8)$$

The angles θ_{1a} and θ_{1b} can be computed from the measured angle θ_1, as shows in figure 5.

$$\theta_{1a}+\theta_{1b}=\theta_1 \quad (9)$$

We can apply the same equations to another pair of landmarks and get a system of six equations in six unknown variables which can be solved to give us an estimate for the robot's position (x_r, y_r). Applying the same equations on the triangle between the robot and the landmarks L_2 and L_3, we get the following equations

$$\tan\left(\theta_{2a}*\frac{\pi}{180}\right)=\frac{y_r-y_2}{x_r-x_2} \quad (10)$$

$$\tan\left(\theta_{2b}*\frac{\pi}{180}\right)=\frac{y_3-y_r}{x_r-x_3} \quad (11)$$

$$\theta_{2a}+\theta_{2b}=\theta_2 \quad (12)$$

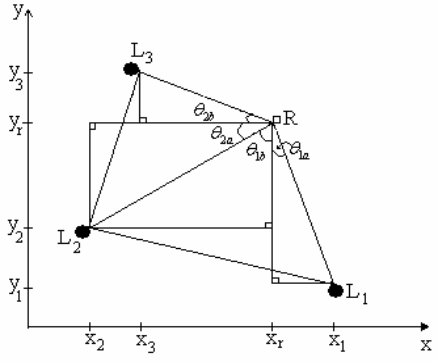


Figure 5: An alternative triangulation algorithm to estimated robot's position R using three Landmarks.

5 RESULTS AND DISCUSSION

Four distinctly coloured landmarks are placed at known positions at the edges of the environment. For a robot's environment with a size of 62 x 74 cm^2, the floor of the environment will be divided into squares (cells) that are used to determine the position of the robot. We have used a grid with 11 by 13 lines in each direction which results in a data set of 143 robot's position. The robot's position error is considering as the Euclidean distance between the estimated robot position and the actual robot position. We averaged the robot's position error over 143 places in the environment.

In the first experiment, the alternative triangulation algorithm has been used to estimate robot's position in the experimental environment. The experiments deal with different combinations of three landmarks. In Figure 6, the position error has been plotted as a function of the robot's position. The average of the position error is 0.35cm which is limited between 0.02cm and 3.5cm. Maximum error is obtained when the two angles between three landmarks relative to the robot's position is less than 90°. It is easily to overcome the error in robot's position by using all the visible landmarks.

Using several landmarks yield a more robust robot's position estimate. If we have n landmarks, there are many different combinations of the three landmarks that can be used to compute the position of the robot. We can obtain the minimum position error by using two different ways. First, we can calculate the average from the estimated robot's position from all different groups of three landmarks. Or by using a suitable algorithm to select the best 3 landmarks with at least one angle greater than 90°.

The position error average from all different groups of landmarks is 0.253cm which is limited between 0.022cm and 0.935cm. The position error, based on selecting the best 3 landmarks with at least one angle greater than 90°, is 0.242cm which is limited between 0.021cm and 0.767cm. So by using a suitable algorithm to select the best 3 landmarks with at least one angle greater than 90°, we can obtain the minimum position error, as show in figure 7.

The second experiment deals with the triangulation algorithm based on the law of cosine. Solving system of nonlinear equations using the least square method requires making a starting guess of the robot's position. Table 1 shows the results of the experiments in which the starting guess of the solution takes across the environment diagonal. Our results show the minimum, maximum, mean, and standard deviation of the position error. We begin to receive a minimum of the position error close to the environment center. The mean of the position error is 0.0035cm when the center coordinate of the environment is used as the starting guess.

The last experiment, we consider the robot's position problem for noisy angle data that are measure using odometry method. Errors in odometry are caused by the intersection between the wheels and the terrain, for example slippage, cracks, debris of solid material, etc. (Borenstein et al., 1997). We add a random noisy angle $\Delta\theta$ of degree to each actual angle. Table 2 shows the position error using the alternative triangulation algorithm with different combinations of three landmarks, and using the random generated angles $\Delta\theta$ on the specified interval [-3°, 3°] which are added to the actual angles.

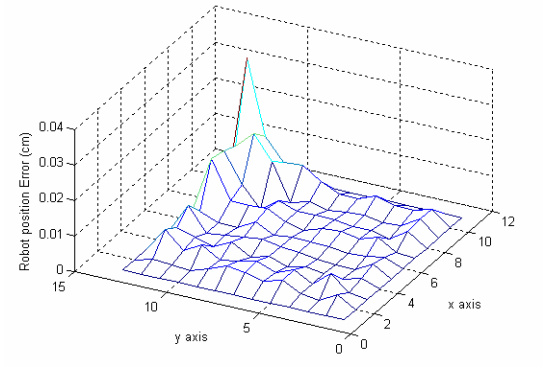


Figure 6: Estimated robot's position using an ATA. Three landmarks have been placed at the right and left bottom corner and at the left top corner of the environment. The maximum error happened when the robot was been almost on the same line with one of the landmarks parallel to the x-axis.

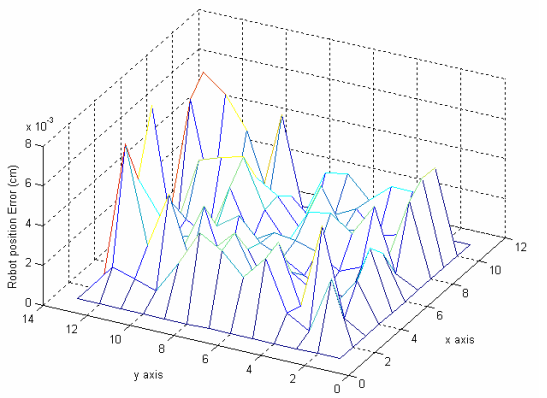


Figure 7: By using a suitable algorithm to select the best 3 landmarks with at least one angle greater than 90°.

Table 1: Minimum, maximum, mean, and standard deviation values (in cm) for the robot's position error at different starting guess of the solution.

	(10,10,10)cm	(15,15,15)cm	(20,20,20)cm
min	0,0002	0,0002	0,0002
max	66,0609	66,0609	0,0351
mean	8,0755	1,8593	0,0035
std	18,1959	10,5717	0,0046

Table 2: Minimum, maximum, mean, and standard deviation values (in cm) for the robot's position error of different groups of landmarks. The results from selecting the suitable three landmarks with at least one angle greater than 90° are given in column Select.

	Landmark groups					
	1, 2, 3	1, 2, 4	2, 3, 4	1, 3, 4	average	Select
min	0,222	0,102	0,097	0,047	0,106	0,20
max	8,631	10,12	11,01	21,24	6,490	4,07
mean	1,924	1,944	2,136	2,154	1,403	1,50
Std	1,511	1,602	2,065	2,520	0,897	0,88

ACKNOWLEDGEMENTS

This work is a result of common research between the Department of System and Circuit Technology, Heinz Nixdorf Institute, University of Paderborn and Scientific Computing Department, Faculty of Computer and Information Sciences, Ain Shames University. It is also supported from the Arab Republic of Egypt cultural department and study mission.

REFERENCES

Betke, M., and Gurvits, L., 1997. Mobile Robot Localization Using Landmarks. *IEEE Transactions On Robotics And Automation*, Vol. 13, No. 2.

Borenstein, J., Everett, H. R. and Feng, L., 1996. Where am I? Sensors and Methods for Mobile Robot Positioning, *Technical Report, The University of Michigan*.

Borenstein J., Everett, H. R., Feng, L., and Wehe, D., 1997. Mobile Robot Positioning: Sensors and Techniques, *Journal of Robotic Systems*, vol. 14, , pp. 231-249.

Casanova, E. Z., Quijada, S. D., García-Bermejo, J. G., and González, J. R. P., 2002. A new beacon based system for the localisation of moving objects. *9th IEEE International Conference on Mechatronics and Machine Vision in Practice*, pp. 59-64, Chiang Mai, Tailandia.

Casanova, E. Z., Quijada, S. D., García-Bermejo, J. G., and González, J. R. P., 2005. Microcontroller based System for 2D Localisation , *Mechatronics,* Vol. 15, Issue 9, pp. 1109-1126.

Chinapirom, T., Kaulmann, T., Witkowski, U., and Rückert, U., 2004. Visual Object Recognition by 2d-Color Camera and On-Board Information Processing For Minirobots. *FIRA Robot World Congress Busan*, South Korea.

Cohen, C. and Koss, F., 1993. A Comprehensive Study of Three Object Triangulation. *SPIE Conference on Mobile Robots, Boston, MA*, pp. 95-106.

DeSouza, G.N., and C.Kak, A., 2002. Vision for Mobile Robot Navigation:ASurvey. *IEEE Transactions on pattern analysis and Machine intelligence*, Vol. 24, No.2.

Ebied, H. M., Witkowski, U., Rückert, U., and Abdel-Wahab, M. S., 2007. Robot Localization System Based on 2D-Color Vision Sensor, *4th International Symposium on Autonomous Minirobots for Research and Edutainment*, pp. 141-150, Buenos Aires, Argentina.

Esteves, J.S., Carvalho, A., Couto, C., 2003. Generalized Geometric Triangulation Algorithm for Mobile Robot Absolute Self-Localization. *IEEE International Symposium on Industrial Electronics*, pp. 346 – 351.

Kose, H., Celik, B., and Levent Akın, H., 2006. Comparison of Localization Methods for a Robot Soccer Team, *International Journal on Advanced Robotic Systems*, Vol. 3, No. 4, pp. 295-302.

K-Team, S.A. 2002. Khepera 2 User Manual. Ver 1.1, Lasanne, Switzerland.

Martinelli, A., Tomatis, N., Tapus, A., and Siegwart, R., 2003. Simultaneous Localization and Odometry Calibration for Mobile Robot. *IEEE International Conference on Intelligent Robots and Systems*, pp. 1499-1504, Las Vegas, Nevada.

Shoval, S., Mishan, A., and Dayan, J., 1998. Odometry and Triangulation Data Fusion for Mobile-Robots Environment Recognition, *International Journal on Control Engineering Practice,* Vol. 6, Issue 11, pp. 1383-1388.

Witkowski, U., and Rückert, U., 2002. Positioning System for the Minirobot Khepera based on Self-organizing Feature Maps. *FIRA Robot World Congress*, pp. 463-468, COEX, Seoul, Korea.

Wolf, J., Burgard, W. and Burkhardt, H., 2005. Robust Vision-Based Localization by Combining an Image-Retrieval System with Monte Carlo Localization. *IEEE Transactions on Robotics*, Vol. 21, No. 2, pp. 208-216.

Yuen, D. C. K., and MacDonald, B. A., 2005. Vision-Based Localization Algorithm Based on Landmark Matching, Triangulation, Reconstruction, and Comparison, *IEEE Transactions On Robotics*, Vol. 21, No. 2, pp. 217-226.

COMPARISON OF DIFFERENT INFORMATION FLOW ARCHITECTURES IN AUTOMATED DATA COLLECTION SYSTEMS

Jussi Nummela, Petri Oksa, Leena Ukkonen, Lauri Sydänheimo and Markku Kivikoski
Tampere University of Technology, Electronics Institute, Rauma Research Unit, Kalliokatu 2, 26100 Rauma, Finland
jussi.nummela@tut.fi

Keywords: Automated data collecting, information flow architecture, traffic load.

Abstract: Automated Data Collecting System (ADCS) is a common name for automatic systems that collect data of any kind. These systems are becoming more and more common in several industries and play an important part in many of today's and future applications. Information flow architecture is an important issue, when employing an ADCS. This paper presents different kinds of architecture models and their typical characteristics, concentrating on traffic load issues in different parts of the system. The results presented in this paper, give a basis for more accurate specifying and designing of the architecture model for each automated data collecting application in question.

1 INTRODUCTION

Information flow architectures play an important role in many present day and especially future applications. As automatization becomes more common in many industries, the problem of information flow architecture must be solved. This actually consists of several "sub-problems": where data is transferred, how it is transferred, where data is stored, who can access the data, how the access is performed, what configuration is needed and which party performs them, how the new participant is added etc. etc. (Jie et. al., 2006)

All the above questions must be answered to make the system optimally suited for an intended application. Every application has its own individual characteristics and therefore a common answer for the information flow architecture cannot be given. All the options have their own pros and cons, and these are discussed in this study. The main focus is however in comparing throughputs and traffic loads in different parts of the Automated Data Collecting Systems (ADCS) and in different models.

ADCS is defined here as including all types of automatic systems that collect any kind of data. Well known examples can be, for example, RFID-systems, supply chain management systems, automatic meter reading (AMR) systems, forest fire surveillance sensor networks or highway speed control systems. The common factor is that systems collect data and in some way make it available for their users. (Bodrozic, Stipanicev, Stula, 2006; Wang et. al., 2005)

An ADCS usually consists of data collection units (DCU) (e.g. RFID readers or water consumption meters), database(s), optional server(s) and network and data links between these components. All of the components have an effect on the nature and behaviour of the system. Therefore, the components must be chosen based on the needs of the application in question. *Video data stream systems* transfer large amounts of data and they require small jitter and high throughput due to their real time operation. *AMR systems* transfer small amounts of data and also the real time demand is very low. *Supply chain management systems* also deal with small data quantities, but they might need very short response times and delays, for example where handling machines are exploiting the data. *EDI systems* (Electronic Data Interchange) do not usually demand real time features, but the transferred data amount might still be high. *Forest fire surveillance sensor networks* put a lot of emphasis on energy efficiency, because of the need for long maintenance intervals (Yu, Wang, Meng, 2005).

Depending on which application the ADCS is designed to be used in, different attributes must be

emphasized. For AMR systems it is not recommended or necessary to roll out a system with effective and high-cost real time operations. In supply chain management it can be considered needless expense to employ a system with very high throughput, instead of concentrating resources on keeping delays low.

The simulations presented in this study present the differences in traffic load in different architectures. These results give a basis for specifying and designing a suitable system for each application. This paper is sectioned as follows: chapter 2 presents three different architecture models and their main characteristics. Chapter 3 contains the simulation descriptions and TCP theory, and traffic load simulation results and discussion are presented in chapter 4. Finally, chapter 5 concludes the study and also takes a look at future work.

2 ARCHITECTURE MODELS

The simulations were done with three different types of architecture model. These were ***centralized***, ***semi-distributed*** and ***distributed*** architectures. The differences between these architectures are:

- The placement and number of data storage(s) e.g. database(s)
- One-way or two-way traffic
- Reaching the database directly or through a dedicated server

All links in the simulation models are marked as A, B, C or D, depending on their characteristics. A, B and C links have 100 Mbps capacity whereas D has 10 Mbps. The delay for every link is 1 µs and BER is 0 %. The same delay and BER are also used for every node in the network as is the buffer size of 50 packets.

2.1 Centralized Architecture Model

The centralized architecture model consists of one server, 12 data collection units (DCUs), 6 switches and 7 routers (GWs) as seen in figure 2.1.

In this simplified model of centralized architecture, all data is stored in the one dedicated server and all users can access the data through that server. This means that the information flow is considered as uni-directional. The links themselves are however bi-directional, as TCP/IP-connections always are, because of the protocol requirements, acknowledge-packets etc.

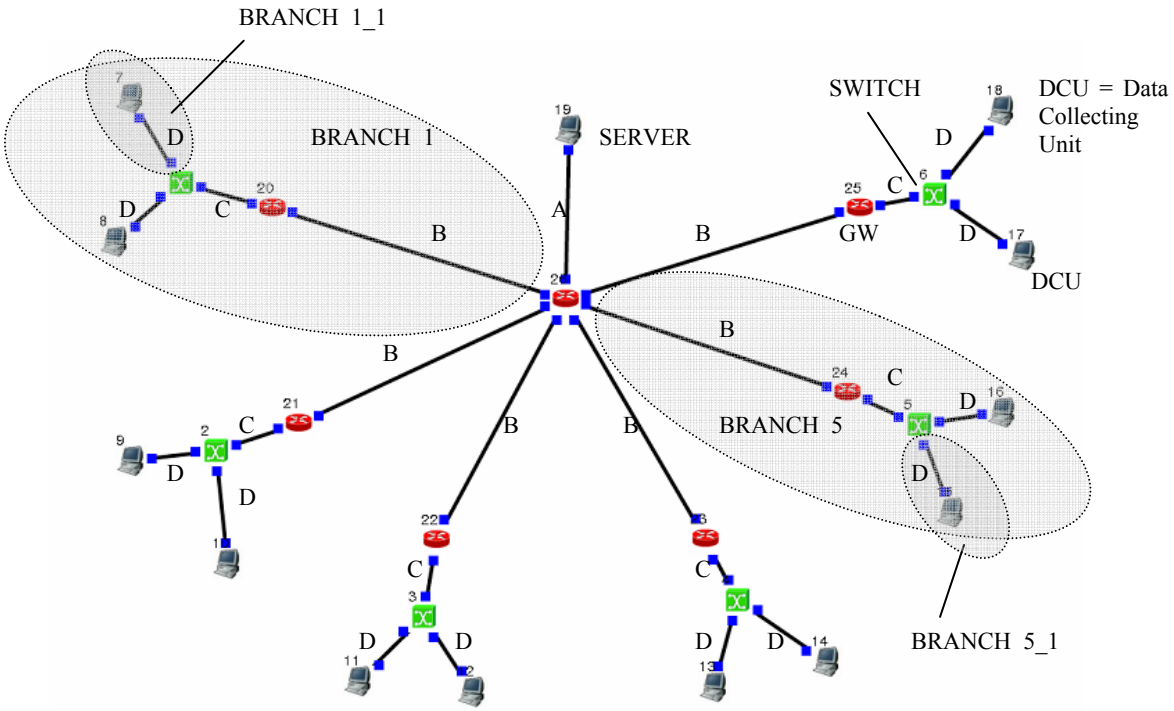


Figure 2.1: Centralized architecture model. Data is stored in the server.

In this model the DCUs send their data direct to server. Other components like switches or GWs only forward the data packets to the following link.

2.2 Semi-Distributed Architecture Model

The semi-distributed architecture model consists of the same components as the centralized model and the used topology is also similar as presented in figure 2.1. In this model the data is however stored in several databases, which are located in every GW. However users will always access the data through a dedicated server, which requests the data in question from each database as needed. Due to these GW-databases, only the on-demand data is transferred beyond its own GW, which decreases the traffic load in the server and A/B links significantly.

In this procedure, the information flow is uni-directional between DCUs and GWs and bi-directional between GWs and the server due to the queries the server uses to request data from GWs.

2.3 Distributed Architecture Model

Unlike the two other architecture models presented above, the distributed architecture model does not have a server. Other components remain the same as in figure 2.2.

In this model users access data, or actually the GW which hosts the database, directly from their own branches (or subnets), not through any dedicated server as in the previous architecture models. The user requests the needed information from a specific GW by sending a query packet(s). The GW then sends the data back to the user.

3 SIMULATIONS

These simulations were performed with the NCTUns Network simulator 3.0 by SimReal Inc, which uses a novel kernel re-entering simulation methodology (Wang et al 2003). The purpose of the simulations was to find the changes in link loads between different architectures.

3.1 Generated Traffic

The generated traffic sequence was similar in all three architectures. The modelled time period was 40 seconds and each DCU produced data for one continuous 10 second period. In the centralized architecture model data was transmitted directly from DCUs to the server, whereas in the semi-distributed and distributed models the data was first stored into GWs, and then requested from there by the server or other DCU.

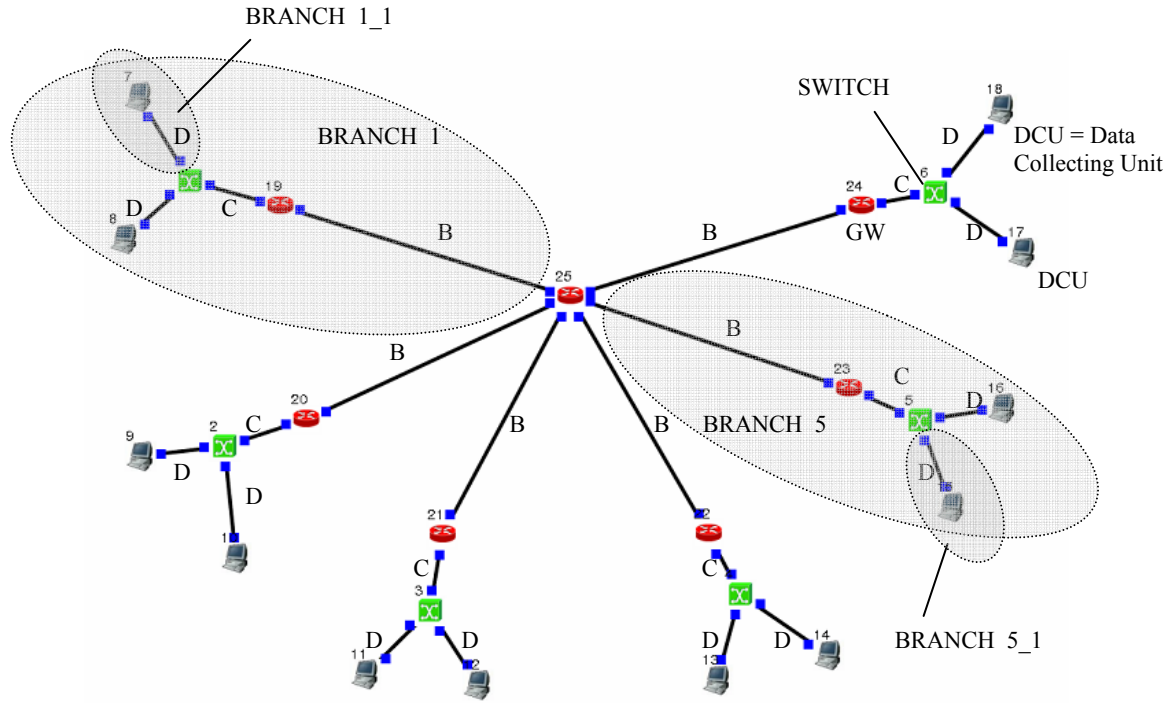


Figure 2.2: Distributed architecture model. Data is stored in the GWs.

These queries lasted 1 second each, as did the answers (e.g. data transfers) for them. Each GW received two of these queries. This means that 10 % of the data each DCU produced was requested by the users and transferred from the databases.

This simulated traffic used basic TCP protocol with 1024 B of payload and all connections used their own individual TCP port numbers (*Transmission Control Protocol*, 1981). Two major characteristics of TCP are powerful mechanisms for error correction and congestion avoidance, which make it suitable for this kind of use, where data is error critical and congestions are highly likely to exist at some point.

The congestion control mechanism of TCP protocol consists of two procedures: *slow start* and *congestion avoidance.* In slow start the extra window for sender, *the congestion window* (*cwnd)*, will be taken into use. The congestion window defines the number of sent segments before an acknowledgement packet is expected to arrive. At the beginning of transmission, the cwnd is 1. When acknowledgement for this first sent packet arrives, the value of cwnd is doubled. This is done after every successful transmission. (Allman et al, 1999)

When the first error occurs, the sender switches to the congestion avoidance procedure to reduce growth speed and achieve network capacity less aggressively. This switching point is called *slow start threshold, sstresh*. The increase in the size of the congestion window, and the number of sent segments before acknowledgements, will continue. The value is increased by one per every *round trip time*. The *round trip time* is a calculated time for a packet to travel from sender to receiver and the receiver's acknowledgement to travel back to the original sender. The increase is now linear, whereas in the slow start phase it was exponential. Eventually, the packet will be lost again. Now the *cwnd* is reset back to 1, and *sstresh* is set to the value of half the current window size. Now the transmission continues with the slow start procedure again, until an error occurs, or the *cwnd* reaches the *sstresh* value, and switches again to congestion avoidance procedure. (Allman et al, 1999)

The transmission continues performing these mechanisms, all the time seeking the current maximum network capacity. It is important to realize, that the *sstresh* does not always fall, it can also rise. If the error in congestion avoidance occurs when the window size is more than twice the *sstresh*, the *sstresh* will increase. The following figure 3.1 presents the changes in the *cwnd* and *sstresh* during slow start and congestion avoidance procedures. Other TCP congestion avoidance algorithms have also been developed, but they are not discussed here, since the focus of this study is in architecture models, not in protocols (Wikipedia, 2007).

TCP protocol was selected for these simulations because it is very commonly used in several kinds of applications and is designed to act well in difficult circumstances. Another protocol option considered was User Datagram Protocol (UDP), which is "lighter" and a connectionless protocol. UDP does not have error correction or congestion avoidance procedures, but because of its low overhead features, it would suit low-power consumption systems well. However systems demanding very low power consumption usually have their own application specific and customized protocols, such as the Kilavi protocol used in building automation (Soini et. al., 2006).

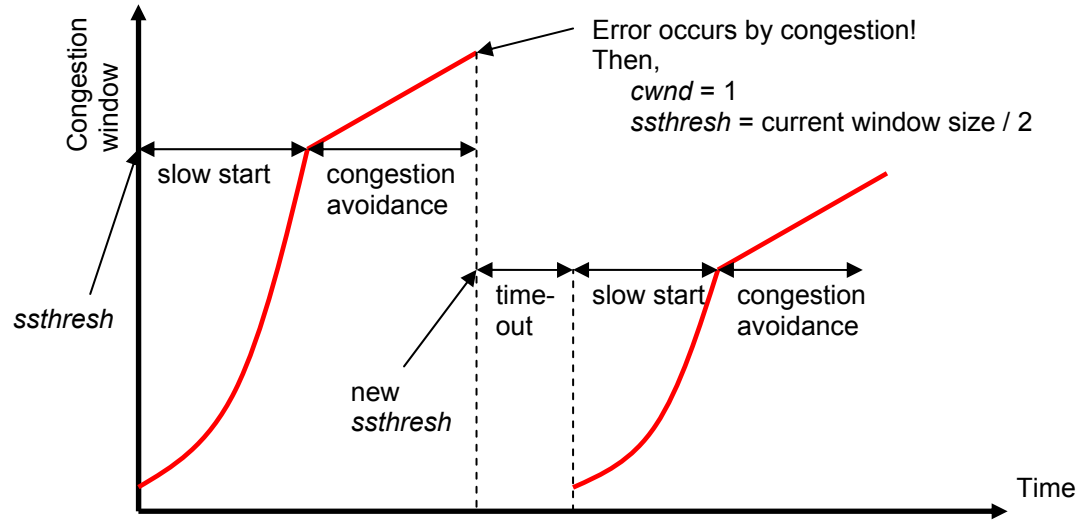


Figure 3.1: The use of congestion window and slow start threshold in TCP transmission (Allman et al, 1999).

4 RESULTS AND DISCUSSION

Throughputs in A- and B-links can be seen from the following graphs. The figure 4.1 shows server link A traffic load in centralized and semi-distributed architectures.

As can be seen, the throughput is substantially lower in the semi-distributed architecture than in the centralized model. This also leads to much lower load on and requirements for the dedicated server.

The traffic in link B is shown in figure 4.2. The picture presents the corresponding graphs from three different architectures. The presented load is measured from branch 5 (as in the graphs in figures 4.3 and 4.4).

As can now be seen, the traffic load in the centralized model is much higher and more continuous than in the other two models. The distributed model has more "spikes" than the semi-distributed, due to data queries, which come directly from DCUs and not from the dedicated server. These queries also produce traffic for link B.

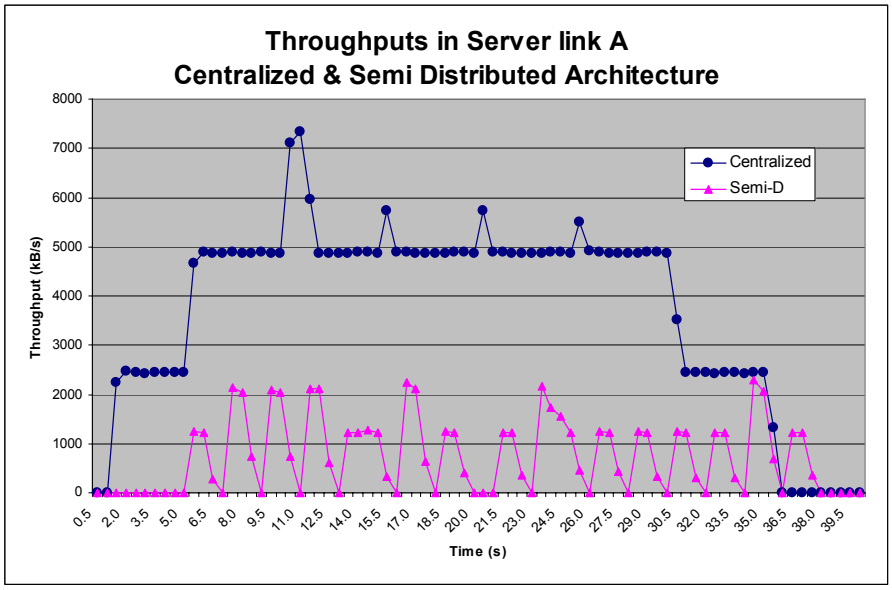


Figure 4.1: Link A throughputs in centralized and semi-distributed architecture models.

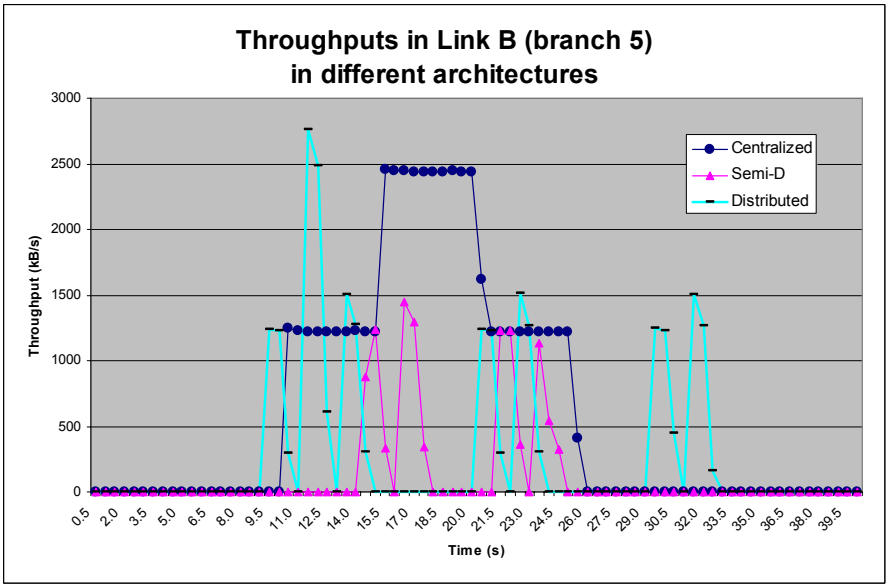


Figure 4.2: Link B throughputs in all three architecture models.

Examining link C it can be seen that data collecting traffic from DCUs is similar in all architecture models. This is presented in figure 4.3.

Graphs indicate that the only difference appears in the distributed model, where data queries also produce load in the link. These queries can be seen as an extra "double spike". In all the other situations it does not matter which architecture model is used when considering traffic load in link C.

These characteristics can also be seen when examining the traffic load of link D, as can be seen in figure 4.4.

When examining the centralized architecture model, a few typical characteristics can be discovered. First of all the traffic load in all links is very high and also continuous. Huge differences compared to the other architectures emerged in links A and B. This leads to the conclusion that server and link capacity must be high for centralized architecture to work well, or alternatively, the amount of collected data must be small. This, added to the fact that administering this kind of one database system is much easier and simpler than systems with several databases due to user

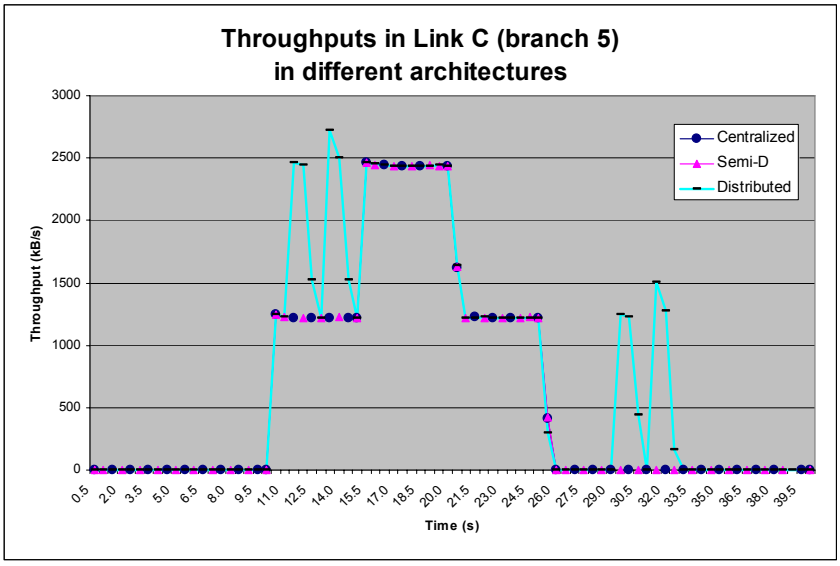


Figure 4.3: Link C throughputs in all three architecture models.

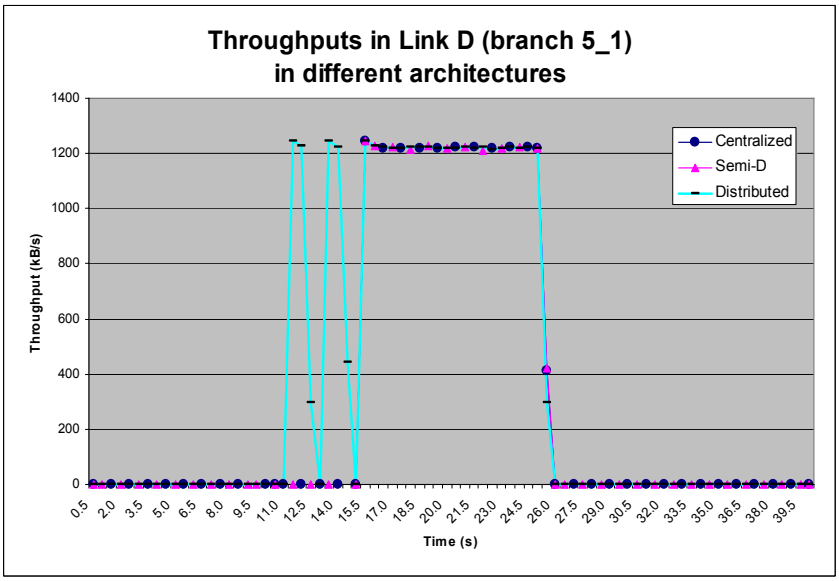


Figure 4.4: Link D throughputs in all three architecture models.

authentication configurations, means that centralized architecture is suitable for ADCS if the system is small and the amount of collected data is also relatively small. Also adding a new DCU is quick and easy because it only communicates with one partner, the server.

The semi-distributed architecture model produced significantly less traffic than the centralized model with the systems main links A and B. This is because only required information is transferred beyond the databases (or gateways in this case). A semi-distributed model is however more complicated to administer, because of distributed resources and databases throughout the system. On the other hand these divided resources do reduce the requirements placed on the equipment, which makes the whole ADCS more reliable and cost-effective. Adding a new DCU or configuring user authentication rules for semi-distributed systems is easy, because all the information is distributed through one dedicated server, which is the only communication partner for the databases. The semi-distributed architecture model is therefore suitable for automated data collecting systems, which have rather large numbers of DCUs and architecture or topology which might change regularly.

The distributed architecture model differs from the other two, because it does not have a server or server link A. The traffic load in link B is quite similar to that of the semi-distributed model, but the distributed model has data queries coming from DCUs too. This slightly increases the throughput, but still the traffic load is much lower than in the centralized model. The distributed model is hard to administer, because of the several databases and communication partners all over the system. Adding a DCU is also complicated, because it needs to communicate and authenticate with several partners. The distributed architecture model is most suitable for an ADCS with a large amount of data and many DCUs, but the architecture is likely to be fixed and new users or DCUs are not expected to be added frequently.

The traffic load in links C or D is very similar in all simulated architectures. Only the distributed model has slightly more traffic here, because data queries come straight from DCUs. Still, the difference is marginal, when most of the traffic load is generated from collected data which is similar in all models.

The security aspects constitute an entity which, despite being essential for each application, is not discussed extensively in this paper. The main security issues for information flow architectures are user authentication and data encryption, which differ more or less for each model. The common factor is that they usually increase system complexity and also traffic load in each model. The need and level of encryption is strongly dependent on the nature of application in use. User authentication, on the other hand, is substantially different for each architecture models, due to different numbers of communication partners, as mentioned earlier. These aspects however require more specific investigation to accurately determine requirements and possibilities for different user authentication methods. (Sikkilä et. al., 2006; Perrig et. al., 2002)

The main characteristics of all three studied models are summarized in the following table 4.1.

5 CONCLUSIONS AND FUTURE WORK

This paper presented three different information flow architecture models for automated data collecting systems, and the main characteristics of each of them. Comparisons of the traffic loads in each part of the systems were also presented, and suitable models for different application types were recommended. These presented results can be used as a basis for designing and specifying an application-specific automated data collecting system.

Table 4.1: The main characteristics of different information flow architecture models for ADCS.

Model	Traffic load	Maintenance	Modifiability	Number of DCUs	Example applications
Centralized	High	Very easy	Very good	Small	Video surveillance
Semi-Distributed	Low	Easy	Good	Large	Water meter reading
Distributed	Low	Hard	Bad	Very large	Supply chain management

As mentioned earlier, every application has its own characteristics and requirements for ADCS. Therefore more application specific studies must be made with each area of intended use in mind. In supply chain management the supply chain must be accurately studied, because even supply chains for different products may have very different needs. In AMR systems the metering environment and needs must be strictly surveyed to achieve an optimal outcome. Therefore this study will be continued with a more accurate definition of the supply chain in the paper reel industry and implementation of an RFID-based ADCS in the paper industry environment. Also the security issues such as user authentication methods will be studied more deeply to determine the procedure options and requirements for adding new parts and partners to ADCS.

REFERENCES

Allman, M., et. al. (1999). IETF Standards Track, RFC 2581, *"TCP Congestion Control"*. <http://www.ietf.org/rfc/rfc2581.txt>. The Internet Engineering Task Force. Accessed 7 Aug. 2007.

Bodrozic, L., Stipanicev, D., Stula, M. (2006). *"Agent based data collecting in forest fire monitoring system"*. International Conference on Software in Telecommunications and Computer Networks, 2006. SoftCOM 2006. Sept. 29 – Oct. 1 2006. IEEE. Page(s): 326 – 330.

Jie, W., Hung, T., Turner, S. J., Cai, W. (2006). *"Architecture Model for Information Service in Large Scale Grid Environments"*. Proceedings of the Sixth IEEE International Symposium, on Cluster Computing and the Grid 2006. CCGRID '06. Volume 1, 16-19 May 2006, IEEE. Pages: 107-114.

Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J.D. (2002). *"SPINS: Security Protocols for Sensor Networks"*. Wireless Networks Journal, Volume 8, Issue 5 (Sept. 2002). Pages: 521-534. Springer Netherlands.

Sikkilä, H., Soini, M., Oksa, P., Sydänheimo, L., Kivikoski, M. (2006). "*KILAVI Wireless Communication Protocol for the Building Environment – Security Issues"*. IEEE Tenth International Symposium on Consumer Electronics, 2006. ISCE '06. IEEE.

Soini, M., Sikkilä, H., Oksa, P., Sydänheimo, L., Kivikoski, M. (2006). "*KILAVI Wireless Communication Protocol for the Building Environment – Networking Issues"*. IEEE Tenth International Symposium on Consumer Electronics, 2006. ISCE '06. IEEE.

Transmission Control Protocol (1981). IETF Standards Track, RFC 793. *"Transmission Control Protocol"*. <http://tools.ietf.org/rfc/rfc793.txt>. The Internet Engineering Task Force, September 1981. Accessed 20 Aug. 2007.

Wang, K., Su, R., Li, Z., Cai, Z., Zhou, L. (2005). *"Study of Secure Complicated Information System Architecture Model"*. Proceedings of the First IEEE International Conference on Semantics, Knowledge and Grid, 2005. SKG'05. IEEE, Nov 2005. Page(s): 101-101.

Wang, S. Y., Chou, C. L., Huang, C. H., Hwang, C. C., Yang, Z. M., Chiou, C. C., Lin, C. C. (2003). *"The Design and Implementation of the NCTUns 1.0 Network Simulator"*. Computer Networks, Vol. 42, Issue 2, June 2003, Page(s): 175-197

Wikipedia, The Free Encyclopedia (Updated 24.7.2007). *"TCP Congestion Avoidance Algorithm"*. [Online], Available: http://en.wikipedia.org/wiki/TCP_congestion_avoidance_algorithm. Accessed 13 Aug. 2007.

Yu, L., Wang, N., Meng, X. (2005). *"Real-time Forest Fire Detection with Wireless Sensor Networks"*. Proceedings on IEEE International Conference on Wireless Communication, Networking and Mobile Computing, 2005. IEEE, Volume 2, 23-26 Sept. 2005 Page(s): 1214 – 1217.

HYDROGEN POWERED CAR CONTROL SYSTEM

Srovnal Vilem, Koziorek Jiri, Horak Bohumil
Department of Measurement and Control, FEECS,VSB
Technical University of Ostrava, 17. listopadu 15, 708 3, Ostrava-Poruba, Czech Republic
jiri.koziorek@vsb.cz, vilem.srovnal@vsb.cz

Adam George, Garani Georgia
Technological Educational Institute of Larissa, Greece
gadam@teilar.gr, garani@teilar.gr

Keywords: Vehicle Control System, Distributed Control System, Multi-agents, Learning, Optimizing.

Abstract: The main goal of the research project was designing and realization a distributed control system of the hydrogen powered prototype car. Next goals of project were real time control, speed and final time optimizing with minimal fuel consumption and monitoring of driver biomedical parameters. The control system was realized by several mobile embedded systems and one central system. The embedded systems hardware was realized with Freescale processors and communication CAN bus. Central system hardware was realized by notebook and communication with embedded systems in car was realized by GSM communication. Control system software using of multi-agent technology with dynamic mutual negotiation of mobile system parts. This task allows in a form of control system for prototype race car modelling of distributed control system. The real hardware and software model is also important motivation for extended research.

1 INTRODUCTION

A team of several specialists and students of Department of Measurement and Control, VSB-Technical University of Ostrava have designed and realized a prototype of hydrogen powered car based on fuel cell technology and electrical DC drive. The project is called HydrogenIX and the works and testing activities came through between October 2004 and today.

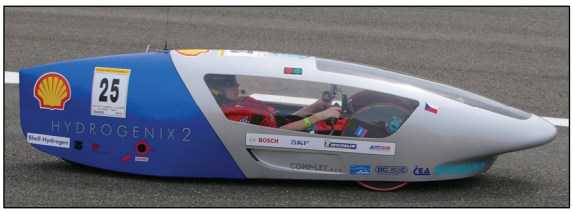


Figure 1: The HydrogenIX car.

The motivations for the project are following:

- There is The Laboratory of Fuel Cells at Department of Measurement and Control. The development of mentioned car is first application of fuel cell in mobile system at the laboratory.
- Activation of the interest of students, Ph.D. students, researchers and public in renewable and alternative energy sources.
- Involve students to design and development activities in interesting area and demonstrate the result of the project in a competition of economization of energy in mobile vehicles. The competition is called Shell Eco-Marathon.

The Shell Eco-Marathon is a competition organized by Shell Company and take place at race circuit in Nogaro, France. Teams of whole Europe try to reach highest distance with 1 liter of petrol, in the other words to have lowest consumption of the fuel. Even if the majority of teams use petrol engines in their vehicles, there are also vehicles powered by diesel, LPG, hydrogen and other alternative energies. The results are obtained by recalculating using calorific value of each type of fuel. So that it is possible to compare different types of fuel.

2 CONTROL SYSTEM

The vehicle powered by hydrogen fuel cell needs electronic a control system assuring operation of its different parts. The complex electronic control is necessary already for basic operation of the vehicle, because there are lots of subsystems that have to be coordinated and controlled. The control system assures especially following tasks:

- Control of fuel cell operation – hydrogen input valve control, combustion products output valve control, fuel cell fan control, coupling of produced electrical energy to electric DC-drive system.
- Control of DC-drive system – motor current control, speed control.
- Processing security tasks – assuring safe operation of fuel cell system and drive system, processing of hydrogen detector information, temperature measuring.
- Managing the driver control panel – complete interface to pilot that allows controlling the car – start/stop, speed set point, time measuring, emergency buttons and indicators.
- Creating data archives with saved process variables – saving important process data to archives that can be then exported and analyzed.
- Sending actual data to display panel in car – display panel in the car is the "process" visualization of the system. All important data are online displayed on it.
- Communication with PC monitoring station – control system send data and receive commands from PC monitoring station using wireless communication system.

The car onboard control system is built on embedded system with Freescale HC12 microprocessors. The control system has distributed architecture and it is divided into two parts:

- A fuel-cell control block that controls whole installation of the fuel-cell, DC drive system and security tasks.
- An interface control block that assures interface to the pilot, a wireless communication with PC monitoring station. This block contains the text display, which is used to monitor important parameter of the car and makes possible to do important settings.

Both part of control system are connected via CAN communication network. The wireless communication between the car and with PC monitoring station is realized by GSM communication – GPRS data transfer. The data transfer is realized by dial-up connection.

The PC monitoring station operates a process visualization application that is realized by SCADA system Promotic. The process visualization displays all parameters measured during the car operation, all the system states and alarms, make possible to display trends of required values and log measured data in data archives.

The complete block diagram of the car control system is demonstrated in figure 3 and realization in figure 2.

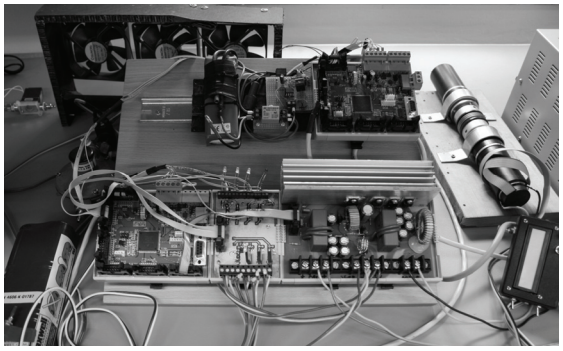


Figure 2: The HydrogenIX car control electronic testing workplace.

2.1 Operating Values Monitoring

The car control system monitors a lot of variables. Some of these variables are used for basic control activities, the others are used for optimization of operation. The measured variables are following:

- Electrical variables – fuel cell voltage and current, motor voltage and current, voltages of super-capacitor and on-board battery.
- Non-electrical variables – temperatures and pressures in fuel cell circuit, car speed.
- The system is ready for measurement of others supplementary variables that can be used for optimization of the operation – wind speed, outside temperature, track position.

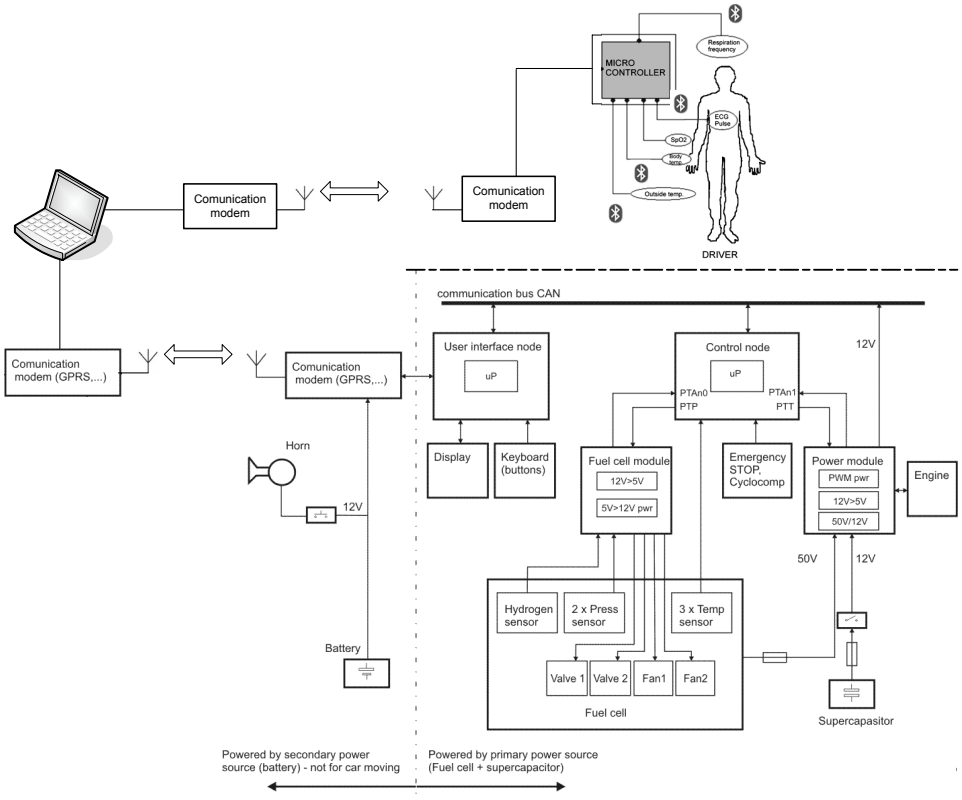


Figure 3: The HydrogenIX car control electronic block scheme.

2.2 Bio-telemetry System

The vehicle is also equipped by bio-telemetry system that makes possible to monitor biological functions of the pilot. The embedded portable telemetry system of biological parameters serves for reading and wireless data transfer of measured values of selected biological parameters to far computer.

The telemetric system can be used for real time monitoring of the basic life functions of race driver. The driver has to be very concentrated and the reactions of driver have to be very quick. The telemetry system provides better control of physical and psychical condition of driver during race. It is possible to analyze effect of a stress situations, high speed and high psychical stress on the race driver during the race and crisis situation, more precisely during high physical and psychical stress.

For biophysical monitoring were chosen these parameters: Electrocardiography – ECG, Pulse frequency, Oxygen saturation – SpO2, Body temperature, Outside temperature and Respiration frequency.

3 MULTI-AGENT CONCEPT OF CONTROL SYSTEM

The track passage optimization task of the laboratory car with minimal energy consumption in real time is quite complex.

Range of inputs and outputs of the control system, communication flows and safety of operation require the adaptability at occurred situations and environment changes – strategy control by multi-agent systems (MAS). Among basic expected properties of proposed MAS belong a strategic, targeted system behavior, robustness and adaptability at environment changes.

This can be provided by decentralization of control activities in the control system, by distribution of functions and by modularity based on fundamental elements – agents (Srovnal, V., Pavliska, A., 2002).

3.1 MAS Structure Description

The higher level of control system is represented by a personal computer. In the PC the signal from differentially GPS positioning system may be entered, which represents the relative coordinate system of environment – allow the precise of the position of the race car on the circuit. At the output is connected GPRS communication modem which transmits commands for race car.

The algorithm of agent's cooperation was proposed with the control agent on a higher level. The control agent determines the required behavior of the whole control system as the response to the dynamic behavior of car and to the one's own global strategy in the task and knowledge about the last situations, which are saved in the database. The agent on a higher level controls the other agents (Srovnal, V., Horák, B. and Bernatik, R., 2004).

The separate task is the transformation which converts the digital data position into the object coordinates (car position on the circuit) which are saved in the database of the circuit. This database is common for all agents in the control system. Each agent sees actual the whole data and is capable of controlling its behavior in a qualified way. The basic characteristic of a control algorithm of a subordinate agent is the independence on the number of decision making agents for car on the circuit.

Agent system has a common goal, to control of the car during race with optimizing - minimizing of fuel consumption and control of critical speed. For successful assertion of one's own race strategy the extraction and knowledge of changeable environment and learning capabilities are very important.

Main architecture of such hybrid agent system is characterized via:

- Layered control. Agent is described by number layers of abstraction and complexity.
- Layered knowledge base.
- Bottom-up activating
- Top-down execution.

Agent is connected with environment through interface with sensors, actuators and communication module. Control is allowed through layers at three levels: reactive layer, layer of local planning, and layer of cooperative planning. They are use information from knowledge bases ("world" model, "mental" model and "social" model), (Garani, G. and Adam, G., 2006).

Reactive layer is responsible for adequate reactions at the stimulations from environment that require immediate reaction and execution of called procedures from local planning layer. Fundamental characterization of such layer is:

- Use of effective algorithm of compare with patterns of behavior. Serve to pick-out of the actual situations.
- Situation description for timely actual reactions at received stimulus.
- Hard-wired links. Recognized situations are fix-connected with targets for reactive behavior. Immediate execution of program actions.
- Solution request of situations not–corresponding with couples situation-action are transmitted in local planning layer.
- Execution liability is coming from local planning layer activate procedures of reactive layer patterns of behavior.

Some situations can be not solved by execution of template action like an answer to stimulation from environment only, but they require certain level of deliberation. A function of plans creation for solving of the targets performs the layer of local planning.

Local planning layer have such fundamental data structures:

- Targets – state sets. Sets are characterized by attributes that are fulfilled at reaching targets.
- Planning – planning from second principles. Sets of plans are defined before in data structure – plans library. Mapping of target sets to plans library is existed. For each target is possible to assign the plan for its reaching.
- Plans library – contain the plans for reaching of agent targets.
- Scheduling – secure the timely limited plans stratification. Be created the plan schedules like the step sequences, to execute.

3.2 Cooperative Planning Layer

A basic control cycle of cooperative planning layer is creation, interpretation, decision making and execution of local plans.

In first phase the reports from nearby layers are processed. Reactive layer sends requests to solve new task or status of executed behavior templates. Schedules of active plans are actualized. Subsequently the status from reactive layer executed procedures is checked.

In case of successful procedures finalization the plan is erased from accumulator. Reports from highest layer are related to creation or cancellation of commitment for the plan execution at local base or plan evaluation. In case of plan execution request or his cancellation the accumulator of active plans is actualized.

The plan availability is a result of difference of his relative value for the agent and his costs for execution. The plan value is derived from target

value that is possible reach by plan. The plan costs are determined by function that assigns for every plan a real number calculated at basis of his fundamental action costs according to specific rules.

4 CONCLUSIONS

The algorithm of the control system should be proposed in a way so that it would ensure the requirements for the immediate response of control, so that the system of race car would be controlled in real-time. That is why, it is very important so that the algorithm for critical speed and fuel consumption would be optimized. The system response should be shorter than the time between two data frames from a GPS station. In the event that this limit is exceeded, the frame is cut out and the control quality may be decreased.

The main possibilities of algorithm adjustment are as follows:

- Dynamic control in the control and decision module of a control agent.
- The control and decision modules and communication protocol of the decision agents.
- The strategy of planning in the control model of the action agent.
- Learning of a race strategy and using the extraction results for decision rules generation as a part of the rules decision database of a decision agent.

ACKNOWLEDGEMENTS

The Grant Agency of Czech Academy of Science supplied the results of the project No. p. 1ET101940418 with subvention.

The paper was also supported by KONTAKT CZ-GR Project No. 7-2006-32 – “Communication Nets in Distributed Systems”.

REFERENCES

Srovnal, V., Pavliska, A., 2002. Robot Control Using UML and Multi-agent System. In: *Proceeding 6th World Multiconference SCI 2002*. Orlando 2002, pp.306-311

Srovnal, V., Horák, B., Bernatik, R., 2004. *Strategy extraction for mobile embedded control systems apply the multi-agent technology*. Lecture notes in computer science, Vol. 3038. Springer-Verlag, Berlin Heidelberg 2004, pp. 631-637.

Garani, G., Adam, G., 2006. Qualitative Modelling of Manufacturing Machinery. In: *Proceedings of the 32nd IEEE Conference on Industrial Electronics*, Paris 2006, pp. 3591-3596

IDENTIFICATION OF THE DYNAMIC PARAMETERS OF THE C5 PARALLEL ROBOT

B. Achili[*+], B. Daachi[*], Y. Amirat[*] and A. Ali-Cherif[+]
[+]*Laboratoire d'Informatique Avancée de Saint Denis, 2 rue de la liberté, 93526 Saint Denis Cedex, France*
[*]*Laboratoire Images, Signaux et Systèmes intelligents, 122-124 rue Paul Armangot, 94400 Vitry/seine, France*
achili@ai.univ-paris8.fr

Keywords: Identification, Parallel Robot, Least Squares Method, Cross validation.

Abstract: This paper deals with the experimental identification of the dynamic parameters of the C5 parallel robot. The inverse dynamic model of the robot is formulated under the form of linear equation with respect to the dynamic parameters. Moreover, a heuristic procedure for finding the exciting trajectory has been conducted. This trajectory is based on Fourier series whose coefficients are determined by using a heuristic method. The least squares method has been applied to solve an over-determined linear system which is obtained by sampling the dynamic model along the exciting trajectory. The experimental results show the effectiveness of the identification procedure.

1 INTRODUCTION

A parallel architecture is a closed-loop mechanism in which the end-effector (mobile platform) is connected to the base by at least two independent kinematic chains. The pioneering works in this field are those of Stewart who proposed in 1965 a parallel platform with 6 DOF. Since then, several authors have proposed a large variety of designs and studies. Parallel architectures were first used for building flight simulators and tire testers. Since then, they were used in other applications like the handling of heavy objects with great accelerations, or the assembly of parts requiring high precision. More recently, parallel robots appeared in the medical field . The latter requires the design of very precise parallel machines performing in a limited workspace

Because of their structure, serial robots have limited dynamic performances. In the other hand, due to their reducted inertia, parallel robots allow for the reduction of coupling dynamic effects and consequently to better dynamic performance.

In the literature, several techniques were proposed for the identification of dynamic parameters of robot. A CAD method based on identifying inertia parameters is proposed in (An et al, 85). Usually these methods lead to an unsufficient precision of inertia parameters estimation and do not allow for the determination of other dynamic parameters (viscous friction, coulomb friction). For better results, an estimation of the whole dynamic parameters of the assembled robot is required.

The identification procedure consists usually of four main steps: (1) Calculation of an identifiable dynamic model, (2) Generation of the optimized excitation trajectory, (3) Estimation of the dynamic parameters, and finally (4) Validation of the obtained model.

The first step consists of calculating the minimal set of dynamic parameters to be identified (set of base parameters). This set can be computed by using the QR decomposition of observation matrix (Gautier, 91). In the second step, the optimal exciting trajectory is calculated in order to guarantee the relevance of the measured data. This step includes the choice of an optimization criterion.

The third step consists of estimating the dynamic parameters from the measured data. Least squares method is one of the most widely used estimation method. It consists of solving an over-determined linear system (Janot, 07). An improvement over the classical LS method is the use of a Weighted Least Squares (WLS) estimator, (Renaud et al, 06). Another approach is the Maximum Likelihood Estimator(MLE) whose principle assumes that the true parametric model is known exactly (Swevers et al, 97). Other estimators like the ellipsoidal algorithm or the interval analysis (Poignet et al, 03) have been proposed in the literature.

The fourth step of identification procedure consists of validating the identified dynamic model. In

most cases, this is realized by comparing the predicted and the measured torques for a trajectory which is different from the exciting trajectory.

In this paper we present the identification of dynamic parameters of the 6 DOF parallel robot with C5 joints. First, the dynamic model is expressed as a linear relation with respect to the dynamic parameters. The parameters are estimated by the classical technique of least squares solving an overdetermined linear system obtained from a sampling of the dynamic model, along the exciting trajectory.

The paper is organized in four sections. First one describes the mechanical architecture of the C5 parallel robot. Second section presents the inverse dynamic model of the robot. Section 3 presents an estimation of the dynamic parameters of the robot. Calculation of the exciting trajectory along with the data filtering procedure are developed in section 4. Fifth section is dedicated to the presentation and analysis of the experimental results including the cross validation procedure. Finally, a conclusion and some perspectives are given in the last section.

2 DESCRIPTION OF THE C5 PARALLEL ROBOT

The C5 parallel robot consists of a static part and a mobile part connected together by six actuated links. Each segment is embedded to the static part at point A_i and linked to the mobile part through a spherical joint attached to two crossed sliding plates at point B_i (Fig. 1).

Theoretical study concerning this architecture has been presented in the literature. The C5 links parallel robot is equipped with six linear actuators; each of them is driven by a DC motor. Each motor drives a ball and screw arrangement. The position measurements are obtained from six incremental encoders, which are tied to the DC motors.

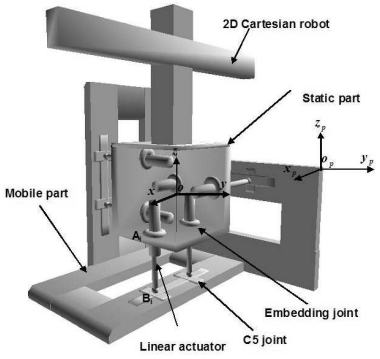


Figure 1: Parallel robot.

3 MODELING OF THE C5 ROBOT

3.1 Inverse Dynamic Model

The inverse dynamic model of the C5 parallel robot is given in (Khalil et al, 04):

To solve our identification problem, we rewrite the inverse dynamic model to make it linear with respect to dynamic parameters (Poignet et al, 02). The dynamic model is rewritten then as follows:

$$\Gamma = D(\omega_p, \dot{\omega}_p, \dot{V}_p, q, \dot{q}, \ddot{q})\ X_s \qquad (1)$$

with

- Γ : (6×1) torque vector
- D : (6×34) observation matrix
- X_s : (34×1) standard parameters vector:

$$X_s = (X_{s1}\ \ X_{s2}\ \ X_{s3}\ \ X_{s4}\ \ X_{s5})^T$$
$$X_{s1} = (M_1\ \ M_2\ \ M_3\ \ M_4\ \ M_5\ \ M_6\ \ M_p)$$
$$X_{s2} = (XX\ \ XY\ \ XZ\ \ YY\ \ YZ\ \ ZZ\ \ MX\ \ MY\ \ MZ)$$
$$X_{s3} = (I_{a1}\ \ I_{a2}\ \ I_{a3}\ \ I_{a4}\ \ I_{a5}\ \ I_{a6})$$
$$X_{s4} = (F_{v1}\ \ F_{v2}\ \ F_{v3}\ \ F_{v4}\ \ F_{v5}\ \ F_{v6})$$
$$X_{s5} = (F_{s1}\ \ F_{s2}\ \ F_{s3}\ \ F_{s4}\ \ F_{s5}\ \ F_{s6})$$

4 DYNAMIC PARAMETERS IDENTIFICATION

For the purpose of dynamic parameters identification, we use the formulation given in (Janot et al, 07). The principle of identification consists in sampling the inverse dynamic model of the robot with respect to the base parameters, obtained by *QR* decomposition (Gautier, 91) along the exciting trajectory. A filtering process is applied to the measured data in order to obtain a good estimation of dynamic parameters. This technique allows us to obtain an over-determined linear system of full rank.

5 EXCITING TRAJECTORY CALCULATION

The quality of the exciting trajectory can be evaluated through a good condition number of the regressor matrix. The calculation of this trajectory can be done by nonlinear optimization. In our case, we used an exciting trajectory based on Fourier series (Swevers et al, 91). For each segment j $(j = 1, 2, ...6)$,

the position q_j can be written as follows:

$$q_j(t) = q_{j,0} + \sum_{k=1}^{M} (a_{j,k}\sin(k\omega_f t) + (b_{j,k}\cos(k\omega_f t)) \quad (2)$$

with

- ω_f the fundamental pulsation of the finite Fourier series.
- t the time.
- $a_{j,k}$ and $b_{j,k}$ $(k = 1,...5)$ the amplitudes of sine and cosine functions
- $q_{j,0}$ is the initial value of the position trajectory.

In order to excite the robot in the bandwidth of the position closed loop, $f_{dyn} < 2\,Hz$, we have chosen the fundamental frequency of trajectories equal to $0.1Hz$ and the number of harmonics $k = 5$.

As the number of Fourier series coefficients is high, it is difficult to determine them by a nonlinear optimization. For this reason, we calculate these coefficients in a heuristic way. The calculation of these parameters is based on the motion constraints which are imposed by physical limitations of robot. These constraints can be expressed as follows:

$$-0.05m < q_j(\alpha) < +0.05m \quad (3)$$

$$-0.1m/s < \dot{q}_j(\alpha) < +0.1m/s \quad (4)$$

$$-0.5m/s^2 < \ddot{q}_j(\alpha) < +0.5m/s^2 \quad (5)$$

Where:

Vector α includes the trajectory parameters $q_{j,0}$, $a_{j,k}$ and $b_{j,k}$.

The heuristic approach allows us to find the exciting trajectories shown in Fig. 2.

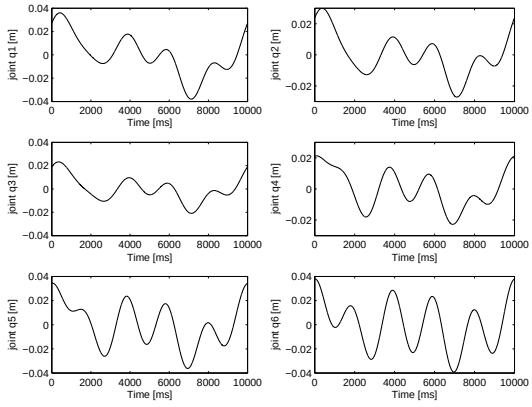


Figure 2: Exciting trajectories.

5.1 Data Filtering

For a good estimation of the dynamic parameters, the measurements signals need to be filtered . So the position is filtered by the 4^{th} order Butterworth filter. The vector Y and each column of matrix W are filtered by the 8^{th} order Tchebychev filter and are resampled at lower rate in order to reject the high frequency ripples of the measured torques. The computation of joint velocities and accelerations is made by using the central difference algorithm in order to avoid any distortion of phase and amplitude.

6 EXPERIMENTAL RESULTS

The table given in Fig. 3 shows the estimated base parameters. The relative standard deviations are also given.

parameters	identified values $\hat{X}$	relative standard deviations $\%\sigma_{\hat{X}jr}$
$M_1 + I_{a1}(kg)$	0.5435	7.0629
$M_2 + I_{a2}(kg)$	0.4075	11.3082
$M_3 + I_{a3}(kg)$	0.5436	7.0617
$M_4 + I_{a4}(kg)$	0.5909	6.7516
$M_5 + I_{a5}(kg)$	0.4909	8.8692
$M_6 + I_{a6}(kg)$	0.5718	6.7108
$M_p(kg)$	8.2652	9.6980
$XX(kg.m^2)$	0.1035	1.7892
$YY(kg.m^2)$	0.2124	5.8701
$ZZ(kg.m^2)$	0.0178	7.0021
$MX(kg.m)$	7.5925	2.0799
$MY(kg.m)$	-1.8212	8.6681
$MZ(kg.m)$	21.2650	16.1002
$F_{v1}(N.m.s.rad^{-1})$	8.8464	2.0209
$F_{v2}(N.m.s.rad^{-1})$	7.9940	2.2183
$F_{v3}(N.m.s.rad^{-1})$	8.7253	2.1198
$F_{v4}(N.m.s.rad^{-1})$	7.5517	2.2366
$F_{v5}(N.m.s.rad^{-1})$	8.1706	2.2777
$F_{v6}(N.m.s.rad^{-1})$	8.7312	2.1182
$F_{s1}(N.m)$	0.5034	5.1484
$F_{s2}(N.m)$	0.3424	7.4246
$F_{s3}(N.m)$	0.2344	9.3723
$F_{s4}(N.m)$	0.2705	8.3329
$F_{s5}(N.m)$	0.1753	13.1492
$F_{s6}(N.m)$	0.2335	9.4081

Figure 3: Identified parameters.

Note that the dynamic parameters present in most cases a relative standard deviation lower than 10%, which represents a good estimation. However the relative standard deviation of the parameters MZ, and F_{S5} is higher than 10%. This deviation is due to mechanical constraints, consequently, we conclude that the obtained results are encouraging and we can state that these identification results are globally acceptable.

In order to validate the estimated dynamic parameters, we proceed to a cross validation which consists in comparing the measured torques with those obtained by the inverse dynamic model with the identified parameters. The trajectory which is used for this validation has not been used previously for the identification. Figure 4 show the results of this validation. For the others axis, we have also obtained the same ting than figure 4

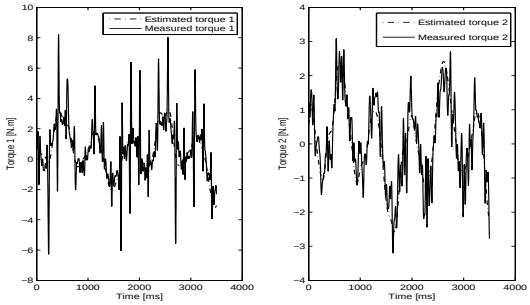


Figure 4: Estimated and measured torque for the joints 1 and 2.

Note that the calculated torques using the inverse dynamic model with the estimated parameters are close to those measured on the robot. Consequently, one can conclude that the estimation of the dynamic parameters, using the least squares method is valid.

7 CONCLUSIONS

In this paper, we identified the physical parameters of the C5 parallel robot. The identification is based on the least squares method. The application of this identification method uses an exciting trajectory calculated from a heuristic approach. To validate the identified parameters, we considered another trajectory different from that used in identification. The cross validation enables us to conclude with the effectiveness of the considered identification. In short term of our project, we propose to include the joint elasticity, which is the major source of flexibility in many practical applications.

REFERENCES

Swevers J., Ganseman C., Bilgin D., De Schutter J., Van Brussel H., 1997. *Optimal robot excitation and identification*. IEEE Transactions on Robotics and Automation, 13(5):730–740.

Renaud, P., Vivas, A., Andreff, A., Poignet, P., 2006. Martinet, P., Pierrot, F., Company, O., *Kinematic and dynamic identification of parallel mechanisms*, In Conrol Engineering Practice 14, pp1099 - 1109.

Poignet, P., Ramdani, N., Vivas, A., 2003. *Robust estimation of parallel robot dynamic parameters with interval analysis*, Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 6503-6508, Maui, Hawaii, USA.

Khalil, W., Ibrahim, O., 2004. *General Solution for the Dynamic Modeling of Parallel Robots*, lnternational Conference on Robotics & Automation, New Orleans, LA.

Gautier, M., Poignet, P., 2002. *Identification en boucle fermée par modèle inverse des paramètres physiques de systèmes mécatroniques*, Journal Eurepeen des Systèmes Automatisés,36:465-480.

Gautier M., 1991. *Numerical calculation of the base inertial parameters*, Journal of Robotics Systems, Vol. 8, No. 4, pp. 485-506.

An, C. H., Atkenson, C. G., Hollerbach, J. H., 1985. *Estimation of inertial parameters of rigid body links of manipulators*, Proceedings of the 24th Conference on Decision and control, pp. 990-995.

Janot, A., Bidard, C., Gosselin, F., Gautier, M., Keller, D., Perrot, Y., 2007. *Modeling and Identification of a 3 DOF Haptic Interface*,IEEE International Conference on Robotics and Automation Roma, Italy, pp 4949-4955

DETECTION AND CONTROL OF NON-LINEAR BEHAVIOR BY SLIDING MODES CONTROL IN A 3 D.O.F. ROBOT

Claudio Urrea and Marcela Jamett
Departamento de Ingeniería Eléctrica, Universidad de Santiago de Chile (USACH)
Av. Ecuador 3519, Santiago, Chile
eurrea@lauca.usach.cl, mjamett@usach.cl

Keywords: Planar Robot Model, Chaotic behaviour, Sliding Modes.

Abstract: Results from simulations of a Planar Robot Model, when it is placed in the same plane of the action of the gravity force are reviewed in this paper. The model includes several parameters usually discarded in current models, such as Driving, and Non-linear Friction, for an industrial-type robotic manipulator and its actuators. When we develop more exact representations of the dynamics of a manipulator and their actuators, chaotic behavior is detected for certain parameter values of the robotic manipulator. This chaotic behavior – without external inputs – was exactly controlled by Sliding Modes.

1 INTRODUCTION

A variable structure system (VSS) is a system whose structure is intentionally changed to achieve the desired performance. This intentional structural change is typically accomplished through discontinuous control action in accordance with a presigned algorithm and switching hyperplanes (Zohdy, M., Fadali, M.S. and Liu, J., 1992).

For nonlinear dynamical systems with uncertainties and disturbance, the sliding modes control (SMC) is a method which has some advantages. The SMC was proposed and elaborated initially in the fifties in the former Soviet Union (Park, D. and Choi, S., 1999). This control method, which can be obtained by an appropriate discontinuous law, is the principal operation mode in the variable structure control system (VSCS). This method is known for its robustness to disturbance and parameters variations (Bartoszewicz, A., 1995).

The sliding mode controllers have excellent characteristic in the sliding movement of the state on the sliding surface. During this sliding movement, the system has invariants properties, producing a robust movement regarding the unknown parameters of the system and the external interferences. The design of the VSS based sliding mode controllers can be broken down into two major phases: the first one is the determination of a stable manifold, called the sliding surface, and the second phase is to design a switching control law according to the sliding surface to satisfy the attraction manifold. When the sliding mode occurs, the system state will remain on it forever and the system behaves as an equivalent system with desired dynamics which is governed by the sliding surface equation; at the same time the system has good characteristics such as fast response, good robustness and disturbance rejection, etc. (Xu, J., Lee, T.H., Wang, M. and Yu, X., 1996).

Efficient control of industrial robots is an important issue to success of industrial automation in these years (Lu, X. and Spurgeon, S., 1999). Along with the development of robot manipulator control theory and its applications, there has been increasing demand for more efficient control schemes to achieve satisfactory results (Chen, C. and Xu, R., 1999).

The paper is organized as follows: in section 2, a friction model is given. In section 3, manipulator dynamic equations are presented. In section 4, the state-space model for a 3 link planar robot and its actuators, is developed. In section 5, the robot controller is developed. Section 6 presents some simulation results. Finally, in section 7, the conclusions are discussed.

2 FRICTION MODEL

Models representing friction effects have been widely studied in concerned literature (Canudas, C., Aström, K. and Braun, K., 1987), (Kircanski, N. and

Goldenberg, A., 1997), (Urrea, C., 1999). In this paper, we use a model that includes both effects, whose friction curve is discontinuous and non-symmetrical.

$$F(\dot{\theta}) = (\alpha_0 + \alpha_1 \cdot e^{(\dot{\theta}/v_0)^2}) \cdot sgn(\dot{\theta}) + \alpha_2 \cdot \dot{\theta} \quad (1)$$

where: $F(\dot{\theta})$: is the friction torque [N · m]; $\dot{\theta}$: angular velocity [rad/s]; $\alpha_0 + \alpha_1$: static friction [N· m]; v_0: Stribeck velocity [rad/s]; α_2: viscous friction [N·m·s/rad].

3 MANIPULATOR DYNAMIC EQUATIONS

The dynamic equation of a robotic manipulator in the joint space can be written as follows (Leahy, M., Valavanis, K. and Saridis, G., 1989); (Mahla, I., Urrea, C., 1999):

$$D(\theta) \cdot \ddot{\theta} + B(\theta, \dot{\theta}) = \tau \quad (2)$$

where:

$$B(\theta, \dot{\theta}) = C(\theta, \dot{\theta}) \cdot \dot{\theta} + F(\dot{\theta}) + G(\theta) \quad (3)$$

in which θ: joint angle vector, $\theta \in R^n$; $D(\theta)$: inertial matrix, $D(\theta) \in R^{nxn}$; $C(\theta,\dot{\theta})$: Coriolis and centrifugal torque matrix, $C(\theta,\dot{\theta}) \in R^{nxn}$; $G(\theta)$: gravity torque vector, $G(\theta) \in R^n$; τ: joint torque vector, $\tau \in R^n$; n: degrees of freedom.

4 STATE-SPACE MODELS

4.1 Manipulator

If in an industrial-type robotic manipulator, the following state variables are chosen:

$$\begin{aligned} x_1 &= \theta_1;\ x_2 = \dot{\theta}_1;\ x_3 = \theta_2; \\ x_4 &= \dot{\theta}_2; x_5 = \theta_3; x_6 = \dot{\theta}_3 \end{aligned} \quad (4)$$

then:

$$\dot{x}_2 = \ddot{\theta}_1;\ \dot{x}_4 = \ddot{\theta}_2,;\ \dot{x}_6 = \ddot{\theta}_3 \quad (5)$$

If some of the elements in the inertial matrix $D(\theta(t))$ are defined as constants, there it is obtained:

$$k_1 = m_1 \cdot l_{c1} + (m_2 + m_3) \cdot l_1 \quad (6)$$

$$k_2 = m_2 \cdot l_{c2} + m_3 \cdot l_2 \quad (7)$$

$$k_3 = m_3 \cdot l_3 \quad (8)$$

where m_1: mass of the first link [kg]; m_2: mass of the second link [kg]; m_3: mass of the third link [kg]; l_1: length of the first link [m]; l_2: length of the second link [m]; l_3: length of the third link [m]; l_{c1}: distance between the gravity centre of the first link and its driving axis [m]; l_{c2}: distance between the gravity centre of the second link and its driving axis [m]. Replacing eq. 1 and eqs. 4 to 8 into eqs. 2 and 3 (n = 3), we have:

$$\begin{aligned} \tau_{L1}^* = {} & (I_1 + I_2 + I_3 + m_1 \cdot l_{c1}^2 + m_2 \cdot (l_1^2 + l_{c2}^2) + m_3 \cdot (l_1^2 + l_2^2 + l_{c3}^2) + 2 \cdot k_2 \cdot l_1 \cdot \cos x_3 + 2 \cdot k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + l_2 \cdot \cos(x_5))) \cdot \dot{x}_2 / N_1 + (I_2 + I_3 + m_2 \cdot l_{c2}^2 + m_3 \cdot (l_2^2 + l_{c3}^2) + k_2 \cdot l_1 \cdot \cos(x_3) + k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + 2 \cdot l_2 \cdot \cos(x_5))) \cdot \dot{x}_4 / N_1 + (I_3 + m_3 \cdot l_{c3}^2 + k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + l_2 \cdot \cos(x_5))) \cdot \dot{x}_6 / N_1 + (k_2 \cdot (-l_1 \cdot \sin(x_3) \cdot (2 \cdot x_2 \cdot x_4 + x_4^2)) + k_3 \cdot (-l_1 \cdot \sin(x_3 + x_5) \cdot (2 \cdot x_2 \cdot (x_4 + x_6) + (x_4 + x_6)^2) - l_1 \cdot \sin(x_5) \cdot (2 \cdot (x_2 + x_4) \cdot x_6 + x_6^2))) / N_1 + g \cdot (k_1 \cdot \cos(x_1) + k_2 \cdot \cos(x_1 + x_3) + k_3 \cdot \cos(x_1 + x_3 + x_5)) / N_1 + F_1(x_2) / N_1 \end{aligned} \quad (9)$$

$$\begin{aligned} \tau_{L2}^* = {} & (I_2 + I_3 + m_2 \cdot l_{c2}^2 + m_3 \cdot (l_2^2 + l_{c3}^2) + k_2 \cdot (l_1 \cdot \cos(x_3) + k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + 2 \cdot l_2 \cdot \cos(x_5))) \cdot \dot{x}_2 / N_2 + (I_2 + I_3 + m_2 \cdot l_{c2}^2 + m_3 \cdot (l_2^2 + l_{c3}^2) + k_3 \cdot 2 \cdot l_2 \cdot \cos(x_5)) \cdot \dot{x}_4 / N_2 + (I_3 + m_3 \cdot l_{c3}^2 + k_3 \cdot l_2 \cdot \cos(x_5)) \cdot \dot{x}_6 / N_2 + (k_2 \cdot (l_1 \cdot \sin(x_3 + x_5) \cdot x_2^2) + k_3 \cdot (l_1 \cdot \sin(x_3 + x_5) \cdot x_2^2 - l_2 \cdot \sin(x_6) \cdot (2 \cdot (x_2 + x_4) \cdot x_6 + x_6^2))) / N_2 + g \cdot (k_2 \cdot \cos(x_1 + x_3) + k_3 \cdot \cos(x_1 + x_3 + x_5)) / N_2 + F_2(x_4) / N_2 \end{aligned} \quad (10)$$

$$\begin{aligned} \tau_{L3}^* = {} & (I_3 + m_3 \cdot l_{c3}^2 + k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + l_2 \cdot \cos(x_5))) \cdot \dot{x}_2 / N_3 + (I_3 + m_3 \cdot l_{c3}^2 + k_3 \cdot l_2 \cdot \cos(x_5)) \cdot \dot{x}_4 / N_3 + (I_3 + k_3^2) \cdot \dot{x}_6 / N_3 + k_3 \cdot (l_1 \cdot \sin(x_3 + x_5) \cdot x_2^2 + l_2 \cdot \sin(x_5) \cdot (x_2 + x_4)^2) / N_3 + g \cdot k_3 \cdot \cos(x_1 + x_3 + x_5) / N_3 + F_3(x_6) / N_3 \end{aligned} \quad (11)$$

where τ_{L1}^*: torque applied in the first link, referred to the first motor axis [N·m]; τ_{L2}^*: torque applied in the second link, referred to the second motor axis [N ·m]; τ_{L3}^*: torque applied in the third link, referred to the third motor axis [N·m]; N_1: reduction factor of the first gear train; N_2: reduction factor of the second gear train; N_3: reduction factor of the third gear train; I_1: moment of inertia of the first link [Kg· m]; I_2: moment of inertia of the second link [Kg·m]; I_3: moment of inertia of the third link [Kg·m]; F_1: is the friction torque in the first link axis, [N·m]; F_2: is the friction torque in the second link axis, [N · m]; F_3: is the friction torque in the third link axis, [N· m]; g: is the gravity force [N·m].

Defining the following functions:

$$f_1^* = (I_1 + I_2 + I_3 + m_1 \cdot l_{c1}^2 + m_2 \cdot (l_1^2 + l_2^2) + m_3 \cdot (l_1^2 + l_2^2 + l_{c3}^2) + 2 \cdot k_2 \cdot l_1 \cdot \cos x_3 + 2 \cdot k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + l_2 \cdot \cos(x_5))) / N_1 \quad (12)$$

$$f_2^* = (I_2 + I_3 + m_2 \cdot l_{c2}^2 + m_3 \cdot (l_2^2 + l_{c3}^2) + k_2 \cdot l_1 \cdot \cos(x_3) + k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + 2 \cdot l_2 \cdot \cos(x_5))) / N_1 \quad (13)$$

$$f_3^* = (I_3 + m_3 \cdot l_{c3}^2 + k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + l_2 \cdot \cos(x_5))) / N_1 \quad (14)$$

$$f_4^* = (k_2 \cdot (l_1 \cdot \sin(x_3) \cdot (2 \cdot x_2 \cdot x_4 + x_4^2)) + k_3 \cdot (l_1 \cdot \sin(x_3 + x_5) \cdot (2 \cdot x_2 \cdot (x_4 + x_6) + (x_4 + x_6)^2) + l_1 \cdot \sin(x_5) \cdot (2 \cdot (x_2 + x_4) \cdot x_6 + x_6^2))) / N_1 - g \cdot (k_1 \cdot \cos(x_1) + k_2 \cdot \cos(x_1 + x_3) + k_3 \cdot \cos(x_1 + x_3 + x_5)) / N_1 - F_1(x_2) / N_1 \quad (15)$$

$$f_5^* = (I_2 + I_3 + m_2 \cdot l_{c2}^2 + m_3 \cdot (l_2^2 + l_{c3}^2) + k_2 \cdot (l_1 \cdot \cos(x_3) + k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + 2 \cdot l_2 \cdot \cos(x_5))) / N_2 \quad (16)$$

$$f_6^* = (I_2 + I_3 + m_2 \cdot l_{c2}^2 + m_3 \cdot (l_2^2 + l_{c3}^2) + k_3 \cdot 2 \cdot l_2 \cdot \cos(x_5)) / N_2 \quad (17)$$

$$f_7^* = (I_3 + m_3 \cdot l_{c3}^2 + k_3 \cdot l_2 \cdot \cos(x_5)) / N_2 \quad (18)$$

$$f_8^* = -(k_2 \cdot (l_1 \cdot \sin(x_3 + x_5) \cdot x_2^2) + k_3 \cdot (l_1 \cdot \sin(x_3 + x_5) \cdot x_2^2) - l_2 \cdot \sin(x_6) \cdot (2 \cdot (x_2 + x_4) \cdot x_6 + x_6^2))) / N_2 - g \cdot (k_2 \cdot \cos(x_1 + x_3) + k_3 \cdot \cos(x_1 + x_3 + x_5) / N_2 - F_2(x_4) / N_2 \quad (19)$$

$$f_9^* = (I_3 + m_3 \cdot l_{c3}^2 + k_3 \cdot (l_1 \cdot \cos(x_3 + x_5) + l_2 \cdot \cos(x_5))) / N_3 \quad (20)$$

$$f_{10}^* = (I_3 + m_3 \cdot l_{c3}^2 + k_3 \cdot l_2 \cdot \cos(x_5)) / N_3 \quad (21)$$

$$f_{11}^* = (I_3 + k_3^2) / N_3 \quad (22)$$

$$f_{12}^* = -k_3 \cdot (l_1 \cdot \sin(x_3 + x_5) \cdot x_2^2 + l_2 \cdot \sin(x_5) \cdot (x_2 + x_4)^2) / N_3 - g \cdot k_3 \cdot \cos(x_1 + x_3 + x_5) / N_3 - F_3(x_6) / N_3 \quad (23)$$

then:

$$\tau_{L1}^* + f_4^* = f_1^* \cdot \dot{x}_2 + f_2^* \cdot \dot{x}_4 + f_3^* \cdot \dot{x}_6 \quad (24)$$

$$\tau_{L2}^* + f_8^* = f_5^* \cdot \dot{x}_2 + f_6^* \cdot \dot{x}_4 + f_7^* \cdot \dot{x}_6 \quad (25)$$

$$\tau_{L3}^* + f_{12}^* = f_9^* \cdot \dot{x}_2 + f_{10}^* \cdot \dot{x}_4 + f_{11}^* \cdot \dot{x}_6 \quad (26)$$

Defining:

$$f_{13}^* = 1/[f_3^* \cdot f_5^* \cdot f_{10}^* - f_3^* \cdot f_9^* \cdot f_6^* - f_2^* \cdot f_5^* \cdot f_{11}^* - f_2^* \cdot f_9^* \cdot f_7^* \cdot f_6^* \cdot f_{11}^* - f_1^* \cdot f_{10}^* \cdot f_7^*] \quad (27)$$

$$f_{14}^* = f_{13}^* \cdot [f_6^* \cdot f_{11}^* - f_{10}^* \cdot f_7^*] \quad (28)$$

$$f_{15}^* = f_{13}^* \cdot [f_9^* \cdot f_7^* - f_5^* \cdot f_{11}^*] \quad (29)$$

$$f_{16}^* = f_{13}^* \cdot [f_5^* \cdot f_{10}^* - f_9^* \cdot f_6^*] \quad (30)$$

$$f_{17}^* = f_{13}^* \cdot [f_3^* \cdot f_{10}^* - f_2^* \cdot f_{11}^*] \quad (31)$$

$$f_{18}^* = f_{13}^* \cdot [f_1^* \cdot f_{11}^* - f_3^* \cdot f_9^*] \quad (32)$$

$$f_{19}^* = f_{13}^* \cdot [f_2^* \cdot f_9^* - f_1^* \cdot f_{10}^*] \quad (33)$$

$$f_{20}^* = f_{13}^* \cdot [f_2^* \cdot f_7^* - f_3^* \cdot f_6^*] \quad (34)$$

$$f_{21}^* = f_{13}^* \cdot [f_3^* \cdot f_5^* - f_1^* \cdot f_7^*] \quad (35)$$

$$f_{22}^* = f_{13}^* \cdot [f_1^* \cdot f_6^* - f_2^* \cdot f_5^*] \quad (36)$$

Redefining functions,

$$f_{23}^* = f_{14}^* \cdot f_4^* \quad (37)$$

$$f_{24}^* = f_{15}^* \cdot f_8^* \quad (38)$$

$$f_{25}^* = f_{16}^* \cdot f_{12}^* \quad (39)$$

$$f_{26}^* = f_{17}^* \cdot f_4^* \quad (40)$$

$$f_{27}^* = f_{18}^* \cdot f_8^* \quad (41)$$

$$f_{28}^* = f_{19}^* \cdot f_{12}^* \quad (42)$$

$$f_{29}^* = f_{20}^* \cdot f_4^* \quad (43)$$

$$f_{30}^* = f_{21}^* \cdot f_8^* \quad (44)$$

$$f_{31}^* = f_{22}^* \cdot f_{12}^* \quad (45)$$

The state equation model for the three-link planar RRR arm can be written as:

$$\dot{x}_1 = x_2 \quad (46)$$

$$\dot{x}_2 = f_{14}^* \cdot \tau_{L1}^* + f_{23}^* + f_{15}^* \cdot \tau_{L2}^* + f_{24}^* + f_{16}^* \cdot \tau_{L3}^* + f_{25}^* \quad (47)$$

$$\dot{x}_3 = x_4 \quad (48)$$

$$\dot{x}_4 = f_{17}^* \cdot \tau_{L1}^* + f_{26}^* + f_{18}^* \cdot \tau_{L2}^* + f_{27}^* + f_{19}^* \cdot \tau_{L3}^* + f_{28}^* \quad (49)$$

$$\dot{x}_5 = x_6 \quad (50)$$

$$\dot{x}_6 = f_{20}^* \cdot \tau_{L1}^* + f_{29}^* + f_{21}^* \cdot \tau_{L2}^* + f_{30}^* + f_{22}^* \cdot \tau_{L3}^* + f_{31}^* \quad (51)$$

4.2 Actuators

By employing state equations models for three DC motors, and from (Craig, J., 1996), we have equations (52) to (54):

$$\dot{x}_7 = [k_{a1} \cdot v_{a1}(t) - r_{a1} \cdot x_7 - k_{a1} \cdot k_{b1} \cdot x_2 \cdot N_1] / L_{a1} \quad (52)$$

$$\dot{x}_8 = [k_{a2} \cdot v_{a2}(t) - r_{a2} \cdot x_8 - k_{a2} \cdot k_{b2} \cdot x_4 \cdot N_2] / L_{a2} \quad (53)$$

$$\dot{x}_9 = [k_{a3} \cdot v_{a3}(t) - r_{a3} \cdot x_9 - k_{a3} \cdot k_{b3} \cdot x_6 \cdot N_3] / L_{a3} \quad (54)$$

where k_{aj} : proportional j-motor-torque constant [N · m / A]; v_{aj} : armature voltage [V]; r_{aj} : j-motor armature resistance [Ω]; $x_{j+6} = \tau_j$: torque generated by the j-motor axis [N · m]; k_{bj} : j-motor proportionality constant [V · rad / s]; L_{aj} : j-motor armature inductance [H]; with $j = 1, 2, 3$.

4.3 State Equations Model

The torque generated in the *j*-motor axis is equal to the sum of the *j*-motor and its load, *i.e.*:

$$\tau_j(t) = J_{mj}(t)\cdot\ddot{\theta}_{mj} + \tau_{Lj}^{*}(t) + T_{fmj}(\ddot{\theta}_{mj}) \quad (55)$$

with J_{mj} : *j*-motor inertia moment reflected to *j*-motor axis [N · m · s^2 / rad]; $\ddot{\theta}_{mj}$: *j*-motor angular acceleration referred to *j*-motor axis [rad / s^2]; T_{fmj} : friction torque generated in the *j*-motor axis referred to *j*-motor axis [N·m]; $\dot{\theta}_{mj}$: *j*-motor angular velocity referred to *j*-motor axis [rad /s], with j = 1, 2, 3. From eq. 55:

$$\tau_{L1}^{*} = x_7 - J_{m1}\cdot\dot{x}_2\cdot N_1 - T_{fm1}(x_2\cdot N_1) \quad (56)$$

$$\tau_{L2}^{*} = x_8 - J_{m2}\cdot\dot{x}_4\cdot N_2 - T_{fm2}(x_4\cdot N_2) \quad (57)$$

$$\tau_{L3}^{*} = x_9 - J_{m3}\cdot\dot{x}_6\cdot N_3 - T_{fm3}(x_6\cdot N_3) \quad (58)$$

From eq. 52 to 54, and replacing eq. 56 to 58 in eqs. 46 to 51, the following state equation are obtained:

$$\dot{x}_1 = x_2 \quad (59)$$

$$\dot{x}_2 = f_{14}^{*}\cdot(x_7 - J_{m1}\cdot\dot{x}_2\cdot N_1 - T_{fm1}(x_2\cdot N_1)) + f_{23}^{*} + f_{15}^{*}\cdot(x_8 - J_{m2}\cdot\dot{x}_4\cdot N_2 - T_{fm2}(x_4\cdot N_2)) + f_{24}^{*} + f_{16}^{*}\cdot(x_9 - J_{m3}\cdot\dot{x}_6\cdot N_3 - T_{fm3}(x_6\cdot N_3)) + f_{25}^{*} \quad (60)$$

$$\dot{x}_3 = x_4 \quad (61)$$

$$\dot{x}_4 = f_{17}^{*}\cdot(x_7 - J_{m1}\cdot\dot{x}_2\cdot N_1 - T_{fm1}(x_2\cdot N_1)) + f_{26}^{*} + f_{18}^{*}\cdot(x_8 - J_{m2}\cdot\dot{x}_4\cdot N_2 - T_{fm2}(x_4\cdot N_2)) + f_{27}^{*} + f_{19}^{*}\cdot(x_9 - J_{m3}\cdot\dot{x}_6\cdot N_3 - T_{fm3}(x_6\cdot N_3)) + f_{28}^{*} \quad (62)$$

$$\dot{x}_5 = x_6 \quad (63)$$

$$\dot{x}_6 = f_{20}^{*}\cdot(x_7 - J_{m1}\cdot\dot{x}_2\cdot N_1 - T_{fm1}(x_2\cdot N_1)) + f_{29}^{*} + f_{21}^{*}\cdot(x_8 - J_{m2}\cdot\dot{x}_4\cdot N_2 - T_{fm2}(x_4\cdot N_2)) + f_{30}^{*} + f_{22}^{*}\cdot(x_9 - J_{m3}\cdot\dot{x}_6\cdot N_3 - T_{fm3}(x_6\cdot N_3)) + f_{31}^{*} \quad (64)$$

$$\dot{x}_7 = [k_{a1}\cdot v_{a1}(t) - r_{a1}\cdot x_7 - k_{a1}\cdot k_{b1}(t)\cdot x_2\cdot N_1]/L_{a1} \quad (65)$$

$$\dot{x}_8 = [k_{a2}\cdot v_{a2}(t) - r_{a2}\cdot x_8 - k_{a2}\cdot k_{b2}(t)\cdot x_4\cdot N_2]/L_{a2} \quad (66)$$

$$\dot{x}_9 = [k_{a3}\cdot v_{a3}(t) - r_{a3}\cdot x_9 - k_{a3}\cdot k_{b3}(t)\cdot x_6\cdot N_3]/L_{a3} \quad (67)$$

5 CONTROLLER MODEL

The controller is modelled as:

$$V_{a1}(t) = -k_1\cdot sgn(s_1) \quad (68)$$

$$V_{a2}(t) = -k_2\cdot sgn(s_2) \quad (69)$$

$$V_{a3}(t) = -k_3\cdot sgn(s_3) \quad (70)$$

in which:

$$s_1 = w_1\cdot(x_1 - x_{1d}) + \dot{x}_1 \quad (71)$$

$$s_2 = w_2\cdot(x_2 - x_{3d}) + \dot{x}_2 \quad (72)$$

$$s_3 = w_3\cdot(x_3 - x_{3d}) + \dot{x}_3 \quad (73)$$

where k_j : *j*-discontinuity gain [V]; s_j : *j*-sliding surface [rad/s] ; w_j : *j*-position gain [1/s]; with j = 1, 2, 3.

6 SIMULATIONS RESULTS

In these simulations the specified changes in revolution joint angles are sinusoidal signals:

Table 1: References trajectories.

Joint	Amplitude [rad]	Frequency [rad/s]
x_1	$\pi/3$	$3\pi/5$
x_3	$\pi/4$	π
x_5	1	0

The given initial conditions were $[x_1(0)\ \ x_2(0)\ \ x_3(0)\ \ x_4(0)\ \ x_5(0)\ \ x_6(0)\ \ x_7(0)\ \ x_8(0)\ \ x_9(0)]^T = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$; the required torque to be delivered by the actuators was determined.

When usually neglected nonlinearities are considered, for certain system parameters (see appendix), chaotic behavior was detected in the end-of-arm, just as it is presented in the figures 1 and 2.

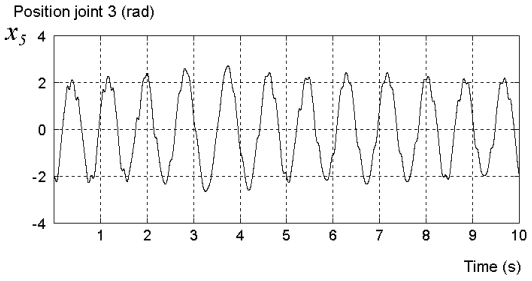


Figure 1: Last link position.

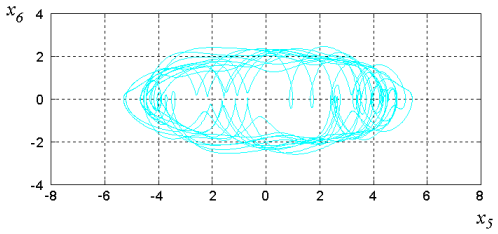


Figure 2: Phase plane (x_5 , $\dot{x}_5$).

For every robot joint, a prescribed path is considered. In $t = 1$ [s] step response for the end-of-arm is imposed; for the first and second link, in $t = 0$ [s] sinusoidal signals are imposed (see figure 3).

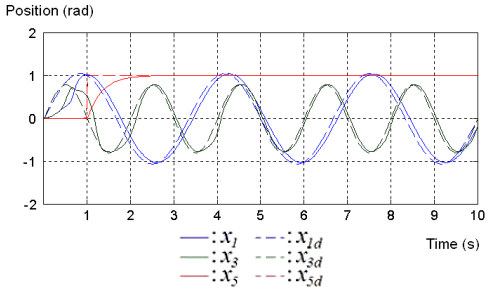


Figure 3: Angular position of the links.

From figure 3, it is possible to appreciate that the chaotic behavior was controlled and the desired paths were tracked.

7 CONCLUSIONS

In this article, models are developed for the actuator and manipulator that address some of the nonlinearities usually neglected in current models.

The manipulator is placed in the same plane of the action of the gravity force and effects such as viscous, static and Coulomb friction in DC motors; viscous, static and Coulomb friction in manipulator joints; actuators and gear trains, are considered in this dynamic model.

The controller design has allowed controlling the detected chaotic behavior.

ACKNOWLEDGEMENTS

This work was possible thanks to the support of DICYT – Universidad de Santiago de Chile, USACH, through Project 060713UO and Project 060713JD.

REFERENCES

Zohdy, M., Fadali, M.S., Liu, J., 1992. Variable Structure Control Using System Decomposition. IEEE Trans. on Automatic Control. 37 (1514-1517)

Park, D., Choi, S., 1999. Moving Sliding Surfaces for High-Order Variable Structure Systems. Int. J. of Control. 72 (960-970)

Bartoszewicz, A., 1995. Sliding Modes for Fast Robot Control. SAMS. 18-19 (539-542)

Xu, J., Lee, T.H., Wang, M., Yu, X., 1996. Design of Variable Structure Controllers with Continuous Switching Control. Int. J. of Control. 65 (409-431)

Lu, X., Spurgeon, S., 1999. Robustness of Static Sliding Mode Control for Non-Linear Systems. Int. J. of Control. 72 (1343-1353)

Chen, C., Xu, R., 1999. Tracking Control of Robot Manipulator Using Sliding Mode Controller with Performance Robustness. Trans. ASME J. Dyn. Syst. Measurement Control. 121 (64-70)

Canudas, C., Aström, K., Braun, K., 1987. Adaptive Friction Compensation in DC-Motor Drives. IEEE J. of Robotics and Automation. RA-3 (681-685)

Kircanski, N., Goldenberg, A., 1997. An Experimental Study of Nonlinear Stiffness, Hysteresis, and Friction Effects in Robot Joints with Harmonic Drives and Torque Sensors. Int. J. of Robotics Research. 16 (214-239)

Urrea, C., 1999. Control de Oscilaciones No Lineales en un Manipulador de Dos Grados de Libertad, Tesis de Magister en Ciencias de la Ingeniería, Universidad de Santiago de Chile.

Leahy, M., Valavanis, K., Saridis, G., 1989. Evaluation of Dynamic Models for PUMA Robot Control. IEEE Trans. on Robotics and Automation. 5 (242-245)

Mahla, I., Urrea, C., 1999. Planar Robot Model Including Driving, Nonlinear Friction and Cubic Stiffness. Proceedings of the IASTED International Conference, Philadelphia.

Craig, J., 1986. Introduction to Robotics: Mechanics and Control. Addison-Wesley.

Hu, J., Dawson, D., 1996. Position Tracking Control for Robot Manipulators Driven by Induction Motors without Flux Measurements. IEEE Trans. on Robotics and Automation. 12, (419-437)

Van Willigenburg, L., Loop, R., 1991. Computation of Time-Optimal Controls Applied to Rigid Manipulators. Int. J. Control. 47 (1097-1117)

Vukobratovic´, M., 1997. The Role of Environment Dynamics in Contact Force Control of Manipulation Robots. Trans. ASME J. Dyn. Syst. Measurement Control. 119 (86-89)

APPENDIX

Simulation Parameters. The following parameter values were taken from (Hu, J. and Dawson, D., 1996), (Van Willigenburg, L. and Loop, R., 1991), (Vukobratovic´, M., 1997).

Motors and their Reduction Gears

Motor M1		Motor M2		Motor M3		
L_{a1}	0.0048	L_{a2}	0.0048	L_{a2}	0.0048	[H]
R_{a1}	1.6	R_{a2}	1.6	R_{a2}	1.6	[Ω]
K_{a1}	0.35	K_{a2}	0.35	K_{a2}	0.35	[N · m / A]
K_{b1}	0.04	K_{b2}	0.04	K_{b2}	0.04	Volts·s / rad]
α_0	260	α_0	260	α_0	260	[N · m]
α_1	1.64	α_1	1.64	α_1	1.64	[N · m]
α_2	0.018	α_2	0.018	α_2	0.018	[N· m· s / rad]
v_0	0.01	v_0	0.01	v_0	0.01	[rad / s]
N_1	62.55	N_2	62.55	N_3	62.55	

Manipulator

Link 1		Link 2		Link 3			
m_1	9.86	m_2	6.38	m_3	3.21	3*	[Kg]
l_1	0.45	l_2	0.5	l_3	0.3		[m]
l_{c1}	0.3	l_{c2}	0.3243	l_{c3}	0.2	0.25*	[m]
I_1	1.1835	I_2	0.1371	I_3	0.0268		[Kg · m]
α_0	100	α_0	100	α_0	100		[N ·m]
α_1	1.01	α_1	1.01	α_1	1.01		[N ·m]
α_2	0.018	α_2	0.018	α_2	0.018		[N·m·s /rad]
v_0	0.01	v_0	0.01	v_0	0.01		[N·m·s /rad]

The manipulator parameter values that generated chaotic behavior were denoted with *. This chaotic behavior was eliminated by the following parameter values that we have proposed for the controllers.

Controllers

1. First Actuator (M1) Controller

k_1	260	[Volts]
w_1	10	[1 / s]

2. Second Actuator (M2) Controller

k_2	100	[Volts]
w_2	20	[1 / s]

3. Third Actuator (M3) Controller

k_3	200	[Volts]
w_3	3	[1 / s]

SHAPE MEMORY ALLOY TENDONS ACTUATED TENTACLE ROBOTIC STRUCTURE
Models and Control

[*]Nicu George Bîzdoacă, [**]Anca Petrişor
[*]*Faculty of Automation,Computers and Electronics, University of Craiova, Romania*
nicu@robotics.ucv.ro, apetrisor@em.ucv.ro

[*]Elvira Bîzdoacă, [*]Ilie Diaconu, [**]Sonia Degeratu
[**]*Faculty of Electromechanical Engineering, University of Craiova, Romania*
diaconu@robotics.ucv.ro, sdegeratu@em.ucv.ro

Keywords: Robotics, Shape memory alloy applications, Serial link, Fuzzy controller.

Abstract: A tentacle manipulator is a manipulator with a great flexibility, with a distributed mass and torque that can take any arbitrary shape. Technologically, such systems can be obtained by using a cellular structure for each element of the arm. Shape memory alloy actuation offers an interesting solution, using the shape transformation of the wire/structure in the moment of applying a thermal type transformation able to offer the martensitic temperature. In order to assure an efficient control of SMA actuator applied to inverted pendulum, a mathematical model and numerical simulation of the resulting model is required. Due a particular possibility SMA actuator connection, a modified dynamics for wire or tendon actuation is presented. For an efficient study a Simulink block set is developed (block for user configurable shape memory alloy material, configurable block for dynamics of single link robotic structure, block for user configurable wire/tendon actuation). As conventional control possibilities were explored, the fuzzy control structure applied in this paper, offer an improved response. A more compact SMA actuation is proposed and experimented. The results are commented.

1 INTRODUCTION

Shape Memory Alloy (SMA) are materials that, once mechanically deformed at given temperature, are able to recover the deformation through an appropriate thermal cycle (Funakubo, 1987).

Between the alloys that show this property, attention has been focused on Nickel – Titanium alloy: it show properties which are suitable for the applications in robotics, general propose actuator and medicine (Faravelli and Marioni, 1996). The nickel titanium alloys, generally refereed to as Nitinol are four times the cost of Cu-Zn-Al alloys, but it possesses several advantages as greater ductility, more recoverable motion, excellent corrosion resistance, stable transformation temperatures, high biocompatibility and the ability to be electrically heated for shape recovery. Other important proprieties of the Nitinol, superelasticity (or pseudoelasticity) refers to the ability of NiTi to return to its original shape upon unloading after a substantial deformation.

This is based on stress-induced martensite formation. The application of an outer stress causes martensite to form at temperatures higher than M_s.

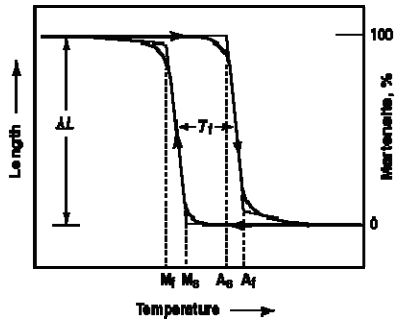


Figure 1: Martensitic and Austentic transformations.

The macroscopic deformation is accommodated by the formation of martensite. When the stress is released, the martensite transforms back into austenite and the specimen returns back to its original shape. Superelastic NiTi can be strained several times more than ordinary metal alloys

without being plastically deformed, which reflects its rubber-like behavior. It is, however, only observed over a specific temperature area. The highest temperature at which martensite can no longer stress induced is called Md. Above M_d NiTi alloy is deformed like ordinary materials by slipping. Below as temperature, the material is martensitic and does not recover. Thus, superelasticity appears in a temperature range from near A_f and up to M_d. The largest ability to recover occurs close to A_f.

Another important feature of superelastic materials is that their unloading curves are flat over large strains. Thus, the force applied by a superelastic device is determined by the temperature, not by the strain as in conventional Hookian materials. The basic rule for electrical actuation is that the temperature of complete transformation to martensite M_f, of the actuator, must be well above the maximum ambient temperature expected.

2 DYNAMICS OF TWO-LINK TENDON-DRIVEN ROBOTIC STRUCTURE

There are many methods for generating the dynamic equations of mechanical system. All methods generate equivalent sets of equations, but different forms of the equations may be better suited for computation different forms of the equations may be better suited for computation or analysis..

Using the kinetic energy and Lagrange methods results:

$$\begin{bmatrix} \alpha+\beta c_2 & \delta+\frac{1}{2}\beta c_2 \\ \delta+\frac{1}{2}\beta c_2 & \delta \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}\beta s_2\dot{\theta}_2 & -\frac{1}{2}\beta s_2(\dot{\theta}_2+\dot{\theta}_1) \\ \frac{1}{2}\beta s_2\dot{\theta}_1 & 0 \end{bmatrix} \bullet \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (1)$$

Where

$$\alpha = \frac{m_1}{12}\left(l_1^2+w_1^2\right)+\frac{m_2}{12}\left(l_2^2+w_2^2\right)+m_1r_1^2+m_2\left(l_1^2+r_2^2\right) \quad (2)$$

$$\beta = m_2 l_1 l_2 \quad (3)$$

$$\delta = \frac{m_2}{12}\left(l_2^2+w_2^2\right)+m_2r_2^2 \quad (4)$$

with w_1, w_2, l_1, l_2 the width and respectively the length of link 1 and link 2.

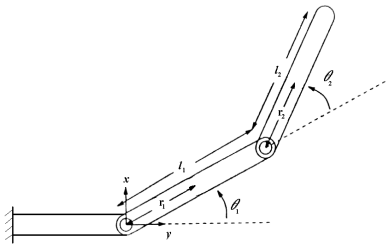


Figure 2: Two link robotic architecture.

3 SHAPE MEMORY ACTUATOR STRUCTURE

Due the actuation architecture a simple mathematical model can be establish. Schematically the shape memory actuation is

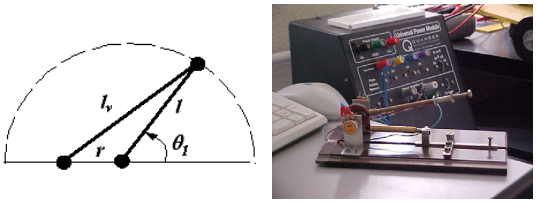


Figure 3: Shape memory alloy actuation structure.

In Figure 3 l_v is the variable length of shape memory alloy wire, the l is the robotic link length between the articulation point and the shape memory alloy wire connection, r is the distance between the second end of the SMA wire (which is a fixed point) and the articulation point of the link (fixed point too).

Using simple mathematical computation the mathematical dependence can be established

$$\theta_1 = \arccos\left(\frac{l_v^2-\left(r^2+l^2\right)}{2lr}\right) \Leftrightarrow \theta_1 = f\left(l_v^2\right) \quad (5)$$

The graphic of θ_1 as function of lv (considering the real domain variation for $\theta_1 \in [0,\pi]$) is linear, that the liniarisation in modeling can be done successfully.

The explanations concern the structural variation of SMA actuator, which are limited superior by l_v and inferior by 0.5 l_v. The mathematical model including the SMA actuation can be developed in two ways: First is possible to consider for position control, ONLY the length variation of the SMA actuator. This approach is a correct one, the

additional torque, provided by the particular proprieties of SMA, enforces the actuation. The situation corresponds to tendon actuation or wire actuation. Using the substitution:

$$\dot{\theta}_1 = \frac{-2l_v}{lr\sqrt{4-\left(\frac{l_v^2-l^2-r^2}{lr}\right)^2}}\dot{l}_v \tag{6}$$

$$\ddot{\theta}_1 = \frac{-2l_v}{lr\sqrt{4-\left(\frac{l_v^2-l^2-r^2}{lr}\right)^2}}\ddot{l}_v - \frac{2}{lr\sqrt{4-\left(\frac{l_v^2-l^2-r^2}{lr}\right)^2}}\dot{l}_v^2 - \frac{4l_v^2\left(l_v^2-l^2-r^2\right)}{l^3r^3\sqrt{\left(4-\left(\frac{l_v^2-l^2-r^2}{lr}\right)^2\right)^3}}\dot{l}_v^2 \tag{7}$$

Analyzing the equilibrium conditions, results that $\tau_1 = b_1(\theta_1)$ and $l_v^2 = r^2 + l^2$, state which correspond to real case.

Second way makes a simplifying assumption: because the SMA connection with single link structure can be choose near to the articulation point, we can assume that the entire SMA torque is directly used for movement. Then the mathematical model can be expressed as

$$\tau_{SMA} = \left(\frac{m_1 w_1^2}{3}\right)\ddot{\theta}_1 + \frac{g m_1 w_1 \cos(\theta_1)}{2} + b_1(\theta_1) \tag{8}$$

4 CONTROL OF SHAPE MEMORY ALLOY TENTACLE ROBOTIC STRUCTURE

In order to investigate the SMA robotic structure comportment a Quanser modified platform was used for experiments. The basic control structure uses a configurable PID controller and a Quanser Power Module Unit for energizing the SMA actuators.

In order to investigate the SMA robotic structure comportment a Quanser modified platform was used for experiments. The basic control structure uses a configurable PID controller and a Quanser Power Module Unit for energizing the SMA actuators.

PID controller was changed, in order to adapt to the particularities of the SMA actuator. A negative command for SMA actuator corresponds to a cooling source. The actual structure use for cooling only the ambient temperature.

The best results arise when a PI controller is used. The PI experimented controller parameters are: the proportional parameter K_R=10 and the integration parameter is K_I =0, 05.

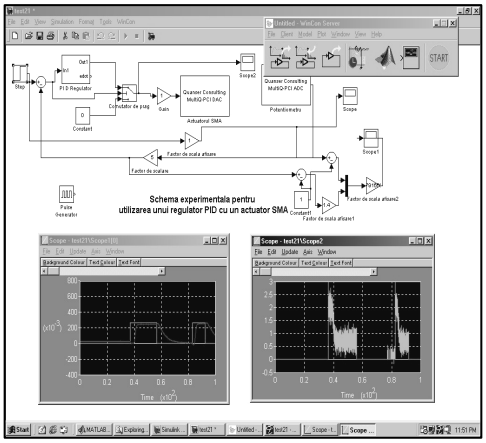


Figure 4: Quanser modified platform.

The input step is equivalently with 30^0 angle base variation and the evolution of this reference is represented with the response of real system in Figure 5. The control signal variation is presented in Figure 6.

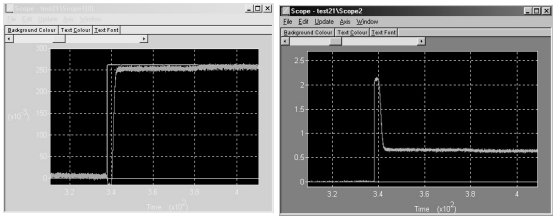


Figure 5: System response, for step input.

Figure 6: PI controller response, for step input.

For negative step, the evolution of the system and the control variable evolution are presented in Figure 7 and Figure 8.

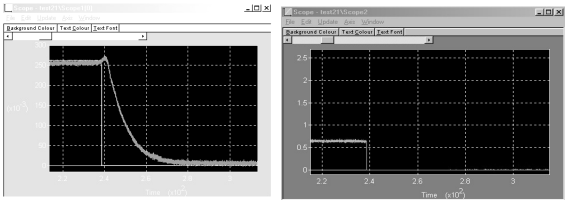


Figure 7: System response, negative step input.

Figure 8: PI controller response, negative step input.

Using PID, PD controller the experiments conduct to less convenient results from the point of view of time response or controller dynamics.

Using heat in order to activate SMA wire, a human operator will increase or decrease the amount of heat in order to assure a desired position to robotic link. Because of medium temperature influence, can not be establish, apriori, a clear control law, available for all the points of the robotic structure workspace. A simple and efficent control structure can be implemented.

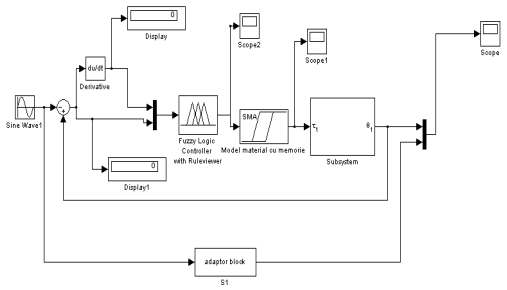


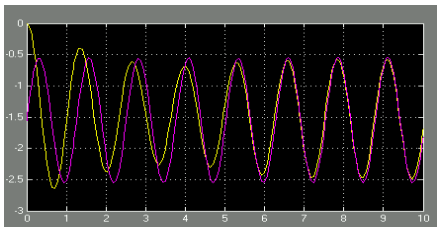
Figure 9: Fuzzy control structure.

For an efficient control it is proposed the following definition for input and output members:
- input 1 is the first derivate of pozition error, with 3 fuzzy member: Negative, Zero, Pozitive
- input 2 is pozition error with 3 fuzzy member: Negative, Zero and Pozitive
- output is temperature heating with 3 fuzzy member: Temperature Negative (temperature under austenitic start transformation), Temperature Zero (temperatures between start and final austenitic transformation), Temperature Positive (temperature above temperature of final austenitic transformation).

Table 1: Fuzzy rules for the proposed controller.

$\dot{e}$ \ e	P	Z	N
P	TP	TP	TP
Z	TZ	TZ	TZ
N	TN	TN	TN

The result of the numerical simulation are



promising, related to the simplicity of the control structure, for the case of the sinusoidal reference with frecvency of 5 rad/sec.

Figure 10: Fuzzy robotic structure output evolution.

5 CONCLUSIONS

The simulations, the mathematical model and the initial experiments developed in the article offer a background in studying the serial link robotic control possibilities. The results respect the real evolution of the structure. In the future, the authors will explore improvement of the control performnces and the extension of the experiments to n link robotic structure.

REFERENCES

Cheng, F. T., "Control and Simulation for a Closed Chain Dual Redundant Manipulator System", Journal of Robotic Systems, pp. 119 - 133, 1995

Cheng, F. T., Orin,D. E., "Optimal Force Distribution in Multiple-Chain Robotic Systems", IEEE Trans. on Sys. Man and Cyb.", Jan., 1991, vol. 21, pp. 13 – 24

Cheng, F. T., Orin,D. E., , "Efficient Formulation of the Force Distribution Equations for Simple Closed - Chain Robotic Mechanisms", IEEE Trans on Sys. Man and Cyb.,Jan. 1991, vol. 21, pp. 25 -32.

Delay, L., Chandrasekaran M., 1987. Les Editions Physique. Les Ulis.

Faravelli L and Marioni A, 1996, Exploiting SMA Bars in Energy Dissipators, Proceedings of the 2nd International Workshop on Structural Control, Hong Kong HKUST 41-50

Funakubo H., 1987, Shape Memory Alloys, Gordon and Breach Science Publishers

Ivanescu, M., Dynamic Control for a Tentacle Manipulator, Proc. of Int. Conf., Charlotte, USA, 1984

Ivanescu, M.,Stoian,V., A Variable Structure Controller for a Tentacle Manipulator, Proc. of the 1995 IEEE Int.Conf. on Robotics and Aut., Nagoya, Japan, May 21 - 27,1995, vol. 3, pp. 3155 - 3160

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

Mason, M. T., "Compliance and Force Control", IEEE Trans. Sys. Man Cyb., Nr. 6, 1981, pp. 418 – 432

Ross, T.J., Fuzzy Logic with Engineering Applications, Mc.Grow Hill , Inc., 1995

Soo Yeong Yi, A robust Fuzzy Logic Controller for Robot Manipulators, IEEE Trans. on Systems, Man and Cybernetics, vol 27, No 4, 706-713, 1997

Tao, C.W. ,Design of Fuzzy-Learning Fuzzy Controllers , FUZZ IEEE'98, 416-421

Utkin, V. I., Variable structure systems with sliding modes, IEEE Trans. Automat. Contr., vol. AC-22, pp. 212-222, 1977.

Utkin, V. I., Variable structure systems and sliding mode\—State of the art assessment, Variable Structure Control for Robotics and Aerospace Applications, K. D. Young, Ed. , New York: Elsevier, pp. 9-32, 1993.

TELECONTROL PLATFORM
Telecontrol Platform for Industrial Installations

Eduardo J. Moya, Oscar Calvo, José María Pérez, José Ramón Janeiro and David García
Fundación CARTIF, Parque Tecnológico de Boecillo, Parcela 205, 47151 Boecillo, Valladolid, Spain
edumoy@cartif.es, osccal@cartif.es, joslar@cartif.es, josjan@cartif.es, davgar@cartif.es

Keywords: GSM modem, analogical modem, PLC, SCADA, monitoring, Industrial Process.

Abstract: This article explains the telecontrol platform for industrial installations developed by CARTIF Foundation. Using this system it will be able to send control orders and receive notification of alarms from the PLC thanks to SMS (Short Messages System) messages which use GSM technology. In case of requiring a greater flow of data it will use telephone line combined with MODBUS protocol. All this will enable us to monitor and control any industrial installation with a very low cost.

1 INTRODUCTION

The possibility of remote connections with industrial processes can represent significant savings of time and money for companies. In fact, you can control and monitor equipments, update the software or locate faults regardless of equipment location with a simple remote connection.

These monitoring tasks can be carried out with conventional technologies, as GSM network or switched telephone network, (International Engineering Consortium, 2007) which have been installed for many years. These systems are reappearing in industrial applications thanks to its low cost and the broad range of possibilities offered

By using of these technologies (GeneralLynx, 2007), CARTIF Foundation has developed a monitoring system based on GSM modems and analogical ones that allow us to monitor and modify variables of processes controlled by SIEMENS S7-200 automatisms.

The article is organized as follows. In Section 2 we explain the first part of this project that consisted in creating a system capable of controlling and monitoring a process through SMS messaging. In Section 3 we expose the second part of this project. In this case we use an analogical modem to link PLC and SCADA. In Section 4 we will explain a practical case of the combined use of the library SMS and communication via RTC modem. Finally conclusions and open issues for future research are discussed in Section 5.

2 MONITORING OF PROCESSES USING GSM TECHNOLOGY

In the case we're dealing with we'll use the GSM network, which will allow us to send simple control commands between a cellular and a PLC from any place as long as we have enough coverage; this system requires only a GSM modem and a SIM card which are very cheap.

In this section we'll explain the library called SMS developed by CARTIF Foundation. This library has been developed to be used with a programmable Siemens PLC although similar developments can be performed to be used with other brands as Telemecanique, Omron or Allen-Bradley.

2.1 Elements of the System

CARTIF Foundation uses mainly programmable PLCs of Siemens or Telemecanique. We decided to develop SMS library to be used with a low-mid range Siemens PLC. In this way, the automatism that has been selected is a S7-200. Developing this library for the S7-200, subsequent developments in other PLC´s will be able to be conducted in a very similar way.

Besides that it will be needed a RS 232/PPI Multi-Master cable because the ports of Siemens S7-200 series are RS-485 and GSM modems usually have RS-232.

The third item of the system is a GSM modem.

At first it was wanted to develop a library that would be independent of the brand and model of GSM modem, but it wasn't possible because each brand have different responses to the commands that are sent to it. In order to simplify the design of the library it will be used a generic one, GSM/GPRS Wavecom Fastrack (Fastrack Modem M1306B, 2007).

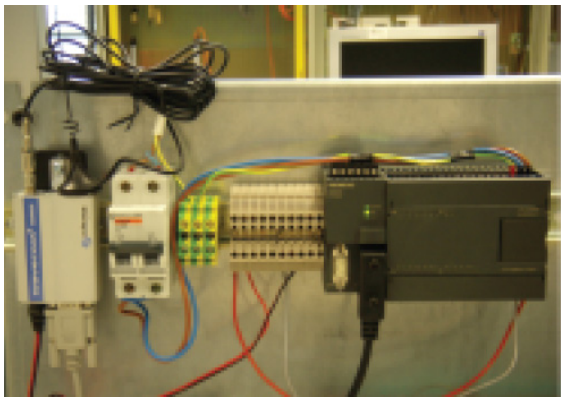


Figure 1: PLC & GSM modem.

2.2 Characteristics of SMS Library

SMS library has as basic functions the reception and sending SMS messages, always besides their treatment. The great advantage is that communication with modems is standardized through the use of AT commands (AT Commands Interface Guide, 2007). AT commands are just coded instructions for communication between a device and a modem. As we said before although the instructions that are sent to modems are standardized through the AT commands, the modems responses are not. This causes the library not to be valid for all GSM modems, although its adaptation to other models is very simple.

2.2.1 AT Commands

The commands AT used in the library are: ATE0, AT+CPIN ?, AT+CPIN = "Nº PIN", AT+CMGF = 1, AT+CSMP = 17,167,0,0, AT+CREG ?, AT+CPMS = "SM", AT+CMGS = "PHONE NUMBER", AT+CMGR = X, AT+CMGD = X.

2.2.2 Library Functions

The library functions are:

- Sending SMS.
- Periodical control of coverage.
- Output of error.
- Automatic blocking in case of entering a wrong PIN.
- Reception of SMS
- Elimination SPAM.
- Automatic Clearing of the read SMS.
- Permits of access.
- Treatment automatic SMS.
- Treatment of SMS by the user.
- Size in program memory: 4.5Kb
- Size in data memory: 490 bytes.

2.3 Compatible SMS

For the use of library SMS is only necessary to have a mobile phone (Moya, 2007). You can send SMS of two types:

2.3.1 SPECIFIC Messages

These messages are customized for an installation in concrete. In this case, the library SMS returns the text message received and is the programmer of PLC who is responsible to deal with it. These messages have the following format:

```
"PERS" "STOP MOTOR 1"
```

In addition, the programmer of PLC can send SMS of notification of alarms or as a response or acknowledgement to an SMS received, being the format of these messages to choice of the controller.

2.3.2 STANDARD Messages

These messages are called standard because they are not specific of an installation. They are treated directly by the library SMS and in the event that it will have configured, send an acknowledgement of receipt.

This kind of message has the following format in the case of messages of modification of variables:

```
"ESTA""SYMBOL""ADDRESS""OPERATION"
```

or in case of messages of consultation

```
"ESTA" "SYMBOL" "ADDRESS" "?"
```

Once received a message of consultation, the response is sent automatically, and in the case of modifications of variables is sent an acknowledgement if it has been configured.

In this kind of messages, "SYMBOL" corresponds with the type of variable (V, VB, VW, VD and VR). The "ADDRESS" corresponds with the memory address (1000.1). The "OPERATION"

field corresponds with the value to write in the direction of memory and in the case of "?" means that it is a consultation of the value of the data located in this address.

Table 1: Operations performed by SMS.

DATA	OPERATION	DESCRIPTION
V	1	Force a bit to be ON
	0	Force a bit to be OFF
	?	Ask about a bit status
VB	VALUE	Change a byte
	?	Ask about a byte value
VW	VALUE	Change a word
	?	Ask about a word value
VD	VALUE	Change a integer
	?	Ask about a integer value
VR	VALUE	Change a real
	?	Ask about a real value

2.4 Program Flow

As is shown in the flow diagram (figure 2), firstly a start stage is done in which, among other things, it's checked whether the PIN code is entered (if it's not, it will be entered) and the existence of coverage is verified.

The following step is to check if there are messages in the SIM card. In the case that there are any, the message is downloaded from the SIM card to a reserved area in the PLC. In the case that there are not, it would go to the stage for sending SMS from which it would send a message in the case that the user program required so. In the next step it's checked whether the SMS sender's telephone has permission to access to the control and modification of variables. This eliminates directly SPAM and telephones without access permission.

In the next step it's verified that the text message has not been read in previous cycles. In the case that the telephone does not have permission to access or the SMS has been read, the message is deleted. In the case that the SMS has not been deleted in the previous stage, it is checked whether the type of message is: "ESTANDAR", "SPECIFIC" or none of the two types. In the last case it is removed.

When it is received a "ESTA" SMS, it is dealed by the library SMS whether it is a monitoring message or if it is a control message.

If it's a message "PERS" type it is returned by the subroutine SMS to the main programme and it is the programmer the responsible for its dealing.

Then, once dealed, the SMS is removed from the GSM modem's SIM memory. In the next step SMS's are sent both of the acknowledgement, responses to consultations or notification of alarms.

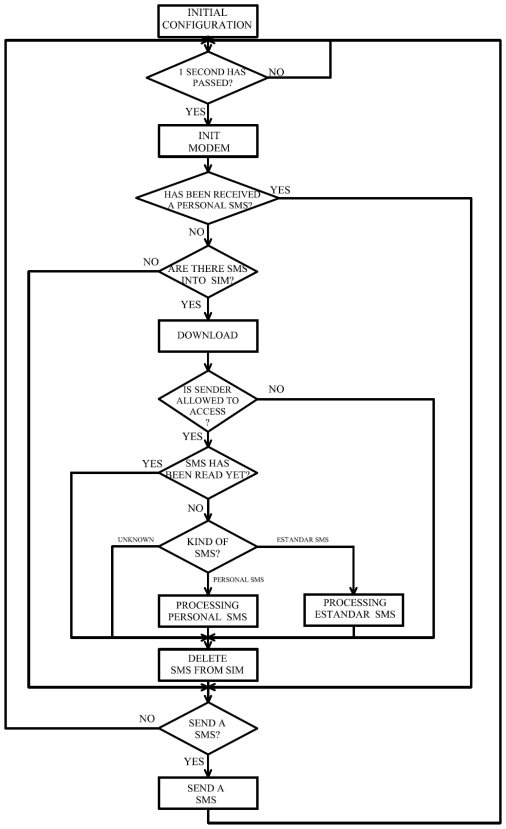


Figure 2: SMS program flow.

3 SUPERVISION & CONTROL OF PROCESS THROUGH ANALOGICAL MODEMS

In recent decades, the improvement of communication systems has caused a change in the form that the society sees the world. These improvements have narrowed the gap between the different points of the planet so any event can be known anywhere almost instantaneously.

Communications via telephone line and its application to computer systems through the modems have largely contributed to this.

The great coverage of the conventional telephone network enables an almost immediate connection between two computers if modems are used, and if it's extended to the industrial area it will provide us a cheap and effective method of controlling a process without the need for large disbursements.

3.1 Items of the System

3.1.1 PLC

Like for communication via GSM the PLC that we'll use will be a S7-200, although once set the foundations of the system, it can be carried out in other brands's PLC by introducing small changes.

3.1.2 Analogical MODEM

The Modem chosen is a module for expansion of S7-200. It does not need any library and its configuration is very simple, thanks to the assistant of the PLC's programming tool. It works in slave mode, and it uses MODBUS protocol to communicate with the PC, which will play a master role in our supervision system (Jiménez, 2007).

3.1.3 Personal Computer

The master of the system is a PC. This has a Modem to communicate with the Modem EM241 via the telephone line. To treat the data being received, a software application in Visual Basic has been scheduled, which transforms the MODBUS strings characters in data that can be displayed in a SCADA also scheduled in Visual Basic (Janeiro, 2006).

3.2 System Characteristics

3.2.1 Description of Modbus RTU Protocol

Once the connection between the modem local modem and remote modem connected to the PLC is established, we must choose a protocol that helps us to exchange data between the PC (MASTER) and the PLC that controls our process (SLAVE).

In our case the protocol which we'll use will be MODBUS RTU, very used in the industry for communications via modem.

The controllers communicate by means of a master-slave technique, in which only one device (master) may start transactions. The other devices (slaves) respond by supplying the master the data requested, or carrying out the action requested in the petition. Among the master devices typical central processors and programming panels are included. Typical slaves are the PLC's.

3.2.2 The Query-Response Cycle

Query: The function code in the petition indicates the slave device directing the type of action to perform. The bytes of data contain any additional information that the slave will need to carry out the function. The data field must contain the information to indicate the slave in what registration it should begin and how many has to read. The error verification field provides a method for the slave to validate the integrity of the contents of the received message.

Response: If the slave develops a normal response, the function code content in the response is a replica of the function code sent in the petition. The bytes of data contain data collected by the slave, such as values of registers or states. If an error occurs, the function code content in the answer is different from the function code sent in the petition, to indicate that the answer is a response of error and the bytes of data contain a code that describes the error. The verification of error field allows the master to confirm that the contents of the message are valid.

3.2.3 Queries Implemented by the Application

Modbus is a protocol developed by Modicon for its range of PLCs. Siemens, particularly S7-200, has implemented libraries, which introduced in the program code, allow to use it. In our case, the EM241 module has them included, for what it's not necessary to modify anything in the PLC's programme.

Table 2: Modbus operations implemented by SIEMENS.

FUNCTION	DESCRIPTION
1	Read coil status
2	Read input status
3	Read holding registers
4	Read input registers
5	Force single coil
6	Preset single register
15	Force multiple coils
16	Preset multiple registers

Despite that Modbus incorporates a wide variety of functions, SIEMENS only has 8 implemented which are those indicated in the table 2.

These operations act on bits in the case of inputs and outputs, or in words if it is variables in PLC memory. In the case of desiring to use different variable sizes, as double words, bits of PLC memory, etc, it must be done from these functions.

The application which has been developed, has taken into account this problem and has been programmed to allow the following requests:

- Read and write digital outputs.
- Read analogical inputs.

- Read and write integers.
- Read and write double integers.
- Read and write real numbers.
- Read and write bits from V memory.

The PLC responds by strings MODBUS RTU in hexadecimal that must be decoded.

3.2.4 AT Commands used

As in the case of communication using a GSM link, it has been needed AT commandsm: AT, ATE0, ATS30=time (s), ATDnumber, ATH.

3.2.5 Application Flow

Before establishing any communication it must be verified that the configuration of PC-Modem is correct. If this is so it can proceed to dial the telephone number of the modem connected to the PLC. In the case of being the line occupied or not to establish the connection it will generate a message informing about what is happening.

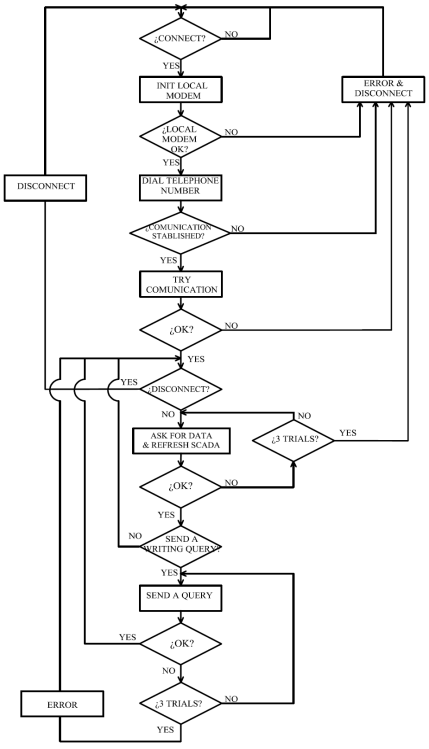


Figure 3: Program flow.

After connecting with the PLC test strings will be sent to see check that the communication is good. If everything is OK application will send petitions to the PLC with the frequency of refresh assigned by the user. The times of refresh can vary, from 2 seconds. Application must send petitions to the PLC, receive the responses from it, decode them and display in a screen through the SCADA. In addition to this, it will have to detect the communication errors that may occur.

On the other hand, it also takes into account the possible delays that may result from a wrong function of the modem, temporary disconnections of the line, etc. If these failure times are excessive, as in the previous case, the Modem gets disconnect and the cause is notified. In the case of wanting to change a variable of the PLC, the SCADA refreshing cycle is interrupted and the petition sent. If the operation is carried out successfully it returns to the routine of variable reading.

4 PRACTICAL APPLICATION

Now we will explain a practical case of the combined use of the library SMS and communication via RTC modem.

The case that we explain is a wheat storage and dosage plan. The process is divided into two parts: the part storage of wheat (Figure 4) consists of the bunghole and the first of two silos, while the part dosage of wheat (Figure 5) includes the last five silos and a weight scale.

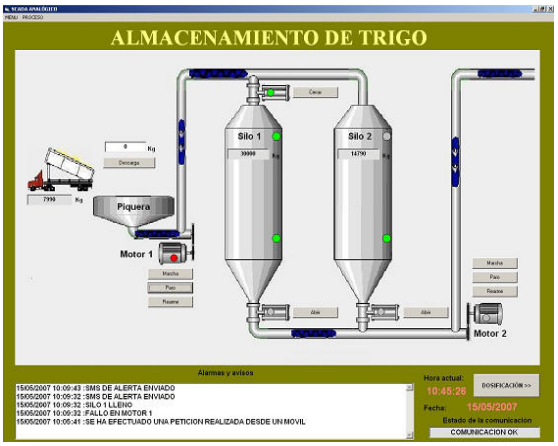


Figure 4: Storage of wheat.

The download of wheat from the truck is carried out in the bunghole. To transfer the burden of wheat from the entry deposit to one of the two storage silos (Silo 1 or 2) the engine on the left must be on to activate the endless screw and the buckets elevators (horizontal and vertical displacement of wheat).

The wheat shall be deposited in Silo 1 or Silo 2, depending on whether the upper chopping block in Silo 1, is open or not. The wheat flow considered

when this transfer performed is 10 kg/s.

The process of dosage consists on four silos (Silo 3, 4, 5 or 6) where different types of wheat are stored and on silo 7, where the mixture composed by wheats form Silos 3, 4, 5 or 6 is obtained. The maximum capacity of all these silos is 10.000 Kg. The flow of endless screws and the buckets elevators is as maximum 50 kg/s. The wheat supply in Silos 3, 4, 5 or 6 comes from storage Silos 1 and 2 of. The operator will be responsible for selecting the destination of wheat from Silos 1 and 2 by opening/closing the different upper chopping blocks of Silos 3, 4, 5 or 6.

Figure 5: Dosage of wheat.

Apart from "ESTANDAR" SMS configured for this installation, the following "SPECIFIC" messages have been programmed:

Control: It can activate the dosage and stop it.

Consultations: It can make consultations such as the weight of different silos, of the bunghole or of the dosage. It can ask about the state of the engines, as well as the state of the chopping blocks.

Maintenance: The alarm notifications have been programmed: unloading a truck in the bunghole, failure of any of the engines, filling of silos, and failure in the dosage and notices that the dosage has been completed.

To display the status of the process and the changes that we are doing via SMS we use the SCADA scheduled in Visual Basic, which uses the RTC line to establish the communication with the PLC.

5 CONCLUSIONS

The aim of this project was to make a system for supervising processes that allowed remote control of any installation in a simple and safe way but without incurring big costs.

This platform has been focused to be used in mid-range automatons such as the S7-200, because this PLC is the most indicated to control the processes that can be supervised by this system.

It has been tried to deal with the issue from two fronts, through a wireless communication by GSM Modems and on the other hand, a communication through telephone line. The choice of one or another depends on different factors such as: the location and accessibility of the plant, means of communication, level of automation, process complexity, etc.

In general, it can be said that for all activities that require an important exchange of data or/and a constant supervision, it would be advisable the implementation of an analogical modem communication system.

In the future is planned to develop this platform with other PLC brands like Telemecanique and Omron.

ACKNOWLEDGEMENTS

This work was supported in part by “Programa de Fomento de la Investigación Técnica para los Centros Tecnológicos”, (PROFIT grant FIT 330220-2005-138) from the Spanish Education and Culture Ministry.

REFERENCES

AT Commands Interface Guide, *<http:// www.rfsolutions.co.uk>,* (in June 6, 2007).

Fastrack Modem M1306B User Guide, *<http:// www.omniinstruments.co.uk>*, (in June 6, 2007)

GeneraLynx. *Remote supervision and control by WAP, <http://www.euroines.com/down/DemoDoc/WapScada%20DD.pdf>*, (in January 10, 2007)

International Engineering Consortium. Global System for Mobile Communication*, <http:www.iec.org/online/tutorials/gsm/>*, (in January 10, 2007)

Janeiro, J. R., 2006. *Supervisión remota de procesos industriales controlados por Autómatas Programables*, University of Valladolid. Spain, 1nd edition.

Jiménez, M., *Comunicaciones Industriales, Protocolo Modbus*, *<http:www.dte.upct.es>*, (in June 6, 2007).

Moya, E., 2007. *Control y mantenimiento de instalaciones remotas*. Wireless. Automática e Instrumentación, 384, pp. 44-47.

AN APPROACH TO OBTAIN A PLC PROGRAM FROM A DEVS MODEL

Hyeong T. Park, Kil Y. Seong, Suraj Dangol, Gi N. Wang and Sang C. Park
Department of Industrial Information & Systems Engineering, Ajou University, Korea
{taiji416, skyblue, suraj, gnwang, scpark}@ajou.ac.kr

Keywords: Programmable Logic Controller(PLC), DEVS, Factory Automation, Simulation.

Abstract: Proposed in the paper is an approach to generate the PLC code from the Discrete Event System Specification (DEVS) model. DEVS have been widely accepted to model the real system for the discrete event system simulation. The objective of this paper is to generate PLC control code from the DEVS model. To achieve it, this paper proposes two steps. First step is to convert the real system into the virtual model using the 'three-phase-modeling procedure'. In the second step, the obtained model is formalized with DEVS formalism. The final model consists of different components, among them the State manager and the Flow controller model plays vital role to generate PLC code. In this paper, proposed steps are described with a work cell example.

1 INTRODUCTION

To survive and prosper in the modern manufacturing era, a manufacturing company should be capable of adapting reduced life cycle of products in a continuously changing market place. Simulation is a useful tool for manufacturers to adapt this kind of rapidly changing market to design and analyze complex systems that are difficult to model analytically or mathematically (Choi, 2000). Manufacturers who are using simulation can reduce time to reach stable state of automated manufacturing process by utilizing statistics, finding bottlenecks, pointing out scheduling error etc... For the simulation of manufacturing systems, manufacturers have been using various simulation languages, simulation software for example ARENA, AutoMod. Most of traditional simulation languages and softwares focus on the representation of independent entity flows between processes; their method is commonly referenced to as a transaction-oriented approach. In this paper, we propose an object-oriented approach that is based on the set of object classes capable of modeling a behavior of existing system components.

The object-oriented modeling (OOM) is a modeling paradigm, that uses real world objects for modeling and builds language independent design organized around those objects (Rumbaugh, 1991). Even though OOM has been widely known to be an effective method for modeling complicated software systems, very few researchers tried to apply the OOM to design and simulate manufacturing system software models. Based on the OOM paradigm, different researchers have proposed various modeling approaches despite the fact that they express them in different ways with different notations. For example, Choi et al. presented the JR-net framework for modeling which is based on the OOM paradigm of Rumbaugh et al., which is made of three sub-models(an object model, functional model, and dynamic model). Chen and Lu proposed an object-oriented modeling methodology to model production systems in terms of the Petri-nets, the entity relationship diagram (ERD) and the IDEF0 (Chen, 1994). Virtual factory (VF) is also very important concept to be considered in today's simulation environment. By using the OOM paradigm, VF concept can be implemented efficiently (Onosato, 1993).

Recently, Park (Park, 2005) proposed a 'three-phase-modeling framework' for creating a virtual model for an automated manufacturing system. This paper employs the three-phase-modeling framework of creating a virtual model, and the Discrete Event System Specification(DEVS) (Zeigler, 1984) for process modeling. The proposed virtual model consists of four types of objects. The virtual device model represents the static layout of devices. This can be decomposed into the shell and core, which

encourages the reusability making possible to adapt different system configurations. For the fidelity of the virtual model, The Transfer handler model handles a set of device-level command that mimics the physical mechanism of a transfer. The Flow controller model decides the firable transfers based on decision variables that are determined by the State manager model. The State manager model and Flow controller model can be converted to PLC part. After finishing the process modeling by employing the three-phase-modeling framework, those two models will be the control information for the converting to PLC.

The overall structure of the paper is as follows. Section 2 represents the brief explanation about the PLC, and Section 3 is about the DEVS. The overall approach to create manufacturing system model for generation PLC code is described in Section 4. Section 5 gives as example cell, which is observed to find correlation between the PLC code and the DEVS model in Section 6. Finally, Conclusion and discussion is addressed in Section 7.

2 PROGRAMMABLE LOGIC CONTROLLER (PLC)

The Programmable Logic Controller (PLC) is an industrial computer used to control automated processes in manufacturing (Parr, 1999). PLC is designed for multiple inputs and outputs arrangements, it detects process state data through the sensing devices such as limit sensors, proximity sensors or signals from the robots executes logics in its memory and triggers the next command through the actuator such as motor, solenoid valve or command signal for the robots etc. PLC executes the control logic programmed in different types of languages. IEC published IEC 61131-3 to standardize PLC languages including Ladder diagram, Sequential Function Chart, Structured Text and Function Block Diagram (Maslar, 1996).

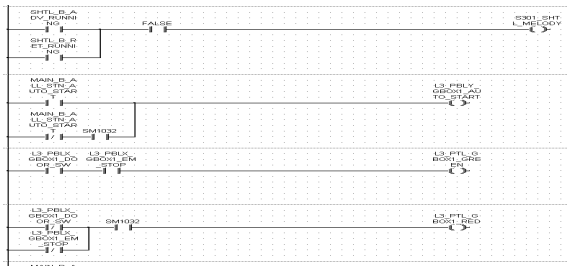


Figure 1: The PLC code in the form of Ladder diagram.

3 DISCRETE EVENT SYSTEM SPECIFICATION (DEVS)

DEVS formalism is introduced by Zeigler, which is a theoretic formalism and it supplies a means of modeling discrete event system in a modular, hierarchical way. With this DEVS formalism, we can perform modeling more easily and correctly by dividing large system into segment models and define the coupling between them. Formally, an atomic model **M** is specified by a 7-tuple:

$M = < X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a >$

X : input events set;

S : sequential states set;

Y : output events set;

δ_{int} : S→S : internal transition function;

δ_{ext}: Q x X → S : external transition function

$Q = \{ (s, e) | s \in S, \ 0 \le e \le t_a(s) \}$: total state of M;

λ : S->Y : output function;

t_a : S→*Real* : time advance function:

The second form of the model, called a coupled model, indicates how to couple several element models together to form a new and bigger model. Formally, a coupled model **DN** is defined as:

DN = < *X, Y, M,* EIC, EOC, IC, SELECT >

X : input events set;

Y : output events set;

M: set of all component models in DEVS;

EIC ∈ DN.IN x M.IN : external input coupling relation;

EOC ∈ M.OUT x DN.OUT : external output

coupling relation;

IC ∈ M.OUT x M.IN : internal coupling relation;

SELECT : 2^M - ø-> M : tie-breaking selector,

Where the extension .IN and .OUT represent the input ports set and the output ports set of each DEVS models.

4 APPROACH TO CREATE MANUFACTURING SYSTEM MODEL TO GENERATE PLC CODE

To construct the automated process, the factory designers have to consider the overall process layout. After deciding skeletal layout, the process cycle time is simulated by the discrete event system software like ARENA or AutoMod. In this stage, including the process cycle time and production capability, the physical validity and efficiency of co-working machines are also described. Simulation and modeling software QUEST or IGRIP are used for this purpose (Breuss, 2005).

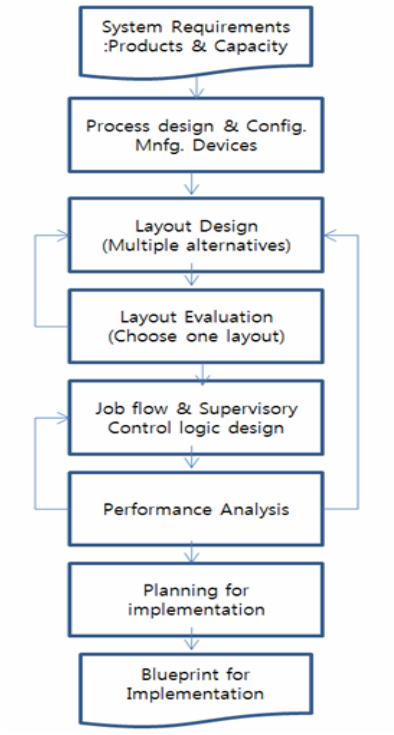


Figure 2: Automated Factory construction procedure.

On the next step, the PLC code programming for logical functioning is done without utilizing information from previous discrete event systems modeling. The gap between the high level simulation of discrete event system and the low level physical process control logic need to be bridged for the utilization of process modeling and practical simulation in terms of physical automated device movement. This paper tries to find the bridge between these two different simulation levels and further describes automatic generation of PLC code from the DEVS model.

In developing the DEVS model, the first thing we have to do is to model the manufacturing system by the three-phase-modeling framework (Park, 2005). The framework describes manufacturing system modeling with 4 components; the Virtual device model, the Transfer handler model, the State manager model and the Flow controller model as shown in Figure 3.

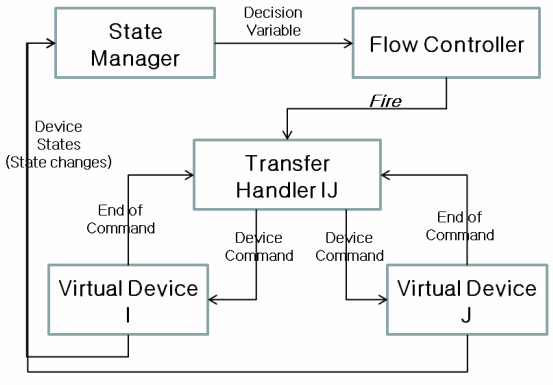


Figure 3: Outline of the virtual manufacturing model.

The Virtual device model shows the manufacturing devices. It has input port to receive the action signal and output port to send the work done signal. The Transfer handler model handles the parts stream and assisting resources (tools and pallets) between devices. This approach focused on the physical mechanism enabling the transfer than conventional approaches. In reality, a transfer happens by the combination of device-level command between co-working devices (giving and taking devices). The State manager model collects the state data of every device. Whenever there is a state change of devices, it will update the device states. Then, this information will be delivered to the Flow controller model as a decision variable. After getting the state information from the State manager model, the Flow controller model will decide firable transfer based on the system state (decision variables).

For the implementation of the virtual manufacturing system model, this paper employs the Discrete Event Systems Specification (DEVS) formalism, which supports the specification of discrete event models in a hierarchical modular manner. The formalism is highly compatible with OOM for simulation. Under the DEVS formalism, we need to specify two types of sub-models: (1) the atomic model, the basic models, from which larger ones are built and (2) the coupled model, how atomic models are related in a hierarchical manner.

When the DEVS model is developed, both the State manager atomic model for the process monitoring and the Flow controller atomic model for the actual control can be replaced the PLC part. Namely, control part for the manufacturing cell. Here is the goal of this paper.

5 DEVS MODELING OF A SIMPLE CELL BASED ON THE THREE-PHASE-MODELING FRAMEWORK

In this Chapter, we will observe a small work cell example. The work cell is modeled according to the three-phase-modeling framework and converted to the DEVS model like mentioned above. Finally, we will compare the DEVS model and the PLC code to find some meaningful bridge.

Figure 4 shows the small cell example. At first, an entity is generated from the Stack, which will lay on the AGV machine in P1, then AGV senses this raw part and moves to the P2 for machining. When machine detects the part arrival by the AGV, the machine starts to operate.

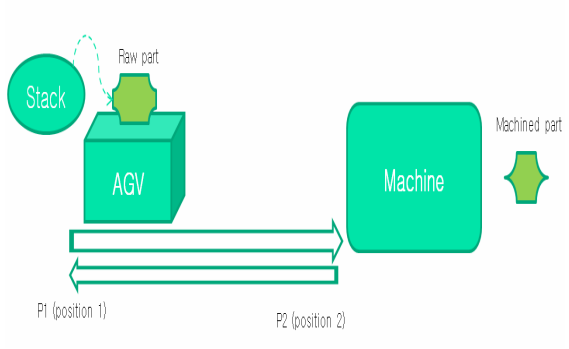


Figure 4: Example cell.

When we consider this example cell in terms of the three-phase-modeling framework, there are three virtual device models; the stack model, the AGV model and the machine model. The stack model generates the raw part entity and places it on the AGV for transfer. Until this point, the entity transfer process is between the stack and the AGV virtual device model as a result the transfer handler model is created between the stack the AGV model. Similarly, entity transferring between the AGV model and the Machine happens. This transfer handling model can be represented as *THam*. If there is any state change among the virtual devices, the changes are supposed to be reported to the State manager model. The State manager model maintains the decision variables in compliance with the reported state changes of the virtual devices and the Flow controller model will make a decision on firable transfer based on the decision variables. Figure 5 represents the constructed model about the example cell.

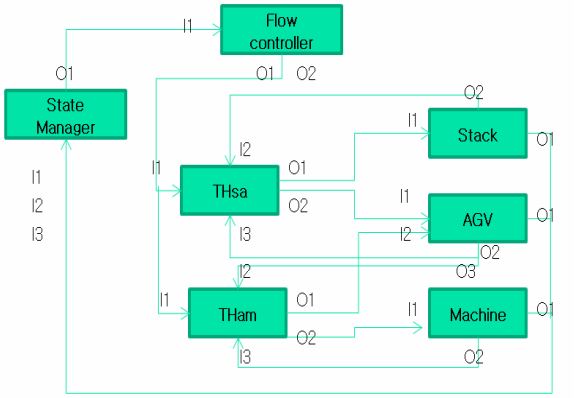
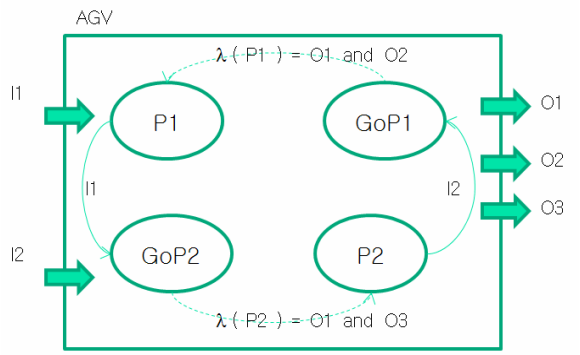


Figure 5: Modeling of the example cell in the Park's methodology.

Once the modeling by means of the three-phase-modeling framework is finished, second step is to convert the model to the DEVS formalism. In this example, every model is converted to the atomic model and entire cell will be the coupled model that is consist of all atomic models. Figure 6 is the converted DEVS model example of AGV. In the traditional implementation of discrete event system simulation using DEVS, DEVSIM++ is a simulation framework which realizes the DEVS formalism for modeling and related abstract simulator concepts for simulation, all in C++ (Kim, 1994). Through this open source frame, we can develop the discrete event system simulation engine easily. Once, both the DEVS implementation and the simulation with PLC control logic is done, we can achieve the overall physical control simulator for automated process.



AGV atomic model

X = { I1, I2 }

Y = { O1, O2, O3 }

S = { P1, GoP2, P2, GoP1 }

δ_{int} :

δ_{int} (GoP2, MT) = P2,

δ_{int} (GoP1, MT) = P1

δ_{ext} :

δ_{ext} (P1, I2) = GoP2, δ_{ext} (P2, I1) = GoP1

λ (P1) = O1 and O2, λ (P2) = O1 and O3

, λ (GoP2) = O1, λ (GoP1) = O1

ta(MT) : Moving Time

Figure 6: DEVS model of the AGV.

6 CORRELATION BETWEEN THE PLC CODE AND THE DEVS MODELS

For the auto generation of PLC code from the DEVS model, we need to examine the PLC code of example cell and the DEVS models, especially the State manager and the Flow controller model.

In the manufacturing unit, PLC collects the process state information through the sensors. These sensor signals are referenced to decide next command or operation. This task is done by the state manager model in the modeled frame. The State manager model detects every change in state of the virtual device and then updates the decision variables. Similar to PLC code, the Flow controller model is supposed to have running logic that is kind of combination of decision variables. As a result, PLC code from the DEVS model can be divided into two parts. One part is for updating the decision variable from the signal of input port in the State manager model. Another is for actual logic composed of decision variables to fulfill the intended process control.

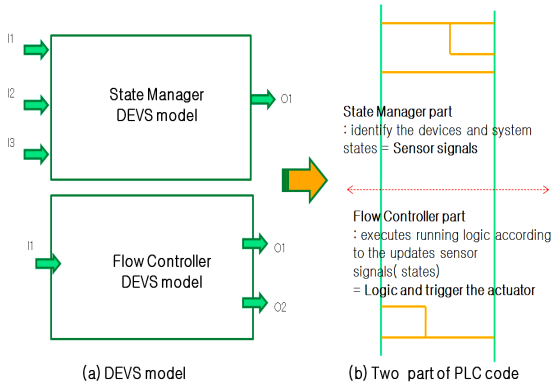


Figure 7: Two part of PLC code.

In the front part, the State manager model collects every state changes through the input port. The one input port of example cell has different kind of signal depend on the state. For example, the input port I2 is the signal from the AGV and it has 4 different kinds of state signals. With the same way, each input port of the State manager model has multiple input signals like shown in Table 1.

Table 1: The States of Atomic models.

	Atomic models	States	Input Signals
I1	Stack	Idle, Release	STACK_IDLE STACK_RELEASE
I2	AGV	P1 GoP2 P2 GoP1	AGV_P1 AGV_GOP2 AGV_P2 AGV_GOP2
I3	Machine	Idle Run	MACHINE_IDLE MACHINE_RUN

The memory structure in the PLC code can be classified into three groups. The first group is input memory which consists of input signal names and the second group is the output memory consisting output signal names and the last is the internal memory which is used to maintain the signal information of input or output and for temporary numerical calculation. The name of input signal can be determined with combination between the input port and its state name. In this way, we can give a name to all input signals.

As mentioned before, the flow controller model reads the decision variables to execute next command. Thus, we have to make decision variables representing the process state as the internal memory. As we did in the input variable for naming, we can give decision variables' name by putting the

'On' between the port name and the state name. Then, this decision variable shows the port's current state is active condition. Once decision variables are set, the Flow controller detects the firable output signals from the set variables. Figure 8 show the decision variables of each input of AGV model and moving condition. To the AGV, the possible condition to move from P1 to P2 is when the raw part is on the AGV, AGV's state is 'GoP2', and the machine state is 'Idle' at the same time.

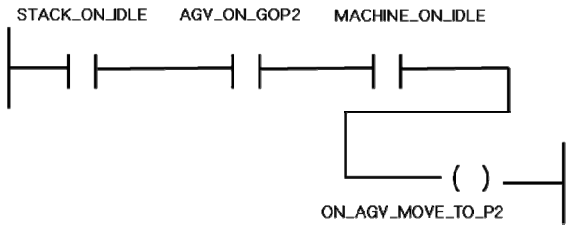


Figure 8: The triggering condition for AGV move.

As we have noticed for the case of the AGV model, the other devices' executing condition can be derived. While the PLC code for the State manager model part can be generated automatically with a combination of decision variables, the flow controller part is sometimes rather ambiguous. That is because unlike the flow controller, DEVS model is quite abstract and high level, the PLC part is very specific control area. Even though, process system designer can construct the DEVS model including low level of PLC, normally DEVS modeling is not fulfilled in this way. This aspect will be limitation or designer's choice in reference to PLC code auto generation. The DEVS modeling here is done specifically in mind of the PLC code generation of the Flow Controller model part. Figure 9 illustrates the two part of PLC code about the AGV from the State manager and the Flow controller model. And the Flow controller DEVS model for PLC code auto generation with the simple work cell is shown in Fig. 10.

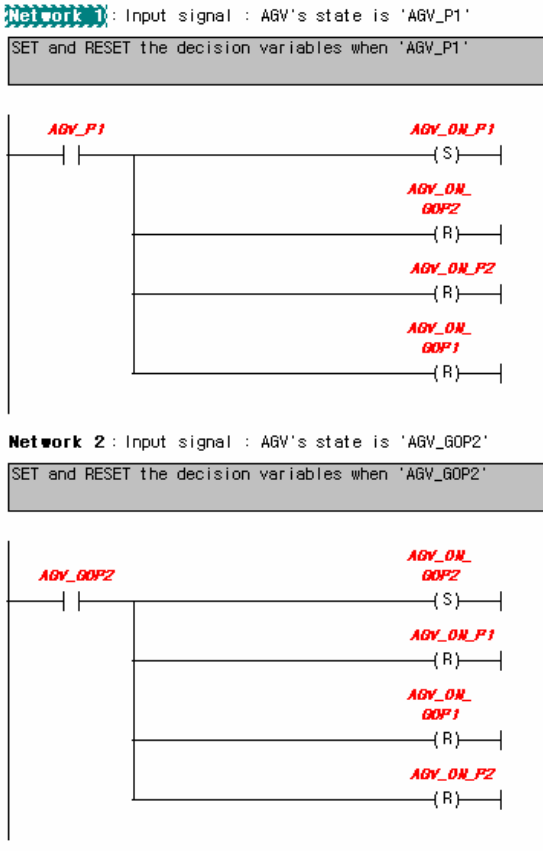


Figure 9(a): PLC code from the State Manager model of AGV model.

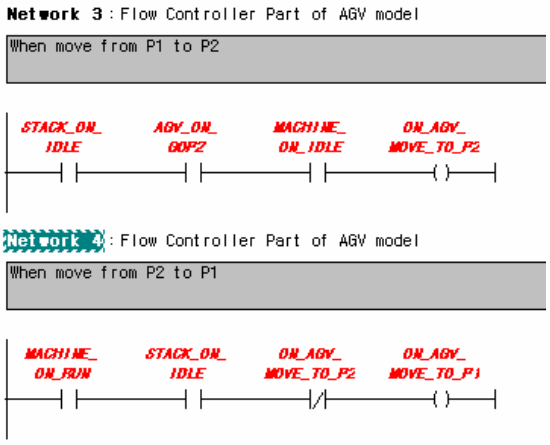


Figure 9(b): PLC code from the Flow Controller model of AGV model

7 DISSCUSSION AND CONCLUSIONS

This paper presents the PLC code auto generation methodology from the DEVS model. The PLC level control logic is rather closed and unopened engineering area while discrete event system modeling and simulation is widely used to measure the process capacity. By using the discrete event system simulation technique, the process or overall cycle time and throughput can be calculated.

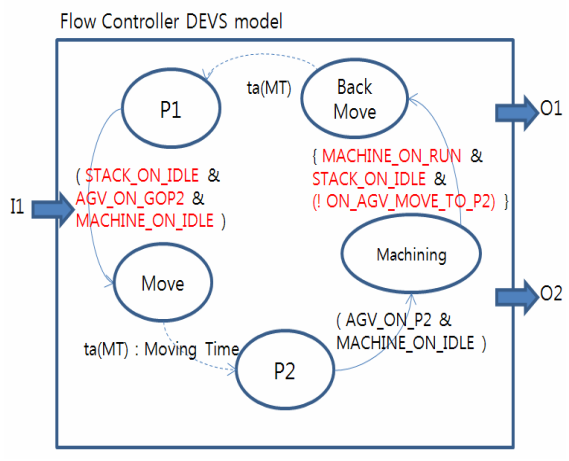


Figure 10: The Flow Controller DEVS model.

However, there is a big gap between the PLC code and the discrete event system simulation. This gap causes the repetition of process analysis work for the PLC programmer and the time delay to implement automated processing system in a manufacturing unit.

The overall procedure for proposed approach has three steps. Modeling the real system according to the three-phase-modeling framework is first step. And this model is converted to the DEVS formalism in second step. Among the 4 kind of models, the State manger and the Flow controller model is going to be replaced to the PLC part.

The generated PLC code from our approach can be categorized into two parts, one is from the state manager and another is from the flow controller. The first part is created from the input signals and the decision variable. And the latter part is from the control part which is from combination of decision variables.

The latter part generation is not achieved perfectly because the DEVS modeling level is more abstracted than the PLC level. However, this approach offers the overall framework for the PLC code generation from DEVS model. In the following future, the direction mentioned above will be the inevitable stream for the more physical process simulation, for the time saving toward the mass production condition and for better competitiveness to the company.

REFERENCES

B. K. Choi, B. H. Kim, 2000. Paper templates, In *Current Advances in Mechanical Design and Production Seventh Cairo University International MDP Conference*. New Trend in CIM: virtual manufacturing systems for next generation manufacturing.

J. Rumbaugh, M. Blaha, W. Premerlani. 1991. Paper templates, In *Prentice Hall Inc*. Object-Oriented Modeling and Design.

B. K. Choi, H. Kwan, T. Y. Park, 1996. Paper templates, In *The International journal of Flexible Manufacturing Systems*. Object-Oriendted graphical modelling of FMSs.

K. Y. Chen, S. S. Lu, 1997. Paper templates, In *International journal of Computer Integrated Manufacturing*. A Petri-net and entity-relationship diagram based object oriented design method for manufacturing systems control.

M. Onosato, K. Iwata, 1993. Paper templates, In *CIRP*. Development of a virtual manufacturing system by integrating product models and factory models.

Sang C. Park, 2005. Paper templates, In *Computers in Industry*. A methodology for creating a virtual model for a flexible manufacturing system.

B. P. Zeigler, 1984. Paper templates, In *Academic Press*. Multifacetted Modeling and Discrete Event Simulation.

E. A. Parr, 1999. *The book*, Programmable Controllers : An Engineer's Guide 3rd ed.

M. Maslar, 1996. Paper templates, In *IEEE Pulp and Paper Industry Technical Conference*. PLC standard programming language: IEC61131-3

F. Breuss, W. Roeger, 2005. Paper templates, In Journal of Policy Modeling. The SGP fiscal rule in the case of sluggish growth: Simulations with the QUEST

T. G. Kim, 1994. *The Book*. DEVS++ User's Manual

RE-USING 3D MODELING DATA FOR CONSTRUCTING 3D SIMULATION DATA

Jonggeun Kwak
Department of Industrial Information & Systems Engineering, Ajou University
San 5, Woncheon-dong, Yeongtong-gu, Korea
jkwak@ajou.ac.kr

Min. S. Ko, Sang C. Park, Gi-Nam Wang
Department of Industrial Information & Systems Engineering, Ajou University
San 5, Woncheon-dong, Yeongtong-gu, Korea
sebaminsuk11@ajou.ac.kr, scpart@ajou.ac.kr, gnwang@ajou.ac.kr

Keywords: CAD, CAM, Modelling, Simulation, Manufacturing system design, PLC.

Abstract: With the aid of the powerful computational ability and software tools, we undergo rapid change in a whole product manufacturing process. In a traditional way, it took long time and cost to build real manufacturing line. The behind time change for the manufacturing process ends up with supplementing large amount of budget. Therefore early detecting the errors on manufacturing process saves quite a big amount of time and money. As a result, the need for plant simulations rises. When we simulate manufacturing line on a virtual environment, it is not easy to acquire 3D data. If we have 3D CAD data, we can reuse them for each tools, products and equipments for the manufacturing line. Even in this case, the size matters. The large size of CAD data makes it difficult for us to directly use CAD data for simulation. As the CAD data and simulation data differs in their own purpose, we can reduce the size of the CAD data without losing simulation purpose. In this paper we propose effective methods for reducing the size of the CAD data and re-using them for simulation, assuming the 3D CAD data are already available.

1 INTRODUCTION

As the computer hardware gets faster, we can deal with large data which needs huge computation. In a product design, the commercial CAD software is widely used to model the products. Preparing 3D simulation data from the scratch takes enormous efforts and time which is not applicable for most cases. The importance of the availability of CAD information in product and process design and process visualization is increasing rapidly. If we already have CAD data for each component used in the manufacturing process, we can reuse the CAD data for simulation.

To build the simulation data, we need to integrate the CAD information with manufacturing models such as machining process simulation models upstream with concurrent engineering activities. In this case, there are two problems. One is the data interfacing problem due to various CAD data format (Iyer, Arjun, 2002). The other is the size of the integrated CAD data. For the data format problem, we used simple STL (Stereolithography) format as a data interface. For the data size, we will discuss in this paper.

Simulation data for the production line consists of hundreds of CAD data for each device, robot and part. Each CAD data can be some hundred megabytes when represented in STL format. Even for a small cell, the size of direct sum of CAD data can reach several gigabytes. As a matter of fact, this approach is not applicable for production line or whole plant even we take consideration of the high performance of the modern computer hardware.

The purpose of the CAD data is different from that of the simulation data. The main purpose of CAD data is to produce the model by verifying the shape and checking the static interference between assembled parts. It contains very detailed level of geometry like small sized holes and small parts like bolts and nuts, which results in a big size of the data. On the other hand, simulation data needs not be

described so detail. Though, it depends on the simulation purpose, generally speaking, simulation data needs far less level of detail compared to that of the CAD data. In a large scale simulation like manufacturing process simulation or whole plant simulation, we assume that the operations which need high accuracy already have been verified on a cell level using OLP(off line programming). It means, the error tolerance of the geometry for the simulation data can be larger than that of CAD data. Using the increased error tolerance, we can reduce the size of CAD data. In this paper, we assume that

1) 3D CAD data is available for simulation.
2) In some cases, we assume that OLP program data is available, like IGRIP or RobCAD for the devices which has kinematic information like robots or jig fixtures.
3) Triangular mesh is used for solid geometry handling for various CAD data interface.
4) We do not assume any restriction for the relationship between original CAD data and converted triangular mesh geometry hierarchy by IGRIP or RobCAD.
5) We assume developing our own simulation software to adopt our simulation data.

2 REUSING 3D CAD DATA

We are developing manufacturing process simulation software called PlcStudio. It is divided into two major modules called generic kernel and graphic module. The generic kernel is used to build logical model of the manufacturing process. It uses either software PLC(Programmable Logic Control) or hardware PLC to control and verify the manufacturing process, and interfaces graphic module to display the simulation results and/or to get the sensory information (Chang Mok Park, 2006).

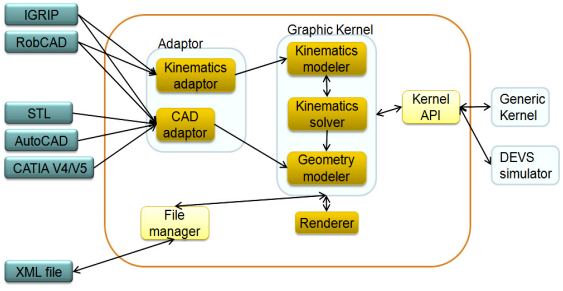


Figure 1: Graphic module architecture.

Graphic module gets the geometry information from the CAD file generated by the CAD software like AutoCAD or CATIA. Similarly, it can get both the geometry information and kinematic information from various OLP software like IGRIP or RobCAD.

2.1 Background and Related Work

There were many research for simplifying mesh for the curved surface. Michael *et el*. summarized these efforts into 3 classes (GarlandM, 1997). Vertex decimation, vertex clustering and iterative edge contraction. They suggested pair contraction method using a quadric error calculation. The implementation of the idea is available as an open source, and it shows nice performance for the curved surface. In many cases, the geometry of device for the production line does not contain curved surface. Therefore this approach is not effective for device CAD model. For device CAD model, it is more plausible to use geometric features only if we can extract them.

2.2 Classification of CAD Data

If we have to build simulation data from the CAD files, the size of total simulation data grows exponentially. We can effectively reduce the size of CAD data based on the features

Conceptually, CAD data can be divided into two groups. The one is manufacturing devices like robots, jigs, conveyors etc., which contains few curved surfaces. The other is parts or workpieces, which contains lots of curved surface. For the convenience, we call the former as *device CAD model* and the latter as *workpiece CAD model*.

2.2.1 Device CAD Model

Device CAD model data has the following characteristics.

1) Has few curved surface :
   In almost case, it can be represented with the combination of box, cylindrical frustum, sphere etc.
2) Has lots of primitives :
   Contains lots of cylinders, boxes etc.
3) Has lots of small part :
   It contains lots of small parts like bolts and nuts, which are not necessary for simulation purpose.
4) Has lots of holes :
   The purpose of the device CAD model is to manufacturing the device itself, so it contains lots of holes for bolts and nuts for part assembly.

5) It contains inner part data :
In many cases, device CAD model data contains invisible inner part data, which are not necessary for simulation purpose.

2.2.2 Workpiece CAD Model

In many cases, workpieces have lots of curved surface. For car assembly line, almost every workpieces have smooth surface, like body, side, door, hood, fender etc. The CAD model for this workpieces are far more bigger than that of the device CAD model, because it represents smooth surface using triangular mesh. Typically, CAD model for car door exceeds some hundreds mega-bytes, while the size of device CAD model for robots is some mega-bytes.

Due to the different characteristics, we apply different methods to reduce the size of CAD model.

2.3 Reducing Device CAD Model Size

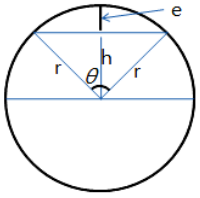
After converted by various software, the structural information for the geometry data can be lost. This makes it difficult to go further processing for geometry handling. However, it is relatively easier to detect geometric features from device CAD model rather than workpiece CAD model.

We used the geometric features extracted from device CAD model to reduce data size.

2.3.1 Replacing the Primitives

In many cases, the device CAD model contains lots of primitives like cylinder, sphere etc. For replacing cylinders, we assume that a *n*-side cylinder representation in the CAD model is an approximation of the pure logical cylinder which has infinitely large *n*.

A cylinder consists of top, bottom and *n*-side faces, where $n \geq 3$. A cylinder with *n*-side consists of *4n-4* triangles. A cylinder contained in the device CAD model typically has large *n* value. By detecting this cylinder geometry and reducing the *n*, we can reduce the size of data.



$$\cos\frac{\theta}{2} = \frac{h}{r} = \frac{r-e}{r}$$

$$\theta = 2 \times \operatorname{acos}\frac{r-e}{r}$$

Figure 2: Calculation of *n*-side for cylinder.

Figure 2 shows how many number of cylinder sides we can reduce, when the final error tolerance *e* for simulation data is given. Using the radius *r* of the pure logical cylinder and error tolerance *e*, we can calculate maximum θ for reduced cylinder. The reduced cylinder has $360/\theta$ side faces.

Though this method is quite effective, it is difficult to apply in real situation, because it is hard to identify cylinder geometry from unstructured triangular meshes. For sphere, it is more difficult to identify. So this method is only applicable when the CAD data contains structural information and we can identify them.

Another problem for this method is after replacing into the simple geometry, some parts attached to the original cylinder side surface can be dangling on a replaced cylinder.

2.3.2 Removing Small Parts

The device CAD model contains lots of small components needed for part assembly. Small bolts and nuts are one of those things. If these parts are so small that the size is less than the error tolerance for the simulation, we can safely remove them.

But in this case, it is not always possible to identify small parts from unstructured triangular meshes.

2.3.3 Removing Small Holes

The effect of removing holes from the geometry is quite potential. A cube without hole can be represented with *12* triangles, while a cube with one hole of *n* side needs *4n + 8* triangles. (n+4 for top, n+4 for bottom, 2n for side) If a hole is consists of 20 sides, the number of triangles reaches 4*20+8 = 88, which is more than 7 times of 12. If number of holes increases, the number of triangles increases rapidly.

The hole removing algorithm is difficult to implement, because it is not always possible to identify holes from unstructured meshes. For this reason, first we detect small inner circle from a surface. It is relatively easy to implement even if we don't have structural information.

Step 1: Group faces which exists in a same plane. That means, the faces has the same plane equation, and they are inter-connected.

Step 2: merge the grouped faces

Step 3: Trace boundary edges from the merged face

Step 4: Identify outmost boundary edges

Step 5: Using the boundary edges, re-triangulate the surface.

By applying these steps, we can easily remove any circle in a plane, but it can only remove top or bottom of the hole, whereas the wall of the cylinder still remains. Theses cylinder walls are hard to identify from unstructured set of meshes. These walls can be further removed by visibility check routine explained later.

2.3.4 Removing Invisible Parts

In many cases, CAD model contains invisible parts. Components or parts of a component placed below other component are invisible from user. For example, the engine installed in a car is not visible to the user unless the hood is open. In a simulation, these invisible components need not be rendered. After applying hole removal process, the cylinder walls of the hole geometry are not removed, yet. These wall are invisible to the user, so it can be removed by applying visibility checking, too.

Deciding whether a given geometry is visible from a given camera position or not requires quite large computation and the implementation of the algorithm is not simple. With the aid of modern graphic card functionality, we can successfully distinguished invisible parts from visible parts (Sang C. Park, 2005). This method is easy to implement and it is fast enough to apply, because it uses hardware function.

We take a shot for the scene while moving the camera position. The typical camera positions looking at the scene are 6 cubic directions plus 8 corner positions. Cubic directions are TOP, BOTTOM, FRONT, REAR, LEFT and RIGHT directions. Corner directions are top-front-left, top-front-right, top-rear-left, top-rear-right, bottom-front-left, bottom-front-right, bottom-rear-left and bottom-rear-right directions as depicted in figure 3.

After taking shots at a given camera positions, we collectively combine all shots, and mark(remember) all visible parts. On the whole scene, parts which is not marked as visible forms invisible parts. The camera positions are heuristically determined, and there are cases that given camera positions can't distinguish invisible parts. But in many cases, 6 cubic angle point and 8 additional corner point is enough to distinguish visibility property.

These process can be implemented by drawing each entities with unique color onto the back-buffer, while moving the camera positions, and read each pixel value back from back-buffer, and checking the color value as a unique key.

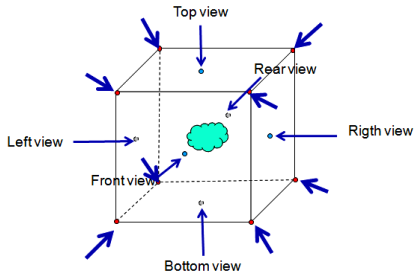
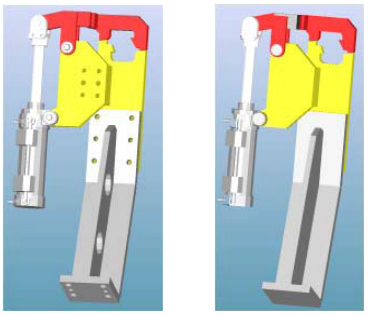


Figure 3: Camera positions to view the scene.

By using graphic hardware for deciding visibility, we need not implement the visibility algorithm and it can be computed very fast. The visibility checking algorithm implemented in a graphic card is widely tested by vendor for a long period of time, so it is well verified algorithm. We can use the algorithm almost for free. In addition, modern graphic card can draw millions of triangles in a second. It means that we can compute the visibility for a large scene in a very short time.

Next figure shows the result of data reduction by removing invisible parts after removing small parts and holes.



triangles = 8220 # triangles = 1845

Figure 4: Processing result: before and after.

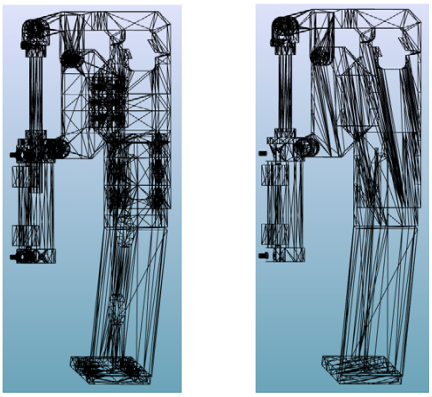


Figure 5: Mesh view: before and after.

2.4 Reducing Workpiece CAD Model Size

Workpiece CAD model which contains very complicated surface is not proper to apply the proposed algorithm. The proposed algorithm is based on the features extracted from 3D CAD model. It is almost impossible to identify features from the workpiece CAD model containing curved surfaces. So we can use well known mesh simplification methods like decimation to the curved surface.

3 RESULT AND DISCUSSION

The proposed methods are based on the feature extraction of the CAD model. Extracting primitives from the CAD is not always successful. If the CAD keeps the information about the primitive, we can easily extract them and we can apply the methods. But in general case, we can't assert that information is available. Identifying small parts has the same problem. But, identifying the holes and removing invisible parts works very well even the CAD data does not keep any information about the feature. It can even be extracted from raw triangular mesh.
The next table shows the result of reducing cell data which contains 16 devices. We only applied non curved surface reduction algorithm for this result.

Table 1: CAD data reduction result.

	Original data	Reduced data	Ratio(%)
# devices	16	16	100
# solid	4,567	2,966	64.94
# mesh	86,833	24,170	27.84
# triangles	3,316,146	600,478	18.11

4 CONCLUSIONS

When we have to construct 3D simulation data from CAD data, we have difficulties for the size of the CAD data. In the proposed method, we used an hole removal and visibility checking algorithm to reduce the data size. This method is easy to implement and very fast because it utilizes graphic hardware functionality. If the original CAD data contains more information for solid identification, we can further apply the methods by replacing the geometries. These methods applied to non-curved surface do not distort the original shape except that it removes the holes and small parts, which are smaller than the simulation tolerance error. Using these methods, we can reduce the size of the sample production line by 20% of the original data. In our future work, we plan to develop improved method to identify features from unstructured triangular mesh.

REFERENCES

Sang C. Park: Pencil curve detection from visibility data. Computer-Aided Design 37(14): 1492-1498 (2005)

Chang Mok Park : Development of Virtual Simulator for Visual Validation of PLC program. In *CIMCA 2006*

Iyer, Arjun, 2002. CAD data visualization for machining simulation using the STEP standard. In *Journal of Manufacturing Systems. 2001/2002*.

GarlandM, Heckbert P. Surface simplification using. quadric error metrics. In *proceedings of SIGGRAPH 1997,. August 1997. p. 209–16.*

COOPERATIVE LOCALIZATION
Self-configuring Procedure of a Multi-robot Localization System with Passive RFID Technology

Mikko Elomaa, Aarne Halme
Automation Technology Laboratory, Helsinki University of Technology, Espoo, Finland
mikko.elomaa@tkk.fi, aarne.halme@tkk.fi

François Vacherand
Laboratoire d'Electronique et de Technologie de l'Information, CEA, Grenoble, France
francois.vacherand@cea.fr

Keywords: RFID, Multi-robot, cooperative, localization.

Abstract: This preliminary simulation study introduces methods to configure a low cost localization system based on existing passive RFID technology. A group of small robots work together in order to configure the system autonomously. Probabilistic estimation methods are used for data fusion. The robots should be able to build and expand the localization system without human aid. When properly configured the system is able to offer positioning information with bounded error. The use of passive RFID tags as beacons makes the cost of expanding the robots' working area negligible.

1 INTRODUCTION

For a successful task execution a mobile robot has to know its position and heading. Different kinds of localization methods have already been developed. There are systems where all the needed equipment for the localization (wheel encoders, gyroscopes, laser scanner, etc.) is on board the robot. One problem with this kind of localization system is that over time the robot's estimate of its absolute location can get too erroneous for effectively continuing the mission. Also the cost of the high precision equipment can be considerable.

Another approach is to use external references such as beacons or landmarks for localization. The mobile robot can use absolute position data provided by measurements and calculations involving these external objects in order to obtain a better position estimate. The main problem with this kind of system is that the installation of the required external objects can be costly and time consuming.

The system introduced in this paper uses passive RFID tags as beacons. A passive RFID tag contains an antenna and an IC with small (0-1kbit) memory capacity. It operates on the power obtained from the reader antenna. A typical operating range for a 900MHz system is up to 2 meters. The cost of one tag is only a few cents and it does not need external connections or a power supply. Also the sticker-like tags are light to carry and a robot can place them on suitable spots when moving to new working areas.

A multi robot approach is used for the system configuration in order to replace the need of highly accurate sensors for robot's localization in the initial phase. Using a group of robots instead of one robot also gives better fault tolerance as a single fault will not endanger the whole mission.

2 RELATED WORK

Several studies have already been made on passive RFID localization in robotics using a mobile RFID reader (Hightower et al. 2000; Hähnel et al. 2004; Bohn 2006; Kulyukin et al. 2004 and Kleiner et al. 2006). All of these are very different from each other. Kulyukin et al. use a RFID system for recognition of specific places inside a building. The system is intended to help visually impaired people recognise specific office doors, elevators, etc. Kleiner et al. have been successfully using RFID tags for sharing mission related information in a

rescue scenario involving both robots and humans. Bohn introduces a super-distributed RFID tag infrastructure, where mobile objects may leave virtual traces in the physical space they traverse by writing an ID to the tags in the floor (up to 120 tags/m^2) while passing directly above them. Hightower et al. have designed a system to help people to localize objects equipped with passive RFID tags in their vicinity.

Hähnel et al. study the problem of localizing RFID tags with a mobile platform equipped with a laser scanner and a pair of fixed RFID antennas. A probabilistic sensor model of each antenna is used to estimate location of a detected tag. When a tag is detected for the first time a set of 1000 randomly chosen positions around the robot are chosen for initial estimates of the location of the tag. With each measurement the probability of these locations is calculated according to the sensor model. This is a single robot approach where the robot builds a database of tag positions. When localizing a robot with the RFID tags and odometry the position error was about 1 m.

The system proposed in this paper is based on similar technology as the aforementioned system by Hähnel et al. However, there are three major distinctions. Our approach uses several simple robots instead of one sophisticated robot. Instead of one database, the localization data is distributed to the tags. Also the relative displacement between the robot and a tag is based on measured bearing angles and not on a simple sensor model.

3 OPERATING PRINCIPLE

Our key interest is in developing a self configuring localization system using a group of simple, inexpensive robots. The idea is that even if the robots have only wheel encoders and an RFID reader for localization purposes they should be able to localize themselves within bounded error. An RFID reader is placed on each robot and stationary tags are placed around the working area of the robot group. The tags can be distributed by humans or by robots if they are equipped with an appropriate system (Kleiner et al. 2006)

The cooperative localization is based on a simple Kalman filter. When the robots are configuring the system and localizing the tags the main source of error is the accumulated odometry error which, on a group of robots, is assumed to follow roughly a Gaussian distribution. Thus when the location estimates of several independent robots on a common object are combined, the location estimate of the object converges towards the correct position.

3.1 System Operation

In the beginning the passive RFID tags contain no data. The robots start at some chosen reference location. A robot uses wheel encoders in order to keep track of its current position. When a robot detects an RFID tag it calculates an estimate for the location of the tag. The estimate of the tag's location has an uncertainty, which is calculated each time the tag's location is estimated. The necessary algorithms are explained in chapter 3.2.

The location estimate and the uncertainty are stored in the memory of the tag. The next robot that detects the tag reads the information found on the tag and calculates a new estimate for the location of the tag by combining the information stored in the tag with the new measurements.

As soon as there is a position estimate stored in the tag's memory the robots can use the tag as a beacon in order to correct their own position. When exploring new areas the robots have to return often to areas with well localized beacons in order to maintain reasonable estimate of their own position.

3.2 RFID Localization

The RFID localization is based on the measured bearing angle to a beacon represented by a passive RFID tag. The antenna of the RFID reader is turned one full circle while trying to contact tags near the robot. For each detected tag a bearing angle is calculated based on the sector where the tag responded to the calls of the RFID reader.

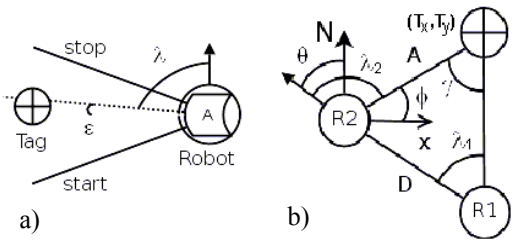


Figure 1a: Bearing angle estimation for a single RFID tag. 1b: Tag localization method.

The Figure 1a shows how the bearing angle of the tag is estimated. The start and stop angles define a sector where the tag responded to the readers calls. The bearing angle estimate λ is obtained by solving for the middle of the sector and subtracting the

systematic error ε. The systematic error depends on the geometry of the antenna and the immediate surroundings of the antenna's mounting place. It must be defined separately on each robot unless the robots and the antennas are exactly alike.

3.2.1 Tag Localization

The robot uses bearing angle measurements and odometry data in order to estimate the tag location (E_x,E_y). In Figure 1b places R1 and R2 represent two places where the robot has measured a relative bearing angle to the tag. The distance A to the tag is solved from the displacement D between the measurement places and the gamma angle. The absolute bearing angle φ is calculated as a function of the measured relative bearing angle λ_2 and the robot's estimated heading angle θ. A variance is calculated for each tag location estimate. The x- and y-coordinates have different variances which depend on the angle φ. The C and G are parameter constants

$$\gamma = |\lambda_2 - \lambda_1| \tag{1}$$

$$A = \frac{D \cdot \sin(\lambda_1)}{\sin\gamma} \tag{2}$$

$$\varphi = \frac{\pi}{2} - \theta + \lambda_2 \tag{3}$$

$$\begin{aligned} E_x &= r_x + a \cdot \cos(\varphi) \\ E_y &= r_y + a \cdot \sin(\varphi) \end{aligned} \tag{4}$$

$$\begin{aligned} var_x &= C + \frac{G \cdot \left|\cos\left(\varphi + \frac{\lambda_1 - \lambda_2}{2}\right)\right|}{\gamma^2} \\ var_y &= C + \frac{G \cdot \left|\sin\left(\varphi + \frac{\lambda_1 - \lambda_2}{2}\right)\right|}{\gamma^2} \end{aligned} \tag{5}$$

With each new angle measurement the robot calculates location estimates for the tag using all the previous angle measurement. Thus after two measurements the robot has one estimate for the tag location and n measurements give 1+2+3+...+(n-1) estimates.

Each estimate is fused with the previous estimate of the tag location using a simple Kalman filter. It is a recursive estimator, so all the prior information is contained in the previous estimate (Maybeck, 1979). The calculated variance P_k represents the uncertainty in the location of the tag. The equations for the Kalman filter are shown below.

$$K(k) = \frac{P(k-1)}{P(k-1) + var} \tag{6}$$

$$x(k) = x(k-1) + K(k) \cdot [E - x(k-1)] \tag{7}$$

$$P(k) = P(k-1) - K(k) \cdot P(k-1) \tag{8}$$

Odometry error and angle measurement error may both cause significant deviation from the correct position. Thus several estimates from different robots are needed in order to properly localize the tag.

4 RESULTS

Simulation models for the multi-robot localization system were built in order to get an idea of the effects of different measurement errors. The simulations were run on Matlab. The system presented here simulates a scenario where a single tag is localized by a group of robots passing the tag one by one. Each robot has random error on its own position and heading angle estimate. Each bearing angle measurement also contains a random error. After each bearing angle measurement the robot calculates new estimate of the location of the tag. The result is then written on the tag's memory. A robot is able to detect the tag inside a circular area (r=50cm) in front of the tag. If the tag already contains an estimate of its position, the robot tries also to correct it's own position.

4.1 Effects of Inaccuracies

Several simulations were run with different parameters in order to discover the effects of inaccuracies in different estimates. First simulation contained 500 independent groups of ten robots passing the tag and localizing it cooperatively. The Figure 2 shows the average and maximum error in the estimated location of the tag as a function of the error in the bearing angle measurement for two different runs.

The curves A and C represent average and maximum error when the robots' position error was at most ±8 cm on each axis. The curves B and D represent a run where robot's position error was doubled to ±16 cm. The maximum heading angle

error was ±4° on both runs. The simulation results show that the bearing angle error is a lot smaller factor than the error in robot's position.

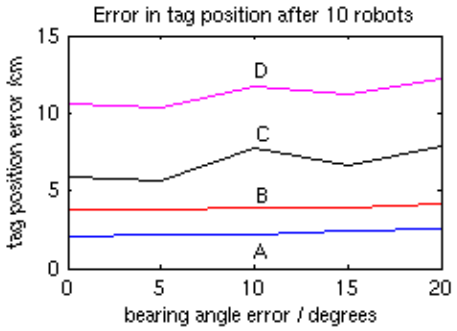


Figure 2: Effect of the bearing angle error.

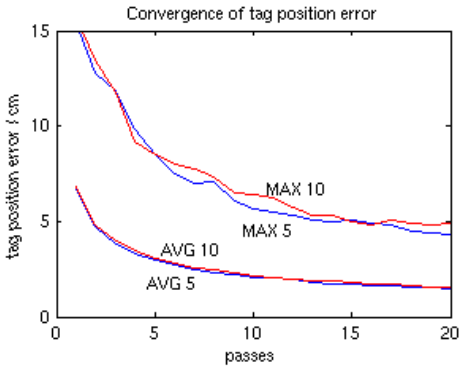


Figure 3: Convergence of tag position estimate.

The Figure 3 shows the convergence of tag's position estimate when 1000 independent groups of 20 robots pass a tag. Two runs with different bearing angle error factors (5 and 10) were made. The curves show average and maximum error in tag position after each robot. The increase in bearing angle error does not have significant effect on the position error. Also a smaller group of robots can localize a tag when they pass the tag multiple times.

5 CONCLUSIONS

The simulations indicate that the tags can be localized with certain accuracy even if the angle measurement to the tag is not very accurate. The error in the tag's position depends on several different factors, such as errors in robot's position and heading angle estimate. When several robots participate in the localization the position error settles on an acceptable level.

This study suggests that it is possible to build a localization system that offers a bounded error after it has been autonomously configured by simple, inexpensive robots with readily available RFID technology.

6 FUTURE WORK

Initial tests for the bearing angle measurements in the office environment are under way. The first RFID localization modules for laboratory tests will be designed according to the information obtained from the angle measurement tests. The system will be tested with a group of small robots. In order to obtain accurate localization information further research is required in tag positioning and filtering of the bearing angle measurements in an office environment.

ACKNOWLEDGEMENTS

I want to thank CEA-LETI for offering the research facilities during my visit in France and Finnish Cultural Foundation for the funding.

REFERENCES

Bohn J., 2006. Prototypical Implementation of Location-Aware Services based on Super-Distributed RFID Tags., *19th International Conference on Architecture of Computing Systems*, LNCS No. 3894, Springer-Verlag, pp. 69-83, Frankfurt am Main, Germany.

Hightower J., Borriello G. ,Want R., 2000. SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength, *UWCSE Technical Report 2000.02.02*

Hähnel D., Burgard W., Fox D., Fishkin K., Philipose M., 2004. Mapping and Localization with RFID Technology., *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, New Orleans, USA

Kleiner A., Prediger J., Nebel B., 2006. RFID Technology-based Exploration and SLAM for Search And Rescue, P*roc. of International Conference on Intelligent Robots and Systems,* Beijing, China, 2006

Kulyukin V., Gharpure C., Nicholson J., Pavithran S., 2004. RFID in Robot-Assisted Indoor Navigation for the Visually Impaired, *Proceedings of Intelligent Robots and Systems,*. Volume: 2, page(s): 1979- 1984

Maybeck, Peter S., 1979. *Stochastic Models, Estimation and Control, Vol.1* Academic Press, Inc., 1979

COMPARATIVE STUDY OF ROBOT-DESIGNS FOR A HANDHELD MEDICAL ROBOT

Peter P. Pott, Markus L. R. Schwarz
Laboratory for Biomechanics and experimental Orthopaedics, OUZ, Medical Faculty Mannheim
University of Heidelberg, Theodor-Kutzer-Ufer 1-3, 68167 Mannheim, Germany
peter.pott@ortho.ma.uni-heidelberg.de, markus.schwarz@ortho.ma.uni-heidelberg.de

Achim Wagner, Essameddin Badreddin
Automation Laboratory, University of Mannheim, Mannheim, Germany
achim.wagner@ti.uni-mannheim.de, badreddin@ti.uni-mannheim.de

Keywords: Hybrid kinematics, Medical robotics, Comparison.

Abstract: Robotic systems are used within a great variety of medical disciplines. A handheld robot promises a number of advantages compared to conventional (medical) robots but this approach leads to strict specifications regarding size, weight and dynamic properties. A new hybrid kinematics – the Epizactor – seems to be advantageous and is compared to two well-known parallel kinematics regarding the ratio of workspace and volume the number of kinematic elements, the cost of computation, the stiffness the effects of clearance, actuation (weight), and accuracy using a well-described industrial method for comparison. It becomes clear that the Epizactor has advantages concerning the ratio of workspace and volume, needs a smaller number of kinematic elements and fewer computations, and has less than half the mass than the parallel kinematics. Its accuracy, stiffness and the effects of clearance are comparable. The advantages of this new kinematic set-up lead to a first deployment within the field of medical robotics.

1 INTRODUCTION

Design and evaluation of robotic set-ups for medical and especially surgical applications has been ongoing for the last 20 years. Systems for a vast variety of medical disciplines and deployments have been investigated (Pott PP et al., 2005). One possible solution to provide a useful tool for numerous medical tasks is to use a handheld robot that combines the process control of the surgical task by the surgeon and the accuracy and repeatability of a robot. Within the project "Intelligent Tool Drive" ITD, a handheld robot for orthopaedic surgery is being developed. The intention is to align a milling tool relatively to a patient and to decouple the tool from unintentional hand movements at the handle (Pott PP et al., 2003; Wagner A et al., 2004). A handheld robot has to be as small and lightweight as possible while providing high dynamics for accurate stabilisation of a surgical tool (Wagner A et al., 2004). This most important criterion is mainly determined by the kinematic set-up.

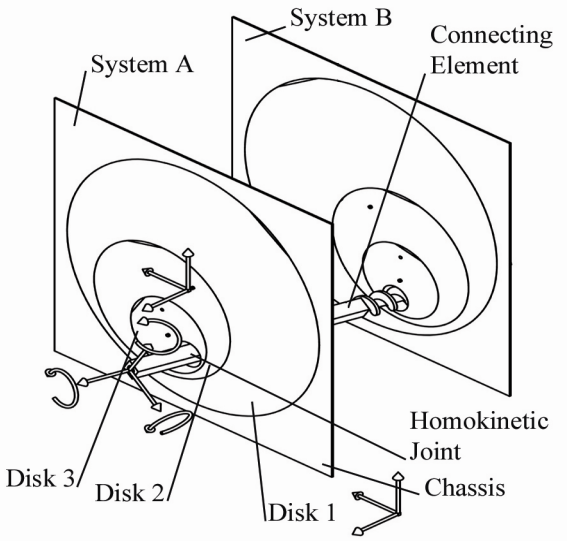


Figure 1: The EPIZACTOR, 6-DOF hybrid kinematics with rotating elements.

Parallel robots are widely used, where high stiffness, high dynamics, or low error propagation

over the kinematic chains is required, e.g. positioning and stabilization platforms (Huynh P, 2001) and vibration isolation (Chen Y et al., 2004). An obvious advantage is that a parallel robot provides high potential for a lightweight construction. The moving masses of parallel kinematics are low and this leads to low static and dynamic forces (Honegger M, 1999; Huynh P, 2001).

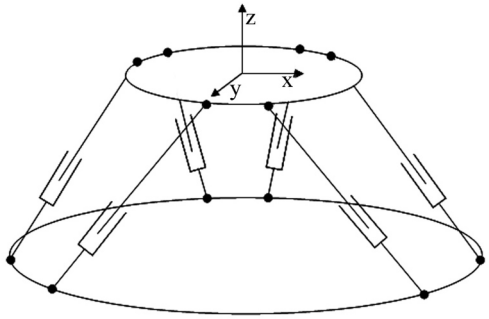


Figure 2: The HEXAPOD, 6-DOF parallel kinematics with actuated prismatic joints in the struts.

Alternatively a new kinematic set-up can be used (). This concept –called "Epizactor"– involves two disk-systems (systems A&B) each described by a planar 3-DOF 4-link manipulator. These two serial kinematic chains act on a connecting element that moves the surgical tool by homokinetic joints with a lead-screw and a prismatic section respectively. Each disk-system uses four links to overcome singularities by redundancy. So this hybrid kinematics uses only rotating elements to provide 6-DOF manoeuvrability (Pott PP et al., 2007; Pott PP, Weiser HP et al., 2004).

The aim of this work is the comparison of a new kinematic set-up with two well-known alternatives for a handheld medical robot.

2 MATERIAL & METHODS

Three different kinematic set-ups were assessed. The well-known Stewart-Gough-platform or Hexapod (Figure 2) with active struts (Gough V et al., 1962; Stewart D, 1965), the Merlet-platform or Hexaglide (Merlet JP, 1988) with base-fixed actuators and passive struts (Figure 3) and the Epizactor (Pott PP et al., 2007; Pott PP, Weiser HP et al., 2004) (). The first two set-ups are based on parallel kinematics while the Epizactor is based on a hybrid kinematics set-up. To describe the set-ups' forward and inverse kinematics as well as the inverse dynamic models a literature research and own considerations were conducted. For the assessment of the actual mechanical design three robots were available. Each is based on one of the kinematic set-ups described and shows a certain state of project ITD.

2.1 Comparison

To compare the kinematic set-ups the method by Kesselring (Kesselring F, 1951) is used. Here a set of criteria is defined and evaluated by one or more experts using a score reaching from 4 (very good) to 1 (poor). To further refine the comparison, each criterion is weighted. Finally for each kinematic set-up the sum of all products of score and weight are added up and lead to a total benchmark for each kinematic set-up. To define the weighting factors the method described by Wenzel (Wenzel R et al., 1971) is used. Here all criteria are listed and each criterion is compared to the remaining leading to a graduation in importance of the different criteria.

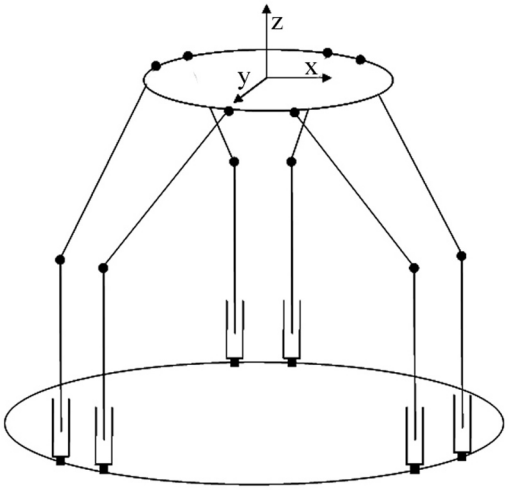


Figure 3: The HEXAGLIDE, 6-DOF parallel kinematics with base-fixed actuates prismatic joints.

To assess the three kinematic set-ups the following criteria were used.

2.2 Ratio of Workspace and Installation Space

Given a certain workspace needed for a specific task, this ratio describes how large a machine will become at least. Especially in a surgical environment the installation space should be as small as possible while the workspace is determined by the surgical task.

To assess this ratio each kinematic set-up was simulated using *Matlab* (The Mathworks Inc, Natick, MA, USA) and a given workspace definition of 40mm translation of the tool in each axis and ±20° rotation in each axis at all times. Using this simulation procedure the three kinematic set-ups were scaled until predefined kinematic restrictions were just not violated but the desired workspace could still be produced. Each kinematic set-up is circumscribed by a cylinder that defines the minimal installation space of an imaginary machine based on the corresponding kinematics. The length of the cylinder is aligned with the tool nearest to the base-platform.

2.3 Number of Kinematic Elements

Kinematic elements are defined as "joints", "struts" or "links", "base", and "tool". The number of kinematic elements can be used as a measure of complexity of a kinematic set-up (El-Shenawy A et al., 2007).

To assess the number of kinematic elements each kinematic set-up was analysed and the elements were counted.

2.4 Cost of Computation

The cost of computation can be a measure for the hardware-effort that has to be made by the control system to compute the forward and inverse kinematic problems in real-time.

To assess the cost of computation for each kinematic set-up the *Matlab*-code was analysed regarding the number of additions, multiplications/powers and trigonometric functions. To compare the three set-ups a computation of both kinematic problems was concerned and all computation steps were summed up. The set-up with greatest number was rated "1" the one with the smallest number was rated "4". In between a linear interpolation was performed.

2.5 Stiffness

Although stiffness is not a kinematic property and system stiffness is mainly affected by the actual design of a machine the three set-ups can be analysed qualitatively regarding the distribution of forces within the kinematic elements. It can be stated that short and compact elements under uniaxial load will be stiffer than flat elements under bending strain.

To analyse the three set-ups the kinematic elements were examined regarding the distribution of force and shape.

2.6 Effects of Clearance

As it can be regarded as one of the major impediments for accurate machine performance clearance becomes one of the most important criteria. Again this is not a purely kinematic feature but the kinematic set-up has an influence on error propagation.

To assess the effects of clearance a score especially for the assessment of parallel and hybrid kinematics was introduced (Pott PP, unpublished). It was assumed that the clearance k_i of i joints of a serial kinematic chain in the most unfavourable case is summed up to

$$k_{tot,ser} = \sum_i k_i \tag{1}$$

For the parallel arrangement of j serial chains it is assumed that the clearance can be treated as

$$k_{tot,par} = \sqrt{\sum_j \left(k_{j,ser}\right)^2} \tag{2}$$

For simplification the clearance in any joint is standardised to "1".

2.7 Actuation / Weight

The actuation of a robot is not a kinematic property but becomes important when size and weight of the actual machine is evaluated. Electromagnetic linear actuators provide high acceleration but a poor force-to-weight ratio. Correspondingly conventional

rotating motors deliver high power at high speeds but poor torque when used without gearing (Pott PP, unpublished). So the actuation has an immanent influence on the weight of a machine based on a certain kinematic set-up.

To assess the three kinematic set-ups they were analysed regarding performance needed and theoretical weight. To determine the strut forces of the Hexapod and the Hexaglide corresponding dynamic models were used (Dasgupta B et al., 1998; Khalil W et al., 2004; Wagner A et al., 2006)., dynamic models (Dasgupta B et al., 1998) were used. The Epizactor was assessed using a model based on the iterative Newton-Euler-Method and own considerations regarding the forward- and inverse kinematic problems (Pott PP et al., 2007). As input for the simulation a vibration trajectory with 12Hz and 1mm amplitude was used. This trajectory was applied to each of a set of 680 pre-defined grid-points throughout the whole desired workspace. Additionally the direction of the trajectory and the static forces $F = [20 \quad 20 \quad 20]^T N$ and moments $M = [1 \quad 1 \quad 0.8]^T Nm$ were permutated in the main coordinate system directions. Forces, moments, frequency (velocity, acceleration) were taken from the specifications of the handheld robot (Pott PP et al., 2003; Pott PP, Wagner A et al., 2004). Masses and mass-related values of the kinematic elements were taken from the CAD-models of the three available robots. The maximum forces and torques in each actuator were computed. For the parallel kinematics this force was multiplied by six, as the symmetry of the set-up leads to the conclusion that any actuator will have to be able to produce this force. Regarding the Epizactor the torques of all actuators were summed up. To achieve the theoretical weight of the actuators of each set-up the over-all force was multiplied with the specific force-to-weight-ratio of the linear actuators and the rotating actuators respectively. It could be shown from manufacturer's data that an average electric linear motor with a maximum force of about 50N (30s) has a force-to-weight ratio of about 47.2N/kg. The torque-to-weight ratio of an average motor with a gear that allows a torque of 1Nm is about 3.4Nm/kg (Pott PP, unpublished).

2.8 Accuracy

The accuracy of a robot is determined by the accuracy and resolution of sensors and actuators, the adjustment of control parameters, the elastic properties of the mechanics, and the transformation of workspace coordinates into actuator axis positions. As the first three parameters are affected by the actual mechanical design, the latter is dependent on the kinematics-type and actual configuration only.

To assess the theoretical accuracy the tolerable position error of the robot of 0.1mm (Pott PP et al., 2003) was applied to the set of grid-points described above. Doing so, the displacement of the actuators was computed and compared to a realistic accuracy of 0.005mm and 0.0005rad respectively, which can be reached by real encoders used in a mechanical design. A score was introduced that describes the number of points where the accuracy specification is reached.

3 RESULTS

3.1 Comparison

Table 1 shows a summary of the results for each kinematic set-up. The comparison criteria are aligned in rows. The columns show results, ratings and weighted scores for each of the three kinematic set-ups. The results of the comparison and a ranking are summed up in the bottom line. It becomes clear that the Epizactor has advantages regarding the ratio of workspace and installation-space and theoretical weight of its actuators. Additionally it needs less kinematic elements, uses rotating actuators that provide a better performance-to-weight ratio than linear motors, and needs fewer computations for the inverse and direct kinematic problem. Main disadvantage of this new kinematic set-up is the limited stiffness.

3.2 Ratio of Workspace and Installation Space

The Hexapod can be enclosed by a cylinder with a minimal volume of 3941cm³. This leads to a ratio of workspace and installation space of 1:62.

The cylinder around the Hexaglide has a minimal volume of 4247cm³ so the ratio of workspace and installation space can be computed to 1:66.

A cylinder circumscribing the Epizactor has a volume of 1445cm³. Compared with the required workspace this leads to a ratio of workspace and installation space of 1:23.

3.3 Number of Kinematic Elements

The Hexapod and the Hexaglide each consist of a base-platform, a tool-platform, and six legs. Each leg uses two rotating and one prismatic (actuated) joint and a strut. Overall 26 kinematic elements can be counted.

The Epizactor consists of a base, a connecting element (tool) and two identical disk-systems. Each disk-system has 3 disks and a homokinetic joint. The first disk is actuated directly. To drive the 2nd disk a single toothed ring is necessary, to drive the 3rd disk two toothed rings are used and the joint is driven by three rings. Overall 22 kinematic elements can be counted.

3.4 Cost of Computation

To calculate the Hexapod's inverse kinematic problem 60 additions, 132 multiplications and powers, and 174 trigonometric functions have to be computed. The forward kinematic problem can only be solved by an iterative procedure. Within the simulations carried out around 40 iteration steps were necessary for each computation. To compute the kinematics once in both directions 15086 computations have to be carried out.

For the Hexaglide the inverse kinematic problem is solved in the same way and also the forward kinematics needs to be computed by iterations. Here around 10 iteration steps were necessary. To compute the kinematics once in both directions 4046 computation steps have to be done.

The inverse kinematic problem of the Epizactor needs 106 additions, 171 multiplications and powers, and 132 trigonometric functions. The forward kinematics are computed by 31 additions, 115 multiplications and 177 trigonometric functions (Pott PP, unpublished). To compute the kinematics once in both directions 732 computations have to be carried out. One has to consider that the inverse kinematic problem finally is solved by a singularity-robust control algorithm (Chung YG et al., 2000; Pott PP et al., submitted) and does not need to be computed.

3.5 Stiffness

The struts of the Hexapod kinematics can be regarded as pendulum links. Each of the struts is loaded by pressure and tension which must be fully

Table 1: Summary of the results and comparion of the three kinematic set-ups assessed. The comparion criteria are aligned in rows, the results of the comparison, the rating, and weighted results are listed in columns. The bottom lines show the result and the ranking of the three kinematic set-ups. The higher the result the better the set-up is suited for the assessed deployment.

criteria	unit	weight	Hexapod			Hexaglide			Epizactor		
			results	rating	result rating	results	rating	result rating	results	rating	result rating
Ratio of Workspace and Instal-lation Space	1	0.28	1:62	2	0.56	1:66	2	0.56	1:23	4	1.12
Number of Kinematic Elements	1	0.09	26	3	0.27	26	3	0.27	22	4	0.36
Cost of Computa-tion	score	0.02	15086	1	0.02	4046	3.36	0.07	732	4	0.08
Stiffness	rating	0.14	high and constant stiffness	4	0.56	high but variable stiffness	3	0.42	medium stiffness	2	0.28
Effects of Clearance	score	0.19	7.3	2	0.38	7.3	2	0.38	7.1	2	0.38
Actuation / Weight	kg	0.19	7.1	1	0.19	5.6	2	0.38	2.2	4	0.76
Accuracy	score	0.09	300	1	0.09	440	2	0.18	420	2	0.18
Results of the compa-rison					2.07			2.26			3.16
Ranking					3			2			1

absorbed by the actuated prismatic joint in each strut. As the passive parts in each strut can be designed as stiff as necessary the stiffness of the prismatic joint depends on its design, the construction of the actuator, and the quality of the control loop. The stiffness of the struts is almost constant over the length so it can be stated that the over-all stiffness of the Hexapod in its workspace is constant.

To consider the stiffness of the Hexaglide a similar approach can be used. Differences exist as the struts of the Hexaglide usually tend to be as light as possible as they do not need to be actuated. Also two variations of parallel kinematics with base-fixed actuation exist. One that uses sliders (Hebsacker M et al., 1998), here the over-all stiffness within the workspace can be seen as constant. The version that uses piston-like actuators (Merlet JP, 1988) provides lower stiffness for extended actuators and changing over-all stiffness within the workspace.

Forces applied to the connecting element of the Epizactor are propagated to the disk-systems and absorbed within the planes of the disk systems. The stiffness of the disk-systems in this direction is rather good. Only the force-component within the axis of the connecting element acts perpendicular to disk-system B so that here the stiffness is less good. The stiffness is constant within the workspace as the connecting element can be designed as stiff as necessary.

3.6 Effects of Clearance

The Hexapod is based on six identical kinematic chains between base-platform and tool. Each chain consists of three joints. The clearance $k_{tot,ser}$ of each chain is computed to

$$k_{tot,ser} = \sum_i k_i = 1+1+1 = 3 \cdot \qquad (3)$$

For the whole set-up the score can be computed to

$$k_{tot,par} = \sqrt{\sum_j \left(k_{j,ser}\right)^2} = \sqrt{6 \cdot 3^2} = 7.3 \qquad (4)$$

The same assumptions can be made for the Hexaglide and lead to a similar result.

The Epizactor uses two serial chains with 5 elements acting in parallel on the connecting element. The clearance $k_{tot,ser}$ of each chain is computed to

$$k_{tot,ser} = \sum_i k_i = 1+1+1+1+1 = 5 \cdot \qquad (5)$$

For the parallel arrangement of the two chains, the overall score can be computed to

$$k_{tot,par} = \sqrt{\sum_j \left(k_{j,ser}\right)^2} = \sqrt{5^2 + 5^2} = 7.1 \cdot \qquad (6)$$

3.7 Actuation / Weight

The simulations for the Hexapod lead to maximum forces of 55.7N in each strut. Thus the six actuators needed to drive the Hexapod weighs at least 7.1kg.

For the Hexaglide the maximum force needed to drive the set-up is 44.4N. With the same considerations regarding the force-to weight-ratio the actuators theoretically weigh about 5.6kg.

The Epizactor has a maximum torque requirement in the specific actuators of 1.33Nm, 1.53Nm, 0.94Nm, 0.8Nm, 0.95Nm, 1.52Nm, 0.26Nm, and 0.02Nm. So the theoretical weight of all actuators of the Epizactor sums up to 2.2kg.

3.8 Accuracy

The Hexapod can provide the desired kinematic accuracy in 300 of the tested 680 grid-points. These points are located near the main *xz*- and *yz*-plane of the base-platform.

The Hexaglide reaches the desired accuracy in 440 of the tested grid-points. These are distributed symmetrically to the main *xz*-plane of the base. Within a small strip just next to this plane the accuracy is not reached.

The Epizactor reaches the accuracy specification on 420 grid-points. These are symmetrically distributed within the workspace. The desired accuracy is not reached at points were a certain configuration of the disks leads to a very sensitive behaviour of the kinematics.

4 DISCUSSION

Three different kinematic set-ups have been evaluated. The method to compare the three kinematic set-ups refers to the German norm VDI 2222. This method leads to a reproducible result when it is done out by a group of experts. Here a single expert carried out the comparison so a certain bias can be assumed. However as primarily measurable criteria were evaluated, the bias is believed to be small. The graduation of the ratings is rather raw but this simplifies the rating itself. It becomes obvious that the Epizactor provides a

number of advantages when compared to two well-known parallel kinematic set-ups. This lead to the decision to design such a set-up within the medical robotics project ITD and for future projects.

It could be shown, that the ratio of desired workspace and theoretical installation space of the Epizactor is about three times better when compared to the well-known parallel kinematics. The desired workspace is derived from the specifications of the ITD project and is described by a cube. If another workspace-specification is taken into account, the comparison will produce different results. The method to simulate the desired workspace and to scale down the kinematic set-up does not lead to an optimization but seems to approximate the actual set-up to this point. This criterion is the most important within the comparison.

The Epizactor needs a smaller number of kinematic elements than the Hexapod or the Hexaglide. However the larger number of common parts within the parallel kinematics simplifies the actual manufacturing of those kinematic set-ups.

The fact that there seems to be no closed-form solution for the inverse kinematics of the Hexapod and the Hexaglide leads to a large number of computations for the forward kinematic problem. In contrast the mathematically well-defined kinematics of the Epizactor leads to only a small number of computations. It has to be considered that the code analysed for comparison is not optimised. Although the cost of computation is unequally distributed between the three kinematic set-ups this criterion has the least importance, as today's computer performance allows even large computations in real-time.

Stiffness is depending on the actual design of a machine and is not an original kinematic property. It is affected by the force distribution in the kinematic set-up and therefore can be regarded quite important as it applies to the robot's accuracy. Here stiffness is analysed qualitatively and leads to the conclusion that the Epizactor appears to be less stiff as forces are distributed through flat rotating elements rather than by robust pendulum supports utilised by the parallel kinematics.

Although the effect of clearance is the second most important due to accuracy reasons the differences between the three set-ups are marginal as it could be shown by the score that was introduced. This can be explained by the fact that the length of the kinematic chains and their quantity compensate for each other. One has to remark that this score can only be applied to parallel kinematics and that experimental results have not yet been made to substantiate this comparison.

While actuation is not a kinematic property this criterion is used to evaluate a theoretical weight of the actuators and hence for the weight of a hypothetically realised machine. During the work on the ITD-project it became obvious that linear actuators seem to provide an unfavourable ratio of force and weight, so that a machine driven by such actuators becomes heavier than a machine driven by rotating actuators considering comparable performance. This also applies when rotating spindles are used because of their additional weight. The parallel kinematics are 3.2 times (Hexapod) and 2.5 times (Hexaglide) heavier than the Epizactor due to the use of rotating actuators in this set-up and its more favourable dynamic properties.

The resolution and accuracy of sensors for linear displacement and rotating angles are limited. So the accuracy of a machine based on a certain kinematic set-up is not only limited by mechanical precision, elasticity and the quality of the control loop but also by the kinematic transformation of tool and axis coordinates. The Epizactor's coordinate transformation seems to be advantageous here.

The discrimination between purely kinematic properties and features of the technical realisation is not easy. This seems not to be a disadvantage as the idea of the Epizactor aims to a practical use of the kinematics in a handheld medical robot.

5 CONCLUSIONS

The Epizactor is a new kinematic concept for a small 6-DOF robot. A first deployment of this approach will be a handheld robot for medical applications. Here sharp restrictions regarding size, weight and workspace exist and it could be shown, that the Epizactor meets the main specifications in a most favourable way.

ACKNOWLEDGEMENTS

The work on the ITD-project is supported by the AiF, Berlin, Germany and has been supported by the German Research Society. The work on the Epizactor has been supported by the state of Baden-Württemberg, Germany. The authors want to express their most sincere gratitude to Prof. P.

Weiser and Mr. Steffen Heute who both contributed greatly to the development of the Epizactor.

REFERENCES

Chen Y, & McInroy JE. (2004). Decoupled control of flexure-jointed hexapods using estimated joint-space mass-inertia matrix. *IEEE Transactions on Control Systems Technology, 12*(3), 413- 421.

Chung YG, & Lee B. (2000). Torque Optimizing Control with singularity-robustness for kinematically redundant robots. *Journal of Intelligent and Robotic Systems, 28*, 231-258.

Dasgupta B, & Mruthyunjaya TS. (1998). A Newton-Euler formulation for the inverse dynamics of the Stewart-Platform manipulator. *Mech. Mach. Theory, 33*(8), 1135-1152.

El-Shenawy A, Wellenreuther A, Baumgart A, & Badreddin E. (2007). *Comparing Different Holonomic Mobile Robots.* Paper presented at the 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, Canada.

Gough V, & Whitehall S. (1962, oder 1949). *Universal Tyre Test Machine.* Paper presented at the IX Int. Techn. Congr. F.I.S.I.T.A.

Hebsacker M, & Codourey A. (1998). *Die Auslegung der Kinematik des Hexaglide – Methodik für die Auslegung paralleler Werkzeugmaschinen.* Paper presented at the VDI Fachtagung Parallele Strukturen, TU Braunschweig.

Honegger M. (1999). *Konzept einer Steuerung mit Adaptiver Nichtlinearer Regelung für einen Parallelmanipulator.* Dissertation, ETH, Zürich.

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

Kesselring F. (1951). *Bewertung von Konstruktionen*. Düsseldorf: Deutscher Ingenieur-Verlag.

Khalil W, & Guegan S. (2004). Inverse an Direct Dynamic Modeling of Gough-Stewart Robots. *IEEE Transactions on Robotics, 20*(4), 754-762.

Merlet JP. (1988). France Patent No. 2628670.

Pott PP. (unpublished). *Untersuchung von Kinematiken für handgehaltene Roboter.* Dissertation, Universität Mannheim, Mannheim.

Pott PP, Scharf H-P, & Schwarz MLR. (2005). Today's State of the Art of Surgical Robotics. *Journal of Computer Aided Surgery, 10*(2), 101-132.

Pott PP, & Schwarz MLR. (2007). The Relation of Workspace and Installation Space of Epicyclic Kinematics with six Degrees of Freedom. *Zeitschrift für Biomedizinische Technik, 52*(5), 323-336.

Pott PP, Schwarz MLR, Köpfle A, Schill M, Wagner A, Badreddin E, et al. (2003). *ITD - A handheld manipulator for medical applications - Concept and design.* Paper presented at the 3rd annual meeting of CAOS, Marbella, Spain.

Pott PP, Wagner A, Badreddin E, & Schwarz MLR. (submitted). Inverse Dynamic Model and a control application of a Novel 6-DOF Hybrid Kinematics Manipulator. *IEEE Transactions on Mechatronics*.

Pott PP, Wagner A, Köpfle A, Badreddin E, Männer R, Weiser P, et al. (2004). *A handheld surgical manipulator: ITD - Design and first results.* Paper presented at the CARS, Chicago, Illinois, USA.

Pott PP, Weiser HP, Scharf H-P, & Schwarz MLR. (2004). A gearing mechanism with 4 degrees of freedom for robotic applications in medicine. *Biomedizinische Technik, 49*(6), 177-180.

Stewart D. (1965). A Platform with six Degrees of Freedom. *Proc. of Mech. Eng., 180*(1), 371-386.

Wagner A, Badreddin E, Weiser P, Köpfle A, Männer R, Pott PP, et al. (2004). *System Design and Position Control of a Handheld Surgical Robotic Device.* Paper presented at the Mechatronics & Robotics Conference, Aachen, Germany.

Wagner A, Pott PP, Köpfle A, Schwarz MLR, Scharf H-P, Weiser P, et al. (2006, 12.-14.9.2006). *Efficient inverse dynamics of a parallel robot with two movable platforms.* Paper presented at the MECHATRONICS 2006 - 4th IFAC-Symposium on Mechatronic Systems, Heidelberg.

Wenzel R, & Müller J. (1971). *Entscheidungsfindung in Theorie und Praxis*. Stuttgart: VDI-Seminar.

PREBUFFERING AS A WAY TO EXCEED THE DATA TRANSFER SPEED LIMITS IN MOBILE CONTROL SYSTEMS

Ondrej Krejcar
Centre for Applied Cybernetics, VSB Technical University of Ostrava, 17. Listopadu 15, Ostrava, Czech Republic
ondrej.krejcar@vsb.cz

Keywords: Wi-Fi, 802.11b, PDA, Localization, Framework, Predictive, Buffering.

Abstract: The proliferation of mobile computing devices and local-area wireless networks has fostered a growing interest in location-aware systems and services. Additionally, the ability to let a mobile device determine its location in an indoor environment supports the creation of a new range of mobile control system applications. Main area of interest is in model of radio-frequency based system enhancement for locating and tracking users of our control system inside buildings. The developed framework described here joins the concepts of location and user tracking as an extension for new control system. The experimental framework prototype uses a Wi-Fi network infrastructure to let a mobile device determine its indoor position. User location is used to data pre-buffering and pushing information from server to user's PDA. All the server data are saved as artefacts with their position info in building. These technique allow to exceed the data transfer speed limits in mobile control systems.

1 INTRODUCTION

The usage of various wireless technologies has increased dramatically and will continue for the coming years. This will lead to the rise of new application domains each with their own specific features and needs. Also, these new domains will undoubtedly apply and reuse existing (software) paradigms, components and applications in information and control systems. Today, this is easily recognized in the miniaturized applications on network-connected PDAs that provide more or less the same functionality as their desktop application equivalents. It is very likely that these new mobile application domains adapt new paradigms that specifically target the mobile environment. We believe that an important paradigm is context-awareness. Context is relevant to the mobile user, because in a mobile environment the context is often very dynamic and the user interacts differently with the applications on his mobile device when the context is changed. While a desktop machine usually is in a fixed context, a mobile device goes from work room, cross the building in company area, to work in-a-meeting, etc. Context is not limited to the physical world around the user, but also incorporates the user's behaviour, and terminal and network characteristics. Context-awareness concepts can be found as basic principles in long-term strategic research for mobile and wireless systems such as formulated in (WWRF, 2007). The majority of context-aware computing to date has been restricted to location-aware computing for mobile applications (location-based services). Our focus here is on position determination in an indoor environment. Location information is used to determine an actual user position and his future position. We have performed a number of experiments with the control system, focusing on position determination, and are encouraged by the results.

2 BASIC CONCEPTS

The proliferation of mobile computing devices and local-area wireless networks has fostered a growing interest in location-aware systems and services. A key distinguishing feature of such systems is that the application information and/or interface presented to the user is, in general, a function of his physical location. The granularity of location information needed could vary from one application to another. For example, locating a book in a library would require fine-grained information whereas locating a nearby room in company buildings area requires fairly coarse-grained location information. While much research has been focused on development of

services architectures for location-aware systems, less attention has been paid to the fundamental and challenging problem of locating and tracking mobile users, especially in in-building environments. In RF area we focus mainly on RF wireless networks in our research. Our goal is to complement the data networking capabilities of RF wireless LANs with accurate user location and tracking capabilities for user needed data pre-buffering. This property we use as information ground for extension of mobile control system to exceed the data transfer speed limits.

2.1 Data Collection

A key step in the proposed research methodology is the data collection phase. We record information about the radio signal as a function of a user's location. The signal information is used to construct and validate models for signal propagation. Among other information, the WaveLAN NIC makes available the signal strength (SS), which is reported in units of dBm. A signal strength of Watts is equivalent to 10*log10(s/0.001) dBm. For example, signal strength of 1 Watt is equivalent to 30 dBm. The WaveLAN driver extracts the SS information from the WaveLAN firmware each time a broadcast packet is received. Then the information is make available to user-level applications via system calls. It uses the wlconfig utility, which provides a wrapper around the calls, to extract the signal information.

2.2 Localization Methodology

The general principle is that if a Wi-Fi-enabled mobile device is close to such a stationary device – Access Point (AP), it can "ask" the location provider's position by setting up a Wi-Fi connection. If the mobile device knows the position of the stationary device, it also knows that its own position is within a range of this location provider (100 meters app.). Granularity of location can be improved by triangulation of two or several visible Wi-Fi APs. The PDA client will support the application in automatically retrieving location information from nearby location providers, and in interacting with the server. Naturally, this principle can be applied to other wireless technologies. The application (locator) is now implemented in C# using the MS Visual Studio .NET 2005 with .NET compact framework and a special OpenNETCF library enhancement (OpenNETCF, 2007). Schema on figure 1 describes a runtime localization process. The stars points are exactly measured and computed points of suppose user position. The real track on figure presents real movement of user during the time. The exact track mean computed track from measured Wi-Fi intensity level.

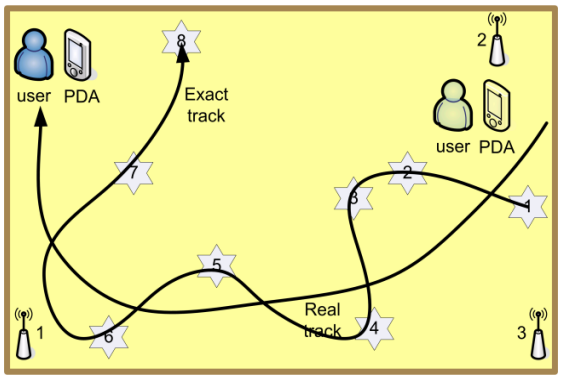


Figure 1: Localization principle - triangulation.

2.3 Predictive Data Push Technology

This part of project is based on a model of location-aware enhancement, which we used in debug control system. These info-data are used in developed framework to increase real dataflow from wireless access point (server side) to PDA (client side). Primary dataflow is enlarged by data pre-buffering. These techniques form the basis for predictive data push technology (PDPT). PDPT push data from information server to clients PDA to be on hand when user comes at desired location. The benefit of PDPT consists in reduction of time needed to display desired information requested by a user command on PDA. Time delay may vary from a few seconds to number of minutes. It depends on two aspects. First one is the quality of wireless Wi-Fi connection used by client PDA. A theoretic speed of Wi-Fi connection is max 687 kB/s. However, the test of transfer rate from server to client's PDA, which we have carried out within our Wi-Fi infrastructure provided the result speed from 43 to 160 KB/s on three various type of PDA (HTC Roadster, Blueangel and Universal). The second aspect is the size of copied data. We advice to use partitioned blocks from original data files or blocks.

2.4 Data Artefact Management

The PDPT Server SQL database manages the information (for example data about Ethernet hardware such as Ethernet switch, UTP socket, CAT5 cable lead, etc.) in the context of their location in building environment. This context

information is same as location information about user track. The PDPT core controls the data which are copied from server to PDA client by context information (position info). Each database artefact must be saved in database along the position info belongs to. The data artefact manager is used to manage these information.

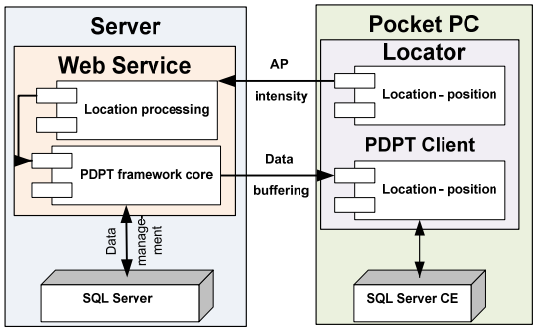


Figure 2: System architecture – UML design.

2.5 Framework Design

PDPT framework design is based on most commonly used server-client architecture. To process data the server has online connection to the control system. Data from technology are continually saved to SQL Server database (Tiffany, 2003) and (Reynolds, 2003). The part of this database (desired by user location or his demand) is replicated online to client's PDA where it is visualized on the screen. User PDA has location sensor component which continuously sends to the framework kernel the information about nearby AP's intensity. The kernel processes this information and makes a decision if and how a part of SQL Server database will be replicated to client's SQL Server CE database. The kernel decisions constitute the most important part of whole framework because the kernel must continually compute the position of the user and track and make a prediction of his future movement. After doing this prediction the appropriate data (part of SQL Server database) are pre-buffered to client's database for future possible requirements. The PDPT framework server is created as Microsoft web services to handle as bridge between SQL Server and PDPT PDA Clients.

2.6 PDPT Client

For testing and tuning of PDPT Core the PDPT Client application was created. This client realizes classical system and extension by PDPT and Locator module. Figure 3 show classical view of data presentation form SQL CE database to user (in this case the image of Ethernet network in company area.

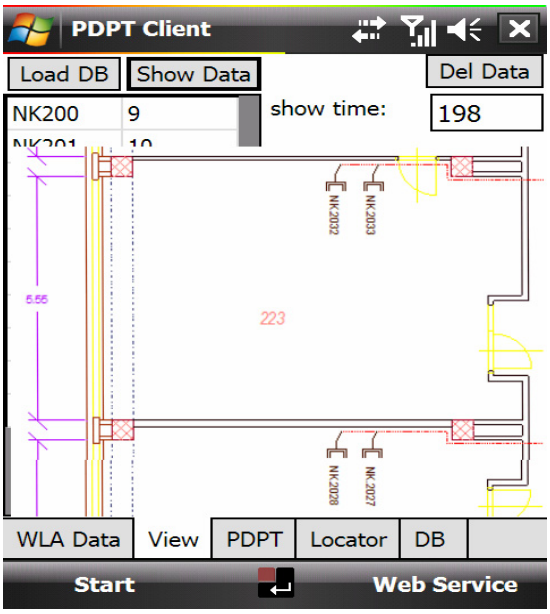


Figure 3: PDPT Client – Windows Mobile application.

3 EXPERIMENTS

We have executed a number of indoor experiments with the PDPT framework, using the PDPT PDA application. Wi-Fi access points are placed at different locations in buildings, where the access point cells partly overlap. We have used triangulation principle of AP intensity to get better granularity. It has been found that the location determination mechanism selects the access point that is closest to the mobile user as the best location provider. This technique partially uses a special Radius server (Radius, 2007) to realize "roaming" known in cell networks. Currently, the usability of the PDPT PDA application is somewhat limited due to the fact that the device has to be continuously powered. If not, the Wi-Fi interface and the application cannot execute the location determination algorithm, and the PDPT server does not receive location updates from the PDA client.

3.1 Data Transfer Tests using PDPT

The result of utilization of PDPT framework is mainly at data transfer speed reducing. The second test is focused on real usage of developed PDPT Framework and his main issue at increased data transfer. At table 1 are summary of eighteen tests

with three type of PDA and three type of data transfer mode. Each of these eighteen tests is fivefold reiterated for better accuracy. The data in table are average values from each iterations.

Table 1: Data transfer tests results.

no	Type	Mode	Data	Time	Speed
1	HTC Blue-angel	Sql CE	257	0.4	643
2		Sql CE	891	0.4	2228
3		Sql	257	5	51
4		Sql	891	13	69
5		PDPT	257	1.1	234
6		PDPT	891	3.2	278
7	HP iPAQ h4150	Sql CE	257	0.5	514
8		Sql CE	891	0.5	1782
9		Sql	257	5	51
10		Sql	891	14	64
11		PDPT	257	1.2	214
12		PDPT	891	3.7	241

The data mode column has three data transfer mode. The SQL CE mode represents the data saved at mobile device memory (SQL Server CE) and the data transfer time is very high. The second mode SQL means data which are stored at server (SQL Server 2005). Primary the data are loaded over Ethernet / Internet to SQL Server CE of mobile device and secondary the data are shown to user. The data transfers time consumption of this method is generally very high and the waiting time for user is very large. The third data mode PDPT is combination of previous two methods. The PDPT mode has very good results in form of data transfer acceleration. Realization of this test consists at user movement from location A to B at different way direction. Location B was a destination with requested data which are not contained at SQL CE buffer in mobile device before test.

4 CONCLUSIONS

The main objective of this paper is in the enhancement of control system for locating and tracking of users inside a building. It is possible to locate and track the users with high degree of accuracy. In this paper, we have presented the control system framework that uses and handles location information and control system functionality. The indoor location of a mobile user is obtained through an infrastructure of Wi-Fi access points. This mechanism measures the quality of the link of nearby location provider access points to determine actual user position. User location is used in the core of server application of PDPT framework to data pre-buffering and pushing information from server to user PDA. Data pre-buffering is most important technique to reduce time from user request to system response. The experiments show that the location determination mechanism provides a good indication of the actual location of the user in most cases. The median resolution of the system is approximately five meters. Some inaccuracy does not influence the way of how the localization is derived from the Wi-Fi infrastructure. For the PDPT framework application this was not found to be a big limitation as it can be found at chapter Experiments. The experiments also show that the current state of the basic technology used for the framework (mobile device hardware, PDA operating system, wireless network technology) is now at the level of a high usability of the PDPT application.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education of the Czech Republic under Project 1M0587

REFERENCES

Reynolds, J., 2003. *Going Wi-Fi: A practical Guide to planning and building an 802.11 Network*, CMP Books.

Wigley, A., Roxburgs, P., 2003. *ASP.NET applications for Mobile Devices.* Microsoft Press, Redmond.

Tiffany, R., 2003. *SQL Server CE Database Development with the .NET Compact Framework*. Apress.

Radius, 2007, http://www.ietf.org/, The Internet Engineering Task Force RADIUS Working Group.

Moore, R., Lopes, J., 1999. Paper templates. In *TEMPLATE'06, 1st International Conference on Template Production*. INSTICC Press.

WWRF, 2007, *http://www.wireless-world-research.org/* The Wireless World Research Forum

OpenNETCF, 2007, *http://www.opennetcf.org*, Open Library for Microsoft .NET Compact Framework.

A PERCEPTUAL MOTOR CONTROL MODEL BASED ON OUTPUT FEEDBACK ADAPTIVE CONTROL THEORY

Hirofumi Ohtsuka, Koki Shibasato
Department of Electronic Control, Kumamoto National College of Technology
2659-2 Suya, Koshi, Kumamoto, 861-1102, Japan
ohtsuka@ec.knct.ac.jp, sibasato@ec.knct.ac.jp

Shigeyasu Kawaji
Graduate School of Science and Technology, Kumamoto University
2-39-1 Kurokami, Kumamoto, 860-8555, Japan
kawaji@cs.kumamoto-u.ac.jp

Keywords: Human-machine system, adaptive control, output feedback, perceptual motor system, cerebellum.

Abstract: In this paper, a Perceptual Motor Control Model(PMCM) based on output feedback adaptive control theory is considered from the viewpoint of voluntary movement such as hand-tracking control. We give an account of the PMCM structure using together with both of the output feedback controller designed by using an almost strict positive real characteristics for the controlled plant and the Smith predictor for plant with pure time delay. In the proposed method, the attractive structural similarity exists between the cerebrum-cerebellum neuro-motor signal feedback loop and the adaptive controller - compensators local minor feedback loop. The proposed perceptual motor control model is evaluated through the comparison of between the experiment and the simulation of for handling 1-link mechanism in order to track an indicator.

1 INTRODUCTION

The construction of collaborative human-machine system is being recognized as an important technology from the viewpoint of human centered assisting system development (Takahashi and Ikeura, 2006; Yamada and Utsugi, 2006). While such assisting systems aim at partial replacement of control task or an amplification of control power, those have insufficiency in order to achieve the accurate maneuvering, where human performs as a main controller in the human-machine system. For the purpose of improvement of the maneuvering performance and the response of human-machine system, authors have developed a new compensator named as "collaborater", which can support the collaborative work of human and machine (Ohtsuka et al., 2007). The model of human response behavior is required to design the collaborater and the collaborative assisting system, but it has been difficult to construct an accurate model of human perceptual motor control system (e.g., limb and muscle). Kleinman et al. applied optimal control theory to develop a model of human behavior in manual tracking tasks (Kleinman et al., 1970). Their model contains time delay, a representation of neuromotor dynamics, and controller remnant as limitations. Recently, Furuta considers that the analysis of human control action is one of fundamental problems in the study of human adaptive mechatronics (Furuta et al., 2004). From such a viewpoint, in the authors' previous study, Delayed Feed-Forward (DFF) Model has been used for describing human's hand-tracking motion with visual information (Ishida and Sawada, 2003). The DFF model can realize the characteristics that the limb motion, with prediction of target position, makes the predicted value to minimize the transient error in the considering frequency range. However, for the non-cyclical target value and/or the controlled machine output, it has been resulted in that the DFF model has an insufficient reliance because of the shortage of consideration through the experimental study.

In this paper, for the upper limb motion in the hand-tracking control, a new Perceptual Motor Control Model (PMCM) is considered. Namely, the visual feedback controller is modeled as the output feedback type adaptive controller stabilizing the closed loop system based on an Almost Strict Positive Real (ASPR) characteristic of the controlled system. The Parallel Feed-forward Compensator (PFC) has been introduced in order to make an ASPR augmented system (Iwai et al., 1993). And, Miall et al. have

proposed a human's brain model by introduction of Smith Predictor (as forward internal model) in order to predict the consequences of actions and to overcome pure time delays of neuro-motor signal transmission associated with feedback control (Miall et al., 1993). So, taking into account of those approaches, both PFC and Smith Predictor are located into the minor feedback loop for the output feedback adaptive controller. So, the PMCM has similar structure to the cerebrum-cerebellum neuro-motor signal feedback loop. The effectiveness of the proposed PMCM is discussed through the comparison between the experiment and simulation results.

2 OUTPUT FEEDBACK TYPE ADAPTIVE CONTROL SYSTEM

In this section, as a preparation for discussion about the PMCM of human, we briefly outline an output feedback adaptive control method, where the controller is designed to realize the plant output converging to reference signal. Let us consider the following SISO plant:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + bu(t) \\ y(t) &= c^T x(t) \end{aligned} \tag{1}$$

where x is the nth order state vector, u and y are scalar input and output, respectively. A, b and c are unknown matrix and vectors with appropriate dimensions. The transfer function form of the plant Eq.(1) is expressed by

$$\begin{aligned} G(s) &= c^T(sI-A)^{-1}b + d = \frac{N(s)}{D(s)} \\ N(s) &= b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0 \\ D(s) &= s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0. \end{aligned} \tag{2}$$

Now, we make the following assumption.

Assumption 1. The Plant Eq.(1) or Eq.(2) is ASPR(Almost Strictly Positive real).

From this assumption, there exists a constant gain k_p such that the transfer function

$$G_c(s) = (1 + k_p G(s))^{-1} G(s) \tag{3}$$

is SPR(Strictly Positive Real). Sufficient condition for Assumption 1 can obtained, such that (1) $N(s)$ is Hurwitz polynomial, (2) $\gamma = n - m \leq 1$, and (3) $b_m > 0$ (Kaufman et al., 1998). Under the Assumption 1, the following adaptive algorithm:

$$u(t) = k(t)e(t) \tag{4}$$

$$\dot{k}(t) = ge(t)^2 \tag{5}$$

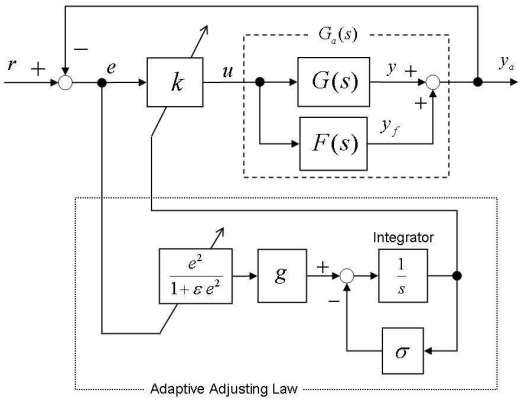


Figure 1: Adaptive Control System with PFC.

generates the control of the plant Eq.(1), where $e(t) = r(t) - y(t)$ and g is positive constant. And, it can achieve the output error $e(t)$ convergence to zero. Furthermore, against to the input disturbance and to the un-modeled dynamics of the plant, the following modified adaptive adjusting law

$$\dot{k}(t) = -\sigma k(t) + g\frac{e(t)^2}{1+\varepsilon e(t)^2}, \tag{6}$$

can be utilized in order to maintain that the all signals in the closed loop system become uniformly ultimate bounded (UUB), where σ and ε are given as sufficiently small positive constants (Iwai et al., 1993).

However, Assumption 1 is not satisfied by most practical systems with large relative degree $\gamma > 1$. To overcome this problem, as shown in Fig.1, dynamic compensator $F(s)$ is introduced to construct the augmented ASPR plant $G_a(s) = G(s) + F(s)$ satisfying the above-stated sufficient condition. Thus, the output feedback adaptive control law can be applied to the augmented plant $G_a(s)$ and maintain the stability of closed loop system. So, $F(s)$, located in parallel path for the plant, is called as parallel feedforward compensator (PFC) (Kaufman et al., 1998).

3 PERCEPTUAL MOTOR CONTROL MODEL

In the brain science research area, the cerebellum has attracted the attention of theorists and modelers, and the need for a unifying theory for the role of the cerebellum in motor control has been recognized for many years (Miall et al., 1993; Ito, 1970; Wolpert et al., 1998; Kleinman et al., 1970). Specially, based on data from the control of the primate arm in visually guided tracking tasks (Fig.2), Miall et al. have

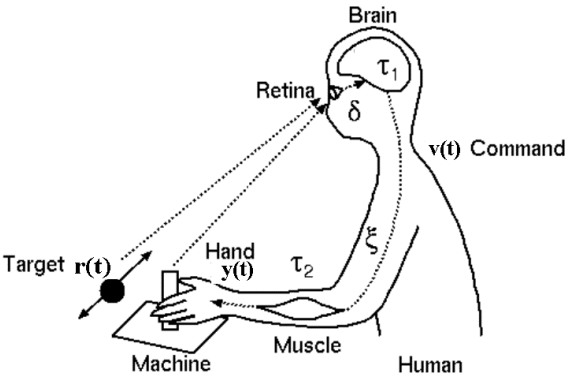


Figure 2: Human Body Dynamics.

Table 1: Parameters and Variables.

Notation	Parameters and Variables
$r(t)$	position of the target
$y(t)$	position of the hand
$v(t)$	command signal from the brain
δ	dead time in the nervous system from the retina to the brain
ξ	dead time in the nervous system from the brain to the muscle
τ_1	time constant of the brain
τ_2	time constant of the muscle dynamics

suggested that the cerebellum acts as a Smith Predictor, which is based on internal representation of controlled object suffering with long and unavoidable feedback delays. Ito et al. also suggested that there exists the cerebrum-cerebellum neuro-motor signal feedback loop (Fig.3) and the cerebellum may form the internal model, based on physiological and clinical evidence (Ito, 1970). There are two varieties of internal model, *i.e.*, forward model and inverse model (Wolpert et al., 1998). Forward models capture the forward or causal relationship between inputs to the system, such as the arm, and the outputs. The Smith predictor can be regarded as a kind of forward model. While the problem of pure time delay can be overcome by Smith predictor, the performance of visual feedback control is mainly affected by the setting of output feedback gain. However, conventional most of neuro motor models have fixed the feedback gain as constant. On the other hand, the adaptive control methods, based on the ability of animal to adapt itself to changes in its surroundings, have been developed.

Taking into account of both the concept of adaptive control method and the brain science researchers' suggestions, let us construct a new perceptual motor

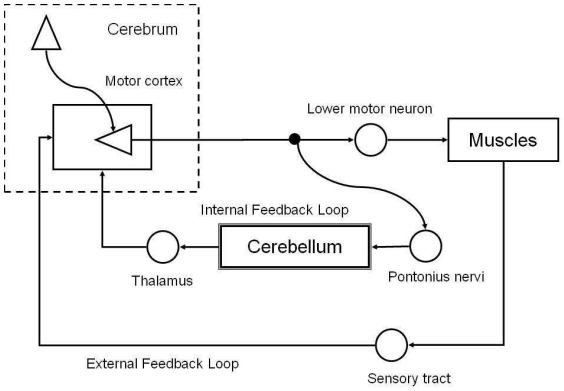


Figure 3: Cerebrum & Cerebellum (Ito, 1970).

control model as shown in Fig.4 for the control problem as shown in Fig.2, in which a human operator controls the machine to follow the target. Here, the time delay of nervous system transmission is successfully compensated by Smith predictor:

$$G_1(s) - G_1(s)e^{-(\delta+\zeta)s} \tag{7}$$

where $G_1(s) = G_P(s)/(\tau_2 s + 1)$.

Then, the controlled system from a side of the output feedback adaptive controller becomes a series of three elements. Namely, it consists a first lag element with time constant τ_1 which is involved in brain dynamics, a first lag element with time constant τ_2 which is involved in muscle dynamics, and a controlled machine dynamics $G_P(s)$. In order to construct a stable output feedback adaptive control system, the ASPR compensation must be implemented for $G(s) = G_1(s)/(\tau_1 s + 1)$. So, suppose that the following assumption holds.

Assumption 2. $G(s)$ satisfies that

(1) $G(s)$ is minimum phase system.

(2) the relative degree γ is larger than 2.

(3) the nominal value of the leading coefficient b_m of $G(s)$ is known.

Then, according to one of practical PFC design method (Iwai et al., 1994), PFC:$F(s)$ as shown in Fig.5 can construct the augmented plant $G(s)+F(s)$ which satisfies the above-mentioned sufficient condition for Assumption 1. Here, δ is sufficiently small positive constant, γ is a relative degree of $G(s)$, α_i are positive constants and β_i are coefficients of the Hurwitz polynomial:

$$R(s) = \beta_{\gamma-1}s^{\gamma-1} + \cdots + \beta_1 s + \beta_0, \tag{8}$$

where β_0 is a leading coefficient of $G(s)$. Both the Smith predictor and PFC can be located into the mi-

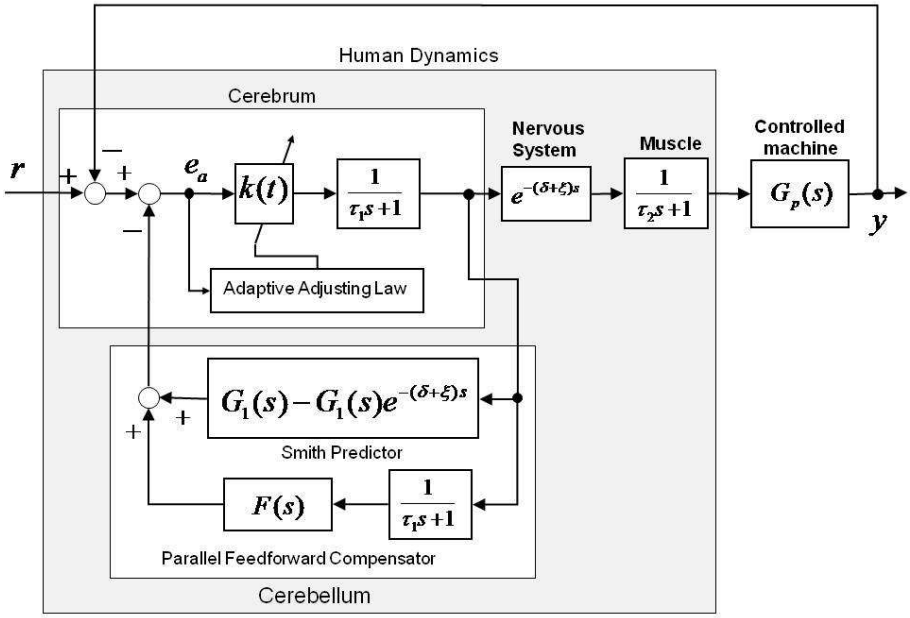


Figure 4: Perceptual Motor Control Model.

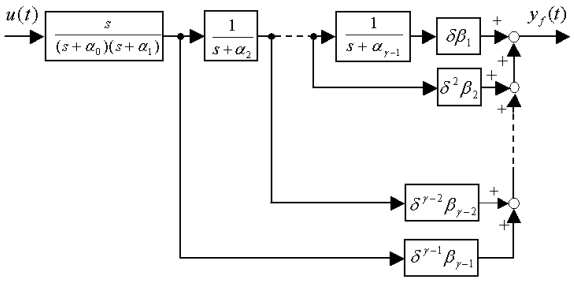


Figure 5: Ladder Network Type PFC (Iwai et al., 1994)

nor feedback loop for the output feedback gain k adjusted by the adaptive algorithm Eq.(6) using $e_a(t)$ instead of $e(t)$. So, it ease to recognize that such minor feedback structure is very similar to the cerebrum-cerebellum neuro-motor signal feedback loop model in Fig.3. Thus, we can imagine that the Smith predictor and PFC perform the role of cerebellum.

4 EVALUATION OF PMCM

Fig.6 shows the experimental equipment for the virtually guided tracking task. An indicator shows the target position, which is driven by AC motor 1, and the operator controls a handle to follow the indicator. AC motor 2 is assembled in order to generate the assisting torque for the operator, while it performs as a load inertia for human in this situation.

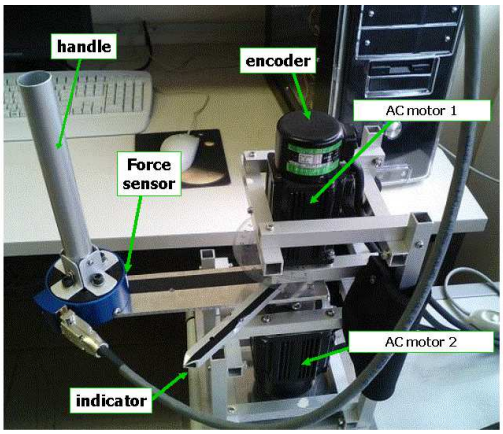


Figure 6: Experimental Equipment.

Mechanical System. From the experimental results of automatic positioning control, the transfer function of the one-link arm mechanism involving AC motor 2 was estimated as follows

$$G_P(s) = \frac{4213}{s(s+1)} \tag{9}$$

Human Dynamics Model. Through the experimental results, the parameters of human dynamics model are estimated such that $\delta+\zeta = 0.13$[s], $\tau_1 = \tau_2 = 0.03$[s],

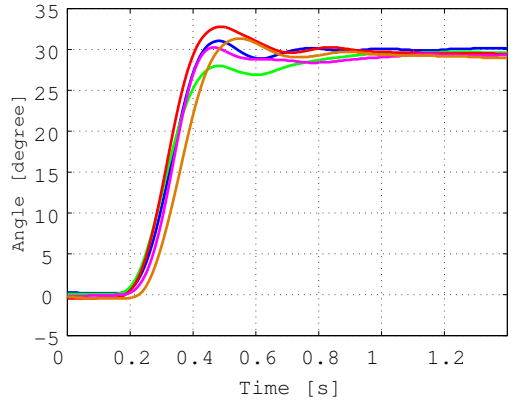


Figure 7: Experimental Results.

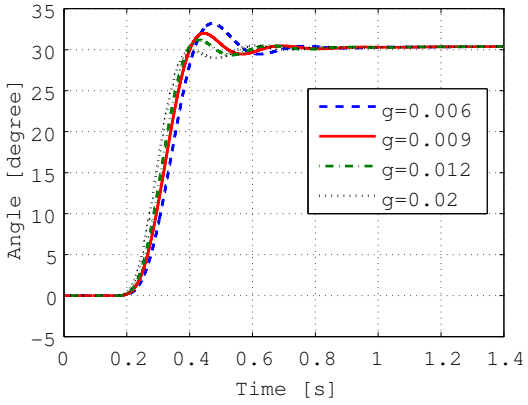


Figure 9: Simulation Results.

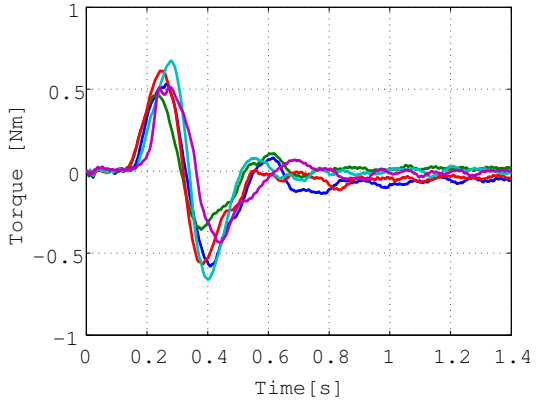


Figure 8: Input Torque (Experiment).

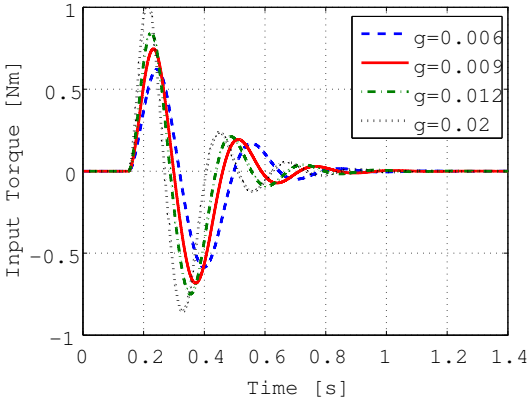


Figure 10: Input Torque (Simulation).

respectively (Saito and Nagasaki, 2002). In this case, the controlled system from a side of the output feedback controller, which is the above-mentioned series of three elements is given as follow.

$$G(s) = \frac{4213}{s(s+1)(0.03s+1)^2} \qquad (10)$$

PFC. Because $G_1(s)$ has a relative order as 3 and minimum phase characteristics, for the simulation, PFC is constructed as follows:

$$F(s) = \frac{f_1 s}{(\tau_1 s+1)(s+\alpha)^2} + \frac{f_2 s}{(\tau_1 s+1)(s+\alpha)}, \quad (11)$$

where design parameters are given as $f_1 = 350$, $f_2 = 6$, $\alpha = 0.5$.

Results of Experiment and Simulation. Experimental results for the target position $r(t) = 30$ [degree] are shown as Fig.7 and Fig.8. And, Fig.9 and Fig.10 also shows the simulation results for the variance of design parameter g in Eq.(6). In the simulation, the other parameters in Eq.(6) are given as $k(0) = 0$, $\sigma = 0.1$ and $\varepsilon = 0.01$. Although there exists some fluctuation in the experimental results obtained for three testors, we can recognize that the both responses are very similar. Because, by comparing between Fig.7 and Fig.9, the overshoots are almost same level and the damping ratio and the values of peak time are close resemblance. Furthermore, by comparison of Fig.8 and Fig.10, both signal wave forms also show a close similarity. So, we can note that the proposed model can maintain its good performance. Furthermore, we can set up a hypothesis such that the fluctuation in the response occurring every experiment can be interpreted as the fluctuation of PMCM parameters.

5 CONCLUSIONS

From the point aimed at the minor feedback loop in the brain, *i.e.*, the nervous network between the cerebrum and the cerebellum performing minor feedback

loop element, and a hypothesis for cerebellum generating a forward model of motor apparatus dynamics, a perceptual motor control model has been discussed. The proposed method is based on output feed back type adaptive control using ASPR characteristics of the controlled plant, which accompany with PFC. In the nervous network, there necessarily exists dead time (pure time delay) of signal transmission between cortex and lower apparatus. To overcome the influence of the feedback of the sensed signal involving time delay, the Smith predictor method is introduced.

From the viewpoint of the mutual connection between the cerebrum and the cerebellum, we showed that the PFC and Smith predictor perform as cerebellum generating a forward model for the controlled machine and human's motor apparatus, and the adaptive controller performs as cerebrum adjusting the visual feedback control signal. The effectiveness of the proposed model was examined through the comparison between of experimental results and simulation results for one-link arm positioning control problem. And, it was confirmed that the proposed model can represent the manual control response with sufficient accuracy.

REFERENCES

Furuta, K., Iwase, M., and Hatakeyama, S. (2004). Analysing saturating actuator in human-machine system from view of human adaptive mechatronics. In *Proceedings of REDISCOVER 2004, Vol.1, (3-1)–(3-9).*

Ishida, F. and Sawada, Y. (2003). Quantitative studies of phase lead phenomena in human perceptro-motor control system. In *Trans. of SICE, Vol.39, No.1, 59-66*.

Ito, M. (1970). Neurophysiological aspects of the cerebellar motor control system. In *International Journal of Neurology, Vol. 7, 162-176*.

Iwai, Z., Mizumoto, I., and Deng, M. (1994). A parallel feedforward compensator virtually realizing almost strictly positive real plant. In *Proc. of* 33^{rd} *IEEE CDC, 2827-2832*.

Iwai, Z., Mizumoto, I., and Ohtsuka, H. (1993). Robust and simple adaptive control system design. In *International Journal of Adaptive Control and Signal Processing, Vol.7, 163-181*.

Kaufman, H., Bar-Kana, I., and Sobel, K. (1998). *Direct Adaptive Control Algorithms Theory and Application*. Springer-Verlag, New York, 2nd edition.

Kleinman, D. L., Baron, S., and Levison, W. H. (1970). An optimal control model of human response part i: Theory and validation. In *Automatica, Vol.6, 357-369*.

Miall, R. C. ., Weier, D. J., Wolpert, D. M., and Stein, J. F. (1993). Is the cerebellum a smith predictor ? In *Journal of Motor Behavior, Vol.25, No.3, 203-216*.

Ohtsuka, H., Shibasato, K., and Kawaji, S. (2007). Collaborative control of human-machine system by collaborater. In *Trans. of The Japan Society of Mechanical Engineers, Series C, Vol.73, No.733, 2576-2582*.

Saito, H. and Nagasaki, H. (2002). *Clinical Kinesiology*. Ishiyaku Publishers, Inc., 3rd edition.

Takahashi, T. and Ikeura, R. (2006). Development of human support system. In *Journal of the Society of Instrument and Control Engineers,Vol.45, No.5, 387-388*.

Wolpert, D. M., Miall, R. C., and Kawato, M. (1998). Internal models in the cerebellum. In *Trends in Cognitive Sciences, Vol.2, No.9, 338-347*.

Yamada, Y. and Utsugi, A. (2006). Human intention inference techniques in human machine systems and their robotic applications. In *Journal of the Society of Instrument and Control Engineering, Vol.45, No.6, 407-412*.

PROGRESSIVE MESH BASED ITERATIVE CLOSEST POINTS FOR ROBOTIC BIN PICKING

Kay Boehnke and Marius Otesteanu
Communication Department, Polytehnica University Timisoara, Bd. Vasile Parvan, nr. 2, 300223 Timisoara, Romania
Kay.boehnke@gmail.com,marius.otesteanu@etc.upt.ro

Keywords: Iterative Closest Points, Surface registration, Progressive Mesh, laser range sensors, Object localization, Robotic bin picking.

Abstract: This paper describes a hierarchical registration process using the iterative closest point algorithm combined with a Progressive Mesh. To find the exact pose of objects in a robotic bin picking process we simulate the appearance of object poses and compare them with the real range data provided by laser range sensors. The coarse pose is estimated in a first step and then refined with the well known iterative Closest Point (ICP) algorithm combined with Progressive Meshes for hierarchical object localization. We evaluate our approach with different test scenarios and show the comprehensive potential of this idea for other registration problems.

1 INTRODUCTION

Today robots get more and more involved in industrial processes, because they are superior to man regarding requirements on strength, speed and endurance. Robotic automation processes became very successful in the last years, and offers a wide range for research. The task of robotic bin picking is easy to explain: Pick a known or unknown object out of a bin with an unsupervised industrial robot. This is called the "bin picking problem"(Hashimoto & Sumi,1999), (Katsoulas, 2005). It is also known as the de-palletizing problem, which occurs in nearly every industrial sector. The approach in this paper focuses on the object localization step, which is the most challenging step in the whole process. We introduce a simulation of a full laser scanning process. Industrial laser range sensors are modelled to transfer a cad-aided-design (CAD) model to a 2.5D range data representation. This virtual range data is aligned to the real range data of the scene with the help of combination of the Iterative Closest Point algorithm (Besl & McKay, 1992) and Progressive Meshes (Hoppe, 1996). Beside the significant improvement in speed our approach leads to better accuracy and robustness of the whole system. After the overview in section 2 the coarse pose estimation for a pose pre-selection is introduced in section 3. In the refinement step of our system we use the Progressive Mesh based Iterative Closest Point (PMICP) algorithm to derive optimal solution with high accuracy. We evaluate our approach with test range data in section 4 and conclude with upcoming extensions of our approach.

2 SYSTEM OVERVIEW

An overview of the proposed object localization system is shown in the figure 1.

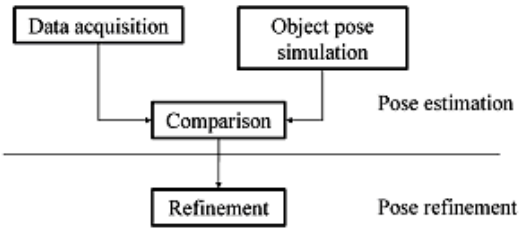


Figure 1: System overview.

The object localization is separated into pose estimation and refinement. To reduce the number of possible poses in the computational expensive refinement step, we make a pre-selection in the pose estimation step. The refinement step uses a modified registration algorithm to increase the accuracy. The components of our system are introduced in detail in the following sections.

3 POSE ESTIMATION

The purpose of the object pose estimation is to find adequate coarse positions of an object in the scene.

The object pose simulation creates a virtual range image (VRI) with help of a simulated sensor and a virtual scene points. The triangulated CAD-based object model is used to generate virtual range images with the help of the simulated range sensor. The sensor model virtually scans the object and produces a range image in the same way like the real scene. For every possible position and orientation a VRI is produced. This VRI is indexed with a known position and orientation of the model in a database. The range data representation of the VRI and the real range image (RRI) are datasets of three dimensional points with the position x, y and z. Every VRI is compared to the RRI determining the difference between the distance values z for acquired data from the range sensor and the simulated data with the following error function:

$$Error = \frac{1}{N}\sum_{i=0}^{x}\sum_{j=0}^{y}\left|z_{VRI}(i,j) - z_{RRI}(i,j)\right| \qquad (1)$$

The error is defined as the mean of the difference between every distance value Z_1 of the simulated object and the distance value Z_2 of the scene. We use this efficient calculation considering the fact, that both coordinate systems of RRI and VRI are equal due to knowledge of the real scene setup. Different VRI's for different kind of objects are compared to the RRI in the same way. So the object classification is integrated in the step of object localization. One advantage of this pre-selection of matching positions is the fact, that all VRI can be calculated offline and stored in a database. So the process for our coarse pose estimation can be summarized in that way:

- the RRI is delivered by the sensor
- all VRI in the database (one for each possible pose) are compared to the RRI
- these VRI with the best error value are selected for pose refinement

We take the VRI candidates within the best 10-20% of all error values in the coarse pose estimation. These VRI candidates are delivered to the pose refinement process, starting with the best matching candidate.

4 POSE REFINEMENT

In the previous section the coarse pose estimation creates an error value for every pose. The best VRI candidates were chosen and used as input for the pose refinement to find the best matching candidate. The task of the pose refinement is to find a nearly exact match between the object in the scene and the simulated image. The classical and most commonly used algorithm for determining rigid transformations is the Iterative Closest Point algorithm (Besl & McKay, 1992), (Chen & Medioni, 1992). Because of the slow convergence speed, the ICP was improved by many researchers (Rusinkiewicz & Levoy, 2001).

We use the ICP algorithm in combination with Progressive Meshes (Hoppe, 1996) to find the exact matching pose for every VRI candidate in the real scene captured by the laser range sensor. We call this combination Progressive Mesh Iterative Closest Point Algorithm (PMICP). The major problems of the ICP algorithm are the low performance calculating a huge amount of points in scene and model and its sensitivity to outliers (Rusinkiewicz and Levoy, 2001). In every iteration step all points of the two datasets must be compared to each other with a complexity of $O(N_{RRI}*N_{VRI})$ where N_{RRI} and N_{VRI} are the numbers points in the datasets. Our hierarchical approach now reduces this complexity by comparing only a reduced number of points N_i of each dataset. The representation of Progressive Meshes provides a highly efficient implementation for adjusting the level of detail in a point dataset and includes an inbuilt noise reduction. This representation is given by a set of meshes M_0 to M_n. M_0 is the mesh with the lowest accuracy and M_n is the mesh with the highest accuracy. The process of generating Progressive Meshes from point datasets is described in detail in the work of Hoppe et. al.(1993). In our experiments we choose the simplest way to connect the hierarchy of the Progressive Mesh representation to the ICP: The Progressive Mesh representation M_i is increased by a fixed increment and starts with a defined level of detail in each iteration step. The obvious advantage is the increased performance. But the profoundly effect is the increased robustness against outliers. By reducing the mesh up to M_0 outliers can no longer affect the result of the distance calculation. The shape of the model in representation of M_0 is similar to the M_0 representation of the scene representation. This leads to a very good initial position in the iterative process of the closest point algorithm. We have evaluated our PMICP with several experiments which are described in the next section.

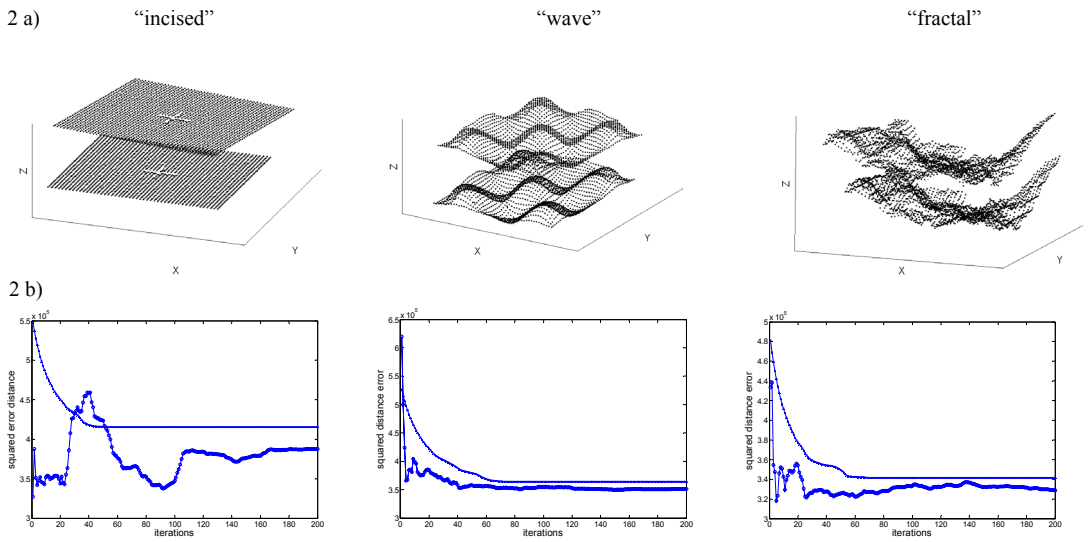


Figure 2: Test scenarios (2a) with convergence performance (2b).

5 EXPERIMENTAL RESULTS

It is obvious that some results depend on the used registration data and the initial pose. In our applications the ICP has to find the transformation between range images (RRI) and the simulated range images (VRI), which are more or less similar in shape to the “fractal” scenario of the following test scenarios. To evaluate our idea we used the reference datasets of Rusinkiewicz and Levoy (2001). We us the same test environment with synthetic meshs of 2000 points added with Gaussian noise and outliers. The datasets are shown in figure 2a. The “incised” dataset has two lines in shape of an “X” in the middle of a planar surface. This “wave” dataset is an easy scenario because of low frequency features and a smooth surface. In opposite to the wave scenario the “fractal” dataset represents landscape data of terrain registration and has features in all level of details.

5.1 Results

To compare our results we implemented the standard ICP algorithm according to (Besl & McKay, 1992) without approximation and any other possible improvements described by Rusinkiewicz and Levoy (2001). We changed the distance error calculation from Euclidian distance to

$$E = \sum (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \qquad (2)$$

to avoid computational floating point errors and increase calculation performance. We implemented the Progressive Mesh based Iterative Closest Point Algorithm (PMICP) based on the Progressive Mesh implementation of Hoppe in DirectX (Hoppe 1998). In our tests we start with an initial meshs consisting of five triangles and increase the number of vertices in the meshs by five triangles for each iteration step.

The initial M_i does not suffer from outliers like the standard ICP does. With the increasing iteration steps and the number of points in the datasets the PMICP implementation degenerates more and more to the reference algorithm with a better robustness. The convergence performance (figure 2b) of the “wave” scenario is similar to the “fractal” scenario. The algorithm outperforms the standard algorithm in the “incised” scenario over most iteration steps. The higher error between iteration step 20 and 40 shows the rotation ambiguity of the alignment of two plane surfaces. Especially in the first few iteration steps the PMICP aligns the datasets to a good initial pose. The squared distance error is always smaller when comparing to the standard ICP.

All this experiments concentrate on convergence robustness and final distance error issues. But the experiments show, beside the improvements mentioned above, that the overall performance of the refinement process can be increased significantly. The complexity of the ICP algorithm depends mainly on the number of points in the dataset. The search of the closest points has a computational complexity of $O(N_j * N_i)$. We reduce the number of

points in the dataset, starting with only a few points and increase the number in every iteration step. The computational complexity is reduced in average to $O((0.5*N_j)*(0.5*N_i))$ assuming we do not stop the iteration until we reach the end (M_n mesh).

If the iteration process is stopped, because the ICP reached the minimum, the performance of our implementation is always better than $O((0.5*N_j)*(0.5*N_i))$. Our Progressive Mesh ICP experiments need in average 25% of the time of the standard ICP implementation.

6 FURTHER EXTENSIONS

Many of known modifications of the ICP can be combined with the PMICP without loss of generalization. For example the performance in the closest points search is often increased using Kd-Trees implementations (Z. Zhang, 1994) to $O(N_j*log(N_i))$. The Kd-Tree search could be used in addition to our PMICP leading to a significant improvement in speed especially in the higher level of details. The ICP algorithm is known to be very sensitive to wrong initial poses of the two datasets because of the fact, that the ICP will always converge to the local minimum (which is of course commonly not identical to the global minimum). So the determination of the optimal initial start value for the number of points in the mesh is very important. In the current implementation the level of detail in the Progressive Mesh is connected to number of iterations in the ICP. Finding the optimal number of points for each iteration step in ICP iterations is one of possible improvements in the next steps of our research.

7 CONCLUSIONS

We described a system to align range data surfaces in a context of industrial process automation. We focused on the improvements in the refinement step of our hierarchical object localization system. The well known and proven ICP Algorithm is modified with the use of Progressive Meshes. To be sure to meet the requirements of different applications we evaluated our system with test scenarios, which cover many types of possible range data scenes.

The simulation of real scenes offers the possibility to use our approach in many scenarios. The described two-step object localization is integrated in our system robotic bin picking covering different application scenarios (Boehnke, 2007).

ACKNOWLEDGEMENTS

The author would like to thank S. Rusinkiewicz for providing the datasets. This work was supported in part by VMT (Pepperl+Fuchs Group) in Mannheim/ Germany. The author wants to thank P. Roebrock, M. Kleinkes, W. Neddermeyer, W. Winkler, and K. Lehmann for their support in this project.

REFERENCES

Besl, J.P., N.D. McKay, 1992. A Method for Registration of 3-D Shapes, In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol 14, pp. 239-256

Boehnke, K., 2007. Object localization in range data for robotic bin picking, In *IEEE Proceedings on Automation Science and Engineering*, pp. 572-577

Chen, Y., Medioni, G., 1992. Object Modelling by Registration of Multiple Range Images, In *Image and Vision Computing*, vol. 10, pp. 145-155

Hashimoto, M., Sumi, K., 1999. 3d object recognition based on integration of range image and grey-scale image, In *Proc. British Machine Vision Conference* pp. 253–262

Hoppe, H., 1996. Progressive Meshes, In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pp. 99–108.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W., 1993. Mesh optimization. In *Computer Graphics (SIGGRAPH '93 Proceedings)* pp 19-26

Hoppe, H., 1998. Efficient Implementation of Progressive Meshes, In *Comp.. & Graphics Vol.22/1*, pp. 27-36

Katsoulas, D., 2004. Robust recovery of piled box-like objects in range images, *Ph.D. dissertation*, Dept. Computer Science, Freiburg Univ., Germany

Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the ICP algorithm, In *Proc. of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pp. 145–152

Zhang, Z.,1994. Iterative point matching for registration of free-form curves and surfaces. In *the International Journal of Computer Vision*, 13(1):119–152.

HYBRID MATCHING OF UNCALIBRATED OMNIDIRECTIONAL AND PERSPECTIVE IMAGES

Luis Puig, J. J. Guerrero
DIIS-I3A, University of Zaragoza, Zaragoza, Spain
lpuig@unizar.es, jguerrer@unizar.es

Peter Sturm
INRIA Rhône-Alpes, Montbonnot, France
Peter.Sturm@inrialpes.fr

Keywords: Computer vision, matching omnidirectional images, hybrid epipolar geometry.

Abstract: This work presents an automatic hybrid matching of central catadioptric and perspective images. It is based on the hybrid epipolar geometry. The goal is to obtain correspondences between an omnidirectional image and a conventional perspective image taken from different points of view. Mixing both kind of images has multiple applications, since an omnidirectional image captures many information and perspective images are the simplest way of acquisition. Scale invariant features with a simple unwrapping are considered to help the initial putative matching. Then a robust technique gives an estimation of the hybrid fundamental matrix, to avoid outliers. Experimental results with real image pairs show the feasibility of that hybrid and difficult matching problem.

1 INTRODUCTION

Recently, a number of catadioptric camera designs have appeared. The catadioptric cameras combine lenses and mirrors to capture a wide, often panoramic field of view. It is advantageous to capture a wide field of view for the following reasons. First, a wide field of view eases the search for correspondences as the corresponding points do not disappear from the images so often. Second, a wide field of view helps to stabilize egomotion estimation algorithms, so that the rotation of the camera can be easily distinguished from its translation. Last but not least, almost complete reconstructions of a surrounding scene can be obtained from two panoramic images (Svoboda and Pajdla, 2002).

A hybrid image matching is a system capable of establish a relation between two or more images coming from different camera types. The combination of omnidirectional and perspective images is important in the sense of that a single omnidirectional image contains more complete description of the object or place it represents than a perspective image thanks to its wide field of view and its ability to unfold objects.

In general if we want to establish a relation between two or more views obtained from different cameras, they must have a common representation where they can be compared. We face this problem when we want to match omnidirectional images with perspective ones. Few authors have dealt with this problem and they followed different strategies. In (Menem and Pajdla, 2004) Plücker coordinates are used to map $\wp^2$ into $\wp^5$ where lines and conics are represented in 6-vectors and a geometrical relation can be established. A change of feature space is introduced in (Chen and Yang, 2005). The perspective image is partitioned into patches and then each patch is registered in the Haar feature space. These approaches have a major drawback, they require information about the geometry of the particular mirror used to get the images, which in most cases is not available. Besides they perform hard transformations over the image.

Another approach to deal with this problem is to establish a geometric relation between the images. Sturm (Sturm, 2002) develops this type of relation between multiple views of a static scene, where these views are obtained from para-catadioptric and perspective cameras.

We propose to explore a way to overcome these drawbacks avoiding the use of the so complex geometry of the mirror and the formulation of a geomet-

ric model. We present an automatic hybrid matching approach with uncalibrated images using the hybrid epipolar geometry to establish a relation between omnidirectional and perspective images.

2 HYBRID IMAGE MATCHING USING EPIPOLAR GEOMETRY

To deal with the problem of the robust hybrid matching a common strategy is to establish a geometrical relation between the views of the 3D scene. We have selected a strategy that does not require any information about the mirror. A geometrical approach which encapsulates the projective geometry between two views is used. Epipolar geometry (EG) is the intrinsic projective geometry between two views. It is independent of the scene structure, and only depends on the cameras' internal parameters and relative pose (Hartley and Zisserman, 2000). This approach needs pairs of putative corresponding points between the views. In this work we use the SIFT descriptor (Lowe, 2004).

Sturm proposes a hybrid epipolar geometry, where a point in the perspective image is mapped to its corresponding epipolar conic in the omnidirectional image. Recently Barreto and Daniliidis (Barreto and Daniilidis, 2006) have exposed a more general scheme where they compare the mixture of central cameras, including pin-hole, hyperbolic and parabolic mirrors in catadioptric systems and perspective cameras with radial distortion.

The fundamental matrix $\mathbf{F}$ encapsulates the epipolar geometry. The dimension of this matrix depends on the image types we want to match. In the hybrid case we have two options, a 4×3 matrix in the case of para-catadioptric and perspective cameras or 6×3 in the case of hyperbolic mirror and perspective cameras in a catadioptric system. In (Barreto and Daniilidis, 2006) for this last case a 6×6 matrix is considered, which can result in a very difficult corresponding estimation problem.

2.1 EG with Perspective and Catadioptric Cameras

In general the relation between omnidirectional and perspective images with the fundamental matrix can be established by

$$\hat{\mathbf{q}}_c^T \mathbf{F}_{cp} \mathbf{q}_p = 0 \qquad (1)$$

subscripts p and c denote perspective and catadioptric respectively.

From Eq.1 with known corresponding points between the two images, we can derive the *hybrid fundamental matrix*. Points in the perspective image are defined in common homogeneous coordinates. Points in the omnidirectional image are defined depending on the shape of the epipolar conic. The general representation for any shape of epipolar conic is a 6-vector. A special case where the shape of the conic is a circle the coordinate vector has four elements. These representations are called the "lifted coordinates" of a point in the omnidirectional image.

In the hybrid epipolar geometry points in the perspective image are mapped to its corresponding epipolar conic in the omnidirectional image. Conics can be represented in homogeneous coordinates as the product $\hat{\mathbf{q}}^T \mathbf{c} = 0$, where $\hat{\mathbf{q}}^T$ represents the lifted coordinates of the omnifirectional point $\mathbf{q}$. In this work we have two representations for this point, one of them is the general homogeneus form of a conic, a 6-vector $\hat{\mathbf{q}} = (q_1^2, q_2^2, q_3^2, q_1 q_2, q_1 q_3, q_2 q_3)^T$. The other one constraints the shape of the conic to be a circle $\hat{\mathbf{q}} = (q_1^2 + q_2^2, q_1 q_3, q_2 q_3, q_3^2)^T$. These representations are called the "lifted coordinates" of the omnidirectional point $\mathbf{q}$.

If the point in the omnidirectional image is represented with a 6-vector lifted coordinates $\hat{\mathbf{q}}$, the fundamental matrix is 6×3 (F63) in such a way that $\mathbf{c} \sim \mathbf{F}_{cp} \mathbf{q}_p$. When the 4-vector lifted coordinates is used the fundamental matrix is 4×3 and the conic (circle) is obtained by the same product $\mathbf{c} \sim \mathbf{F}_{cp} \mathbf{q}_p$.

2.2 Computation of the Hybrid Fundamental Matrix

The algorithm used to compute the fundamental matrix is similar to the 8-point algorithm (Hartley and Zisserman, 2000) for the purely perspective case, with the difference that the points in the omnidirectional images are given in lifted coordinates.

The automatic computation of the fundamental matrix is summarized as follows:

1. **Initial Matching**. Scale invariant features are extracted in each image and matched based on their intensity neighborhood.
2. **RANSAC Robust Estimation**. Repeat for n samples, where n is determined adaptively:
   (a) Select a random sample of k corresponding points, where k depends on what model we are using (if F43, $k = 11$ or if F63, $k = 17$)[1]. Compute the hybrid fundamental matrix $\mathbf{F}_{cp}$ as described above.

[1] Matrices are up to scale, so we need the number of elements of the matrix minus one corresponding points.

(b) Compute the distance d for each putative correspondence, d is the geometric distance from a point to its corresponding epipolar conic (Sturm and Gargallo, 2007).

(c) Compute the number of inliers consistent with $\mathbf{F}_{cp}$ by the number of correspondences for which $d < t$ pixels.

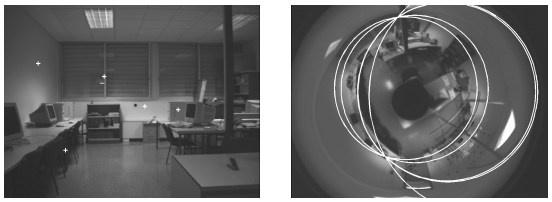
Choose the $\mathbf{F}_{cp}$ with the largest number of inliers.

The $\mathbf{F}_{cp}$ is used to eliminate outliers which are those point correspondences for which $d > t$.

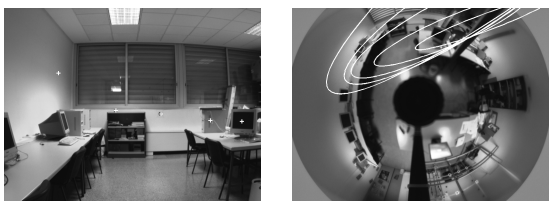
3 HYBRID EG EXPERIMENTS

In this section we present experiments performed with real images obtained from two different omnidirectional cameras. The first one is a model composed by an unknown shape mirror and a perspective camera. The second one is composed by a hyperbolic mirror and a perspective camera.

The purpose of this experiment is to show the performance of this approach to compute the hybrid epipolar geometry in real images. We use 40 manually selected corresponding points to compute it. In order to measure the performance of $\mathbf{F}$ we calculate the geometric error from each point to its corresponding epipolar conic. Fig.1(a) shows the epipolar conics computed with the F43 matrix using the unknown shape mirror. Fig. 1(b) shows the epipolar conics computed with the F63 matrix using the hyperbolic mirror. The mean of distances from the points to their corresponding epipolar conics are 2.07 pixels and 0.95 pixels respectively.

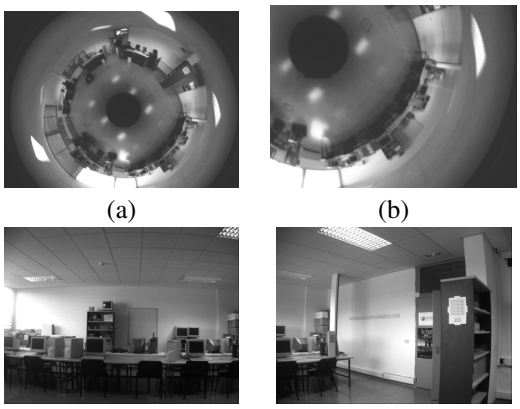


(a) Experiment with the unknown shape mirror using F43.



(b) Experiment with the hyperbolic using F63.

Figure 1: Experiments with real images using both mirrors and both hybrid fundamental matrices.



(a) (b) (c) (d)

Figure 2: Set of images used to test the automatic approach. (a) Omnidirectional image. (b) An image cut from (a). (c) Perspective1 image. (d) Perspective2 image.

4 AUTOMATIC MATCHING

In this section we present some experiments performing automatically the matching between an omnidirectional and a perspective image. There exist approaches matching automatically omnidirectional images using scale invariant features (Murillo et al., 2007) which demonstrate a good performance, but it does not work with hybrid matching. An easy way to transform an omnidirectional image into a common environment with the perspective one is to unwrap the omnidirectional image, which consists in a mapping from Cartesian to polar coordinates.

The initial matching between the perspective and the unwrapped omnidirectional image has a considerable amount of inliers but also a majority of outliers. This scenario is ideal to use a technique like RANSAC where we have a set of possible correspondences of points useful to compute the hybrid epipolar geometry.

One omnidirectional image, a part of it and two perspective images, all of them uncalibrated, are used to perform the following experiment. We use the algorithm explained in section 2.2. Table 1 summarizes the results of this experiments giving the quantity of inliers and outliers in the initial and the robust matching. For example, in Experiment 1 we use images Fig.2(a) and Fig.2(c). The initial matching gives a 35% of inliers. After applying the robust estimation we obtain a 80% of inliers. Notice that just 2 inliers have been eliminated. Fig. 3 shows the final matches.

The results show that the epipolar geometry eliminates most of the outliers. For practical reasons in the estimation problem, we use the F43, the simplest model of the hybrid epipolar fundamental matrix.

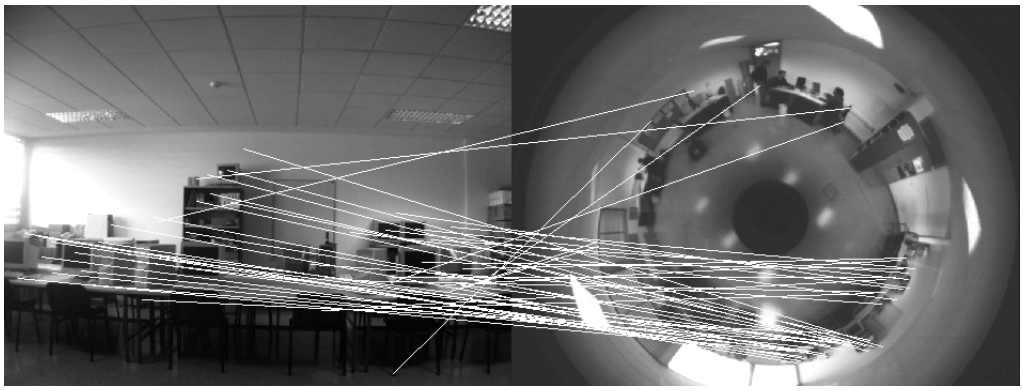


Figure 3: Matching between omnidirectional and perspective image using the unwrapping and the hybrid EG as a tool.

Table 1: Numerical results of the matches using the set of images.

	Omni SIFT	Persp SIFT	Initial Matches (inliers/outliers)	Robust EG matches (inliers/outliers)
Experiment 1	636	941	37/70	35/9
Experiment 2	636	1229	19/36	16/11
Experiment 3	246	941	36/11	34/4
Experiment 4	246	1229	20/14	16/6

5 CONCLUSIONS

In this work we have presented an automatic wide-baseline hybrid matching system using uncalibrated cameras. We have performed experiments using real images with different mirrors and two different ways to compute the hybrid fundamental matrix, a 6-vector generic model and a reduced 4-vector model. The last one has demonstrated be useful in order to require fewer iterations to compute a robust fundamental matrix and having equal or even better performance than the 6-vector approach. We also prove that an easy polar transformation can be a useful tool to perform a basic matching between omnidirectional and perspective images. Finally the robust automatic matching proved its efficiency to match an omnidirectional image and a perspective image, both uncalibrated.

ACKNOWLEDGEMENTS

This work was supported by project NERO DPI2006 07928 and DGA(CONSI+D)/CAI.

REFERENCES

Barreto, J. P. and Daniilidis, K. (2006). Epipolar geometry of central projection systems using veronese maps. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1258–1265, Washington, DC, USA. IEEE Computer Society.

Chen, D. and Yang, J. (2005). Image registration with uncalibrated cameras in hybrid vision systems. In *WACV/MOTION*, pages 427–432.

Hartley, R. I. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110.

Menem, M. and Pajdla, T. (2004). Constraints on perspective images and circular panoramas. In Andreas, H., Barman, S., and Ellis, T., editors, *BMVC 2004: Proceedings of the 15th British Machine Vision Conference*, London, UK. BMVA, British Machine Vision Association.

Murillo, A. C., Guerrero, J. J., and Sagües, C. (2007). Surf features for efficient robot localization with omnidirectional images. In *2007 IEEE International Conference on Robotics and Automatio, Roma*.

Sturm, P. (2002). Mixing catadioptric and perspective cameras. In *Workshop on Omnidirectional Vision, Copenhagen, Denmark*, pages 37–44.

Sturm, P. and Gargallo, P. (2007). Conic fitting using the geometric distance. In *Proceedings of the Asian Conference on Computer Vision, Tokyo, Japan*. Springer.

Svoboda, T. and Pajdla, T. (2002). Epipolar geometry for central catadioptric cameras. *Int. J. Comput. Vision*, 49(1):23–37.

THE ROLE OF SENSORY-MOTOR COORDINATION
Identifying Environmental Motion Dynamics with Dynamic Neural Networks

Stephen Paul McKibbin, Bala Amavasai, Arul N. Selvan, Fabio Caparrelli and W. A. F. W. Othman
Microsystems and Machine Vision Laboratory, Materials and Engineering Research Institute
Sheffield Hallam University Pond Street, Sheffield S1 1WB, U. K.
Stephen.p.mckibbin@student.shu.ac.uk

Keywords: Sensory-motor coordination, Particle Swarm Optimisation, Dynamic Neural Networks, Mobile robot controllers, Bio-inspired controllers.

Abstract: We describe three recurrent neural architectures inspired by the proprioceptive system found in mammals; Exo-sensing, Ego-sensing, and Composite. Through the use of Particle Swarm Optimisation the robot controllers are adapted to perform the task of identifying motion dynamics within their environment. We highlight the effect of sensory-motor coordination on the performance in the task when applied to each of the three neural architectures.

1 INTRODUCTION

In situated agents, the actions that they perform are the pre-cursor for the senses that they experience which, in turn, are the basis for their next action. Often it is assumed that senses are read and then actions are made. It has been suggested that the coordination of the action is as important as the sensing and that the close coupling of these behaviours is fundamental to building complex behaviours (Nolfi, 2002b) and even knowledge (O'Regan, 2001).

We investigate the task of a mobile robot being able to identify the dynamics of a moving target in its environment using only local information. In nature this is an important skill that enables animals to hunt prey, evade predators and also to communicate with gesture or dance. The work is an extension of the experiment by (McKibbin *et al*, submitted for review) where three recurrent neural architectures are evaluated however in this paper, a comparison is drawn between those controllers that are allowed to invoke Sensory-Motor Coordination (SMC) in their motion strategy and those that are not. The controllers have been designed in such a way that they can be conceptually defined by a number of features. This definition makes the study of the effect of each feature more apparent.

The controllers that are prevented form using SMC are given a pre-trained set of weights that control their movement and these weights are not adpated throughout the optimisation process. Each controller is given the same pre-trained weights allowing a comparison to be drawn between them and also with the controllers that are free to adapt their motion strategy. The work carried out in this paper is an extension of previous work by the author (McKibbin *et al*, submitted for review) and focuses on the role of SMC in simplifying or complicating a task that requires some amount of deliberative processing.

2 TASK DESCRIPTION: IDENTIFYING MOTION DYNAMICS

The task under investigation in this paper requires a mobile robot to discriminate and identify the two phases of the trajectory of a moving target object using only local information (McKibbin *et al*, submitted for review).

The robot used in the task is a simulated version of the Khepera II robot from k-team, it is cylindrical in shape with a diameter of 32mm and it is simulated in the Webots 3D fast prototyping software package from Cyberbotics. The robot has two wheels controller by independent motors (m0 and m1) that, when spun in opposite directions, allow the robot to rotate on the spot. It has 8 IR sensors (ds0 – ds7)

distributed around its perimeter and the particular configuration of the sensors is shown in figure 1.

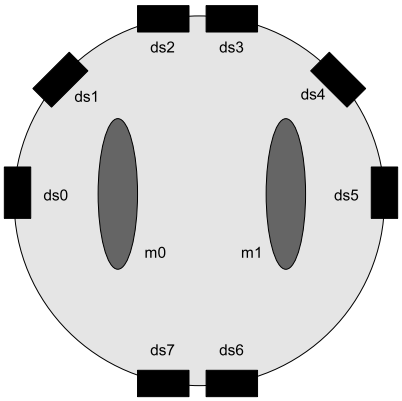


Figure 1: A functional diagram of the Khepera robot showing the configuration of the IR sensors (ds0 – ds7) and the motor driven wheels (m0, m1).

The target object moves in a bounded arena with a constant "figure of 9" trajectory as shown in figure 2. It moves with a constant speed and takes no input from the environment and will not stop if confronted with an obstacle *i.e.* a robot. The size and shape of the target are approximate to that of the robot, being cylindrical in shape with a radius of 32mm. Since the shape of the target is cylindrical its sensory profile will remain the same from which ever angle it is sensed and on its trajectory as it changes direction and turns corners this sensory profile will remain unchanged from the point of view of the robot. As a result of this uniformity of shape, there is only one distinguishing feature of the target and that is the dynamics of its motion plan; the path of its trajectory.

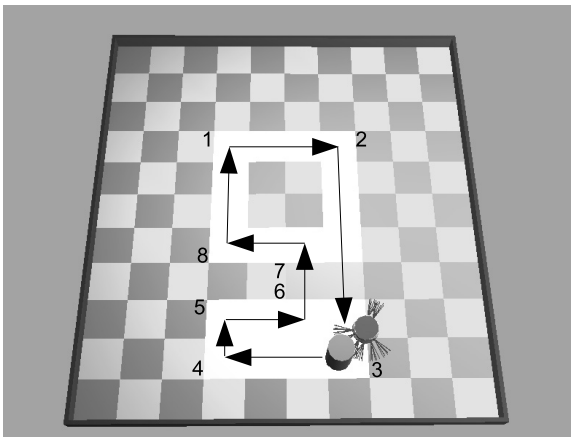


Figure 2: Screenshot of the arena showing the robot next to the target. The arrows indicate the target's straight edged "figure of 9" trajectory.

The targets trajectory is considered to have two phases, the lower part of the loop, *phase0*, comprising of a flat horizontal rectangular shape, and the upper part, *phase1*, comprising of a regular square-like shape. The task for the robot is to follow the target and to decide which phase of the trajectory it is currently executing and to display this using an output LED. When the LED is switched on it denotes *phase1* and when it is off it denotes *phase0*.

The task has 2 parts;

1) Follow the target through the "figure of 9" loop, keeping it within sensory range
2) Indicate at each time step which phase of its trajectory the target is currently performing

Part 1 of the task is a predicate for part 2. Since the only sensory information available to the robot is that provided by its IR sensors, in order to decide which part of the loop the target is in at any given time, the robot should be able to sense it. The only way the robot can sense the target is when it is at close range (<50mm). As detailed in Section 4.2, the robot needs to remain close to the target at each time step in order to gain fitness.

Considering part 2 of the task, the robot must be able to discriminate the two phases of the trajectory using only local information. As we have already described, the sensory profile of the target remains constant throughout each phase. The target moves in straight lines and takes corners at 90 degrees for each turn. The transition from one phase to another is performed in a straight line through the grey banded "no man's land". This locally uniform motion does not give any clue to the transition between phases. In view of these constant and regular conditions, there are no explicit signals or sensory states presented to the robot to aid it in its identification task. There is no single sensory state afforded by its environment that would allow the robot to distinguish the two phases of the targets movement. The robot must incorporate an ability to add context to its current sensory information and, depending on the context, identify the current trajectory of the target.

3 RECURRENT NEURAL NETWORK CONTROLLERS

The neural networks examined for this task are DNNs with update functions that take into account previous activation levels when producing new activations. The architectures of the networks are

inspired by a rough model of how the human body uses internal and external senses. The use of the combination of external and internal senses is called proprioception and it is this feedback system that allows the human body to modulate its behaviour. For example, it is through the use of proprioception that we are able to touch our nose with our finger whilst we have our eyes closed. Motor commands are sent to the muscles to cause actions and so too sensory signals are returned for processing to provide closed loop control. The mechanisms used to process these flows and contra-flows of information are still active areas of research in biology.

We describe 3 types of DNN below, each of which have an input layer, a fully connected hidden layer and an output layer. Each of the 3 architectures uses different types of recurrent feedback. We have named the 3 types of DNN Ego-sensing, Exo-sensing and composite. They are so named due to the type of sensing they employ;

1. The Ego-sensing controller takes the output of the previous motor actions as inputs to the hidden layer.
2. The Exo-sensing controller takes inputs to the hidden layer only from the IR sensors.
3. The Composite controller uses inputs to the hidden layer from both the motor actions and the IR sensors.

The input layer consists of 6 input nodes each connected to one of the 6 frontal IR sensors of the robot. The actual input value is the IR activation normalised in the range [0, 1]. There is also a bias node which provides a constant input of 1. This layer feeds forward only to the motor outputs in the Ego-sensing architecture, in the Exo-sensing and Composite architecture it also feeds forward to the hidden layer and the IDU output. The hidden layer consists of 5 nodes that are fully connected to each other also with recursive connections that encode the hidden node activation at the previous time step. The input to the layer is the weighted sum of all its inputs and each node operates with the logistic transfer function however the output of each node is both a function of its current inputs and its previous output (Nolfi, 2002a). In each of the 3 types of controller this layer feeds forward to the IDU output but does not connect to the motor outputs. The update equation for the hidden nodes is given in (1).

$$hidden\_unit\_out_i = \sqrt{(mem\_coef_i * hidden\_unit\_out_i)} + (1 - \sqrt{(mem\_coef_i * hidden\_unit\_out_{i(t-1)})} \quad (1)$$

Where $hidden\_unit\_out_i$ is the output of the current hidden node in the supervisory layer and mem_coef_i is the memory coefficient associated with the current node in the supervisory layer. The memory coefficient for each hidden node is an encoded parameter in the PSO algorithm and is bound in the range [0, 1]. This parameter determines to what extent a hidden node is affected by its current inputs and its previous outputs. This has the effect of altering how quickly a particular node reacts to changes at its inputs.

3.1 Ego-Sensing Controller

The architecture for the Ego-sensing controller consists of 2 parts. The first part is a purely reactive system that connects the sensors at the input directly to the motors with weighted connections. The second part consists of a fully connected hidden layer containing recursive feedback loops that takes input from the sensors and feeds forward to the Identification Unit (IDU). In an analogy with natural systems, the first part is similar to the reflex system found in mammals where motor actions are coupled closely to sensory input and the second part is loosely based on the afferent signal feedback system in proprioception that processes self-initiated motor actions. Figure 3 shows the reactive part on the left with the connections directly from the sensors to the motor shown with the thick arrow and the deliberative part on the right with connections from the outputs of the motors and the IDU feeding back into the hidden layer. These connections are weighted and they connect to each node in the hidden layer. The hidden layer only connects to the IDU and is thus the decision maker.

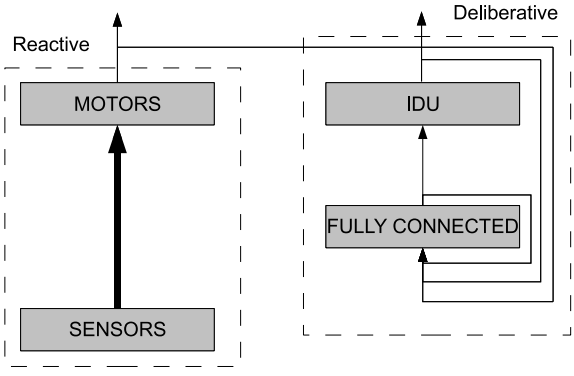


Figure 3: The architecture of the Ego-sensing controller. The thick arrow indicates the reactive part of the network.

3.2 Exo-Sensing Controller

The Exo-Sensing controller also has two parts, the reactive part with its direct coupling to the motors and the deliberative part which processes information over time that feeds forward to the IDU. In the Exo-Sensing controller however, the input to the fully connected hidden layer comes from external information sensed by the sensor nodes. There are no feedback connections from the output layer thus providing no information on internal states or actions. There is still feedback in this controller however, provided by the recursive connections in the hidden layer. The analogy for the this controller is the exteroceptive sensing system found in mammals that respond to stimuli originating outside the body such as the sense of touch, smell, sight and sound. Figure 4 shows the input connections from the sensors that feed forward to the hidden layer and to both the motor outputs and the IDU output.

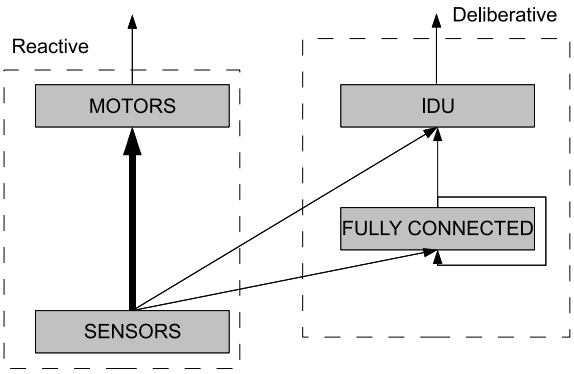


Figure 4: The architecture of the Exo-Sensing controller. The thick arrow indicates the reactive part of the network.

3.3 Composite Controller

The Composite controller as shown in Figure 5 is a hybrid of the Exo-Sensing and the Ego-sensing controllers described previously. Again this controller has two parts, the reactive part which is the same as the other two controllers and the deliberative part. The deliberative part in this controller takes inputs both from the outside world using its IR sensors and from its internal states and actions provided by the feedback inputs from the output layer. In fact a truer description would be that the previous two controllers are a decomposition of the composite controller. This represents a more complete system as found in nature where organisms are furnished with sensory information from the outside world along with information of their own internal state. An example of this would be moving one's hand through space, whilst watching it move and feeling it move at the same time. The structure of the composite controller is shown in Figure 5 with connections to the hidden layer from both the external sensors and from the feedback from the output layer.

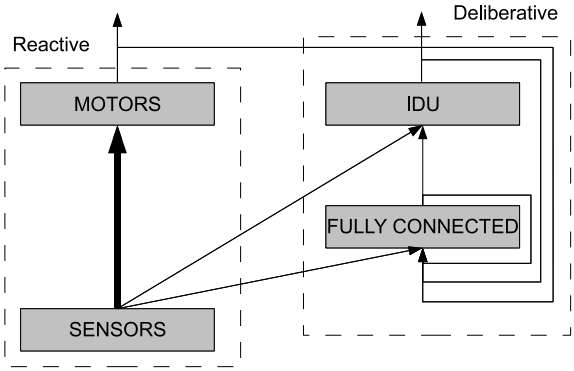


Figure 5: The architecture of the Composite controller. The thick arrow indicates the reactive part of the network.

3.4 Pre-Trained Reactive Controller

For each of the three types or architecture detailed above, there are 2 sets of experiments carried out. In the first set, all of the weights for each of the connections shown in figures 3 – 5 are allowed to adapt freely throughout the adaptation process. In the second set of experiments, the weights of the connections from the sensors to the motors, indicated by the thick arrow in each figure, are not adapted by the PSO algorithm. The weights associated with these connections are fixed and are taken from a pre-trained architecture which was trained only on its ability to follow the target. By comparing the performance of the architectures in each of the two experiments we should be able to highlight the role of sensory-motor coordination in their identification strategy. The pre-trained architecture used is the same as the reactive part of the network in each of the architectures, without the hidden layer and the IDU output. Figure 6 shows the details of the reactive network that was trained only on its ability to follow the target. This can be considered an expansion of the boxes labelled "MOTORS" and "SENSORS" and the connections between them in figures 3 – 5.

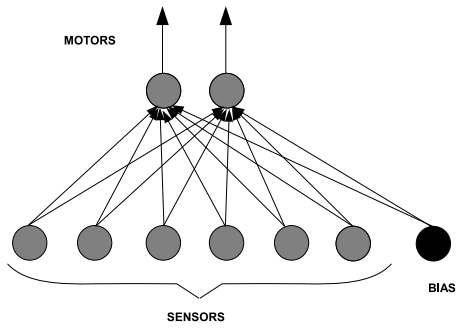


Figure 6: Reactive architecture used to obtain the pre-trained weights for the second set of experiments.

4 PSO FOR ADAPTATION

In this paper, we employ Particle Swarm Optimisation (PSO) to adapt the weights of a robot neural controller. It is a bio-inspired technique that was introduced by Kennedy and Eberhart (Kennedy, 1995) and draws inspiration from the flocking models of birds and fish.

The free parameters, n, of an individual robot controller are represented as the position of a single particle that is flying through an n-dimensional hyperspace. The particle updates its velocity at each iteration based on its own previous best position, *pbest*, and also the previous best position of its neighbours, *nbest*. With time, the particles tend to explore the solution space and, by sharing information on the areas each of them have covered, converge on good solutions.

The methodology for applying PSO to adaptation in robotics is akin to that used in *Evolutionary Robotics* (ER). A similar iterative process is used for PSO in robotics, however, the selection methods and update operators are PSO specific. Figure 7 shows the basic methodology and iterative process of the PSO algorithm.

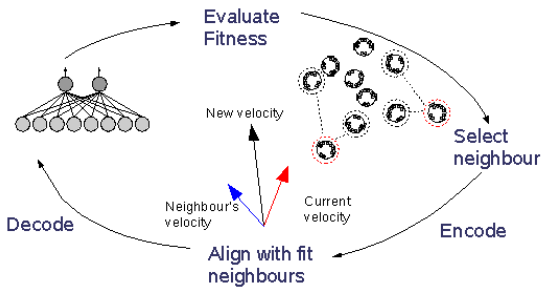


Figure 7: PSO in robotics.

4.1 PSO Parameters

The PSO used in this work was the constriction factor version that was developed by (Clerc, 1999).

$$v_i = K(v_{i(t-1)} + c_1 r_1(p_i - x_i) + c_2 r_2(p_g - x_i))$$
$$K = 2/(|\,2 - \varphi - \sqrt{(\varphi^2 - 4\varphi)}|)\,, \quad \varphi = (c_1 + c_2), \quad \varphi > 4 \qquad (2)$$

Where:

K = constriction factor
v_i = velocity
x_i = position
p_i = own best position
p_g = group best location
c_1 = constant weight of attraction to own best location
c_2 = constant weight of attraction to group best location
r_1 & r_2 = uniform random variables in the range [0,1]

This version of the algorithm has been shown to always converge towards a solution (Clerc, 2002) for a particular range of parameters. The constriction factor version of the algorithm is given in (2).

Some standard parameter settings are used rather than trying to tune the algorithm using empirical methods to be problem specific (Eberhart, 2000). The constriction factor, K, has been set to 0.729 and the cognitive coefficient and the social coefficient (c_1 and c_2) have both been set to 2.05.

The free parameters of the controller that are to be adapted include the dynamic range of the weights, which are randomised in the range [-10, 10] and the memory coefficient, that has been randomised in the range [0, 1]. These two parameters are encoded to represent a particle's position vector. Each particle's velocity vector is also initialised to a random value in the same range as the position vector. The velocity and position vectors are also hard-limited to the range [-10, 10] throughout the adaptation process. A population size of 40 particles has been used for the swarm. This value was achieved through empirical testing and is an acceptable compromise between performance and training time. The neighbourhood topology used for each experiment is the ring topology with a neighbourhood size of 3. Each particle has 2 neighbours and since the neighbourhood size is restricted, the current particle can be its own neighbourhood best particle.

4.2 Fitness Function

Per iteration of the algorithm, each robot is allowed to live for 4 epochs of 2500 time steps of 96ms each. An epoch is ended early if the robot crashes. At the beginning of each epoch, the robot is place close to the target in one of each of the four starting positions; 2 in the part0

The fitness function (4) for the task has two parts. The first part rewards for staying close to the target object as it moves along its trajectory. The definition of "close" here is that the robot must be within sensory range of the target. For each time step that the robot is close to the target *found_target_count* is incremented. Fitness is given as the percentage of the total time (4 epochs x 2500 life steps) that the robot is close to the target. If *found_target_count* is greater than *threshold*, then the second part of the fitness function is evaluated and the first part is ignored.

For the pre-trained Reactive Controller, only the first part of the fitness is evaluated since this network has no identification output. The result is a controller that consistently follows the target well and attains maximum fitness in doing so.

The second part of the fitness function rewards for the robot correctly identifying which part of its trajectory the target is currently in. Figure 6 shows the trajectory of the target and the robot, with its projected IR sensor beams, next to it. The white square shape in the upper part indicates *phase1* of the trajectory and the white rectangle shape in the lower part indicates *phase0*. The grey band between the two phases represents "no man's land" where no reward is given as the robot and target travel between the two phases. The *identified_phase0_count* is incremented for each time step that the robot correctly identifies *phase0* of the trajectory and *identified_phase1_count* is incremented for correctly identifying *phase1*. The second part of the fitness function uses these values represented as percentages of the correct identifications for each phase.

$$fitness = (found\_target\_count / max\_life\_span) \quad (4)$$

if($found\_target\_count > threshold$)

$$fitness = 1 + (perc\_identified\_phase0 * perc\_identified\_phase1)$$

where:

max_life_span
= number of epochs * life span
= 10000

threshold
= *max_life_span* * 0.8
= 8000

5 RESULTS

For each of the neural network architectures and both the fixed and non-fixed weight test, the experiment was run 10 times and the fitness data was recorded and averaged. Figures 8a, 8b and 8c show the plots of the training data. Each plot shows the fixed weight training data and also the same data shifted to the right to the point where the non-fixed weight controller achieves a similar fitness score.

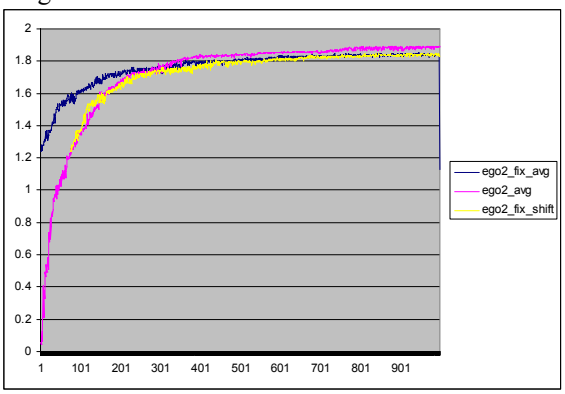


Figure 8a: Training data of the best individual at each iteration for the Ego-sensing controller. Maximum theoretical fitness is 2.0.

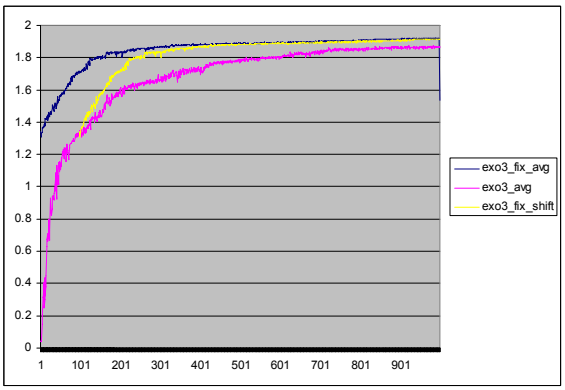


Figure 8b: Training data of the best individual at each iteration for the Exo-sensing controller. Maximum theoretical fitness is 2.0.

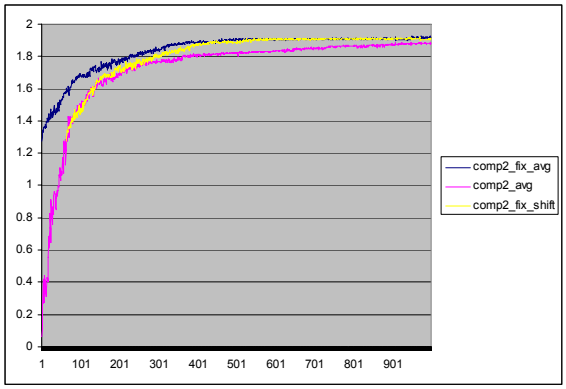


Figure 8c: Training data of the best individual at each iteration for the Composite controller. The plots in figures 8a, 8b and 8c show the data from both the fixed weight and non-fixed weight training. Also the fixed weight data is shown shifted so the fitness scores start at the same point to aid in their comparison. Data is averaged over 10 runs. Maximum theoretical fitness is 2.0.

The reason for this is to make the comparison between the experiments more clear. The controllers in the non-fixed weight experiment had to learn to complete the following part of the task, part 1, before they could gain any fitness from the identification part, part 2. The fixed weight controllers in each case made use of the pre-trained reactive weights and so where already able to complete part 1 of the task from the start of the adaptation process.

The Ego-sensing controller was the only one of the three architectures that performed less well when the motion strategy of the robot was not allowed to adapt along with the identification strategy. For the other two architectures, when the weights controlling robot's motion strategy were fixed, they were both able to train faster on the identification task and achieve a higher maximum score.

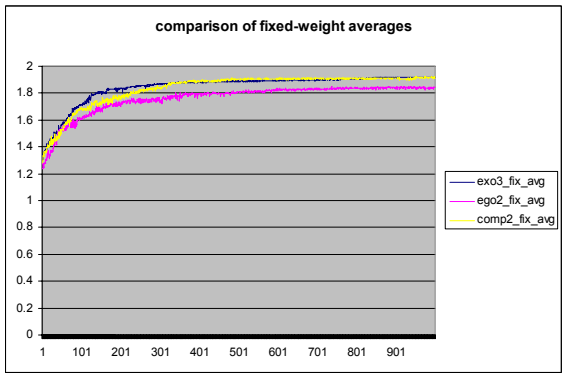


Figure 9: Training data of each of the three architectures for the fixed weight version of the experiment. Data is averaged over 10 runs. Maximum theoretical fitness is 2.0.

When comparing the three architecture types of the fixed weight experiment, it is clear from figure 9 that the Ego-sensing controller also performs the worst with a highest average score of 1.851. The Exo-sensing controller and the Composite controllers perform similarly well scoring highest average scores of 1.921 and 1.925 respectively.

6 DISCUSSION

For the non-fixed weight experiment, each of the controllers was able to quickly adopt a target following behaviour. For each architecture this took between 43 and 44 iterations. This shows that each controller was able to learn a good motion strategy that enabled it to remain close to the target object at all times and that also prevented it from crashing into the target as it suddenly changed direction when turning corners.

In the fixed weight experiment each controller used the same pre-trained weights from the reactive architecture and so all of them were instantly able to follow the target without crashing. All fitness scores above 1.0 are attributed to the robot's ability to identify the phase of the target's trajectory. Each of the fixed weight fitness scores start above 1.0 due to the chance level of identification ability they might have.

When the Ego-sensing controller's performance in the fixed weight experiment is compared to non-fixed weight experiment, the controller performs better when it is allowed to adapt its own motion strategy. The reason for this may be that the information that is fed to the hidden layer in the Ego-sensing controller is somewhat bland (only 2 inputs instead of 6 in the Exo-sensing version). The Ego-sensing controller uses only its own outputs as inputs the hidden layer and these outputs are in fact a function of the sensor inputs. The motion strategy used in the fixed weight experiment was not adapted to aid in the completion of the identification task and was used only as a method to allow the robot to remain close to the target. This fixed motion strategy combined with the fact the that the inputs to the hidden layer were not as rich in patterns as some of the other controllers may have made it harder for the Ego-sensing controller to perform the identification part of the task without being able to adapt its sensory-motor coordination and thus its motion strategy.

The Exo-sensing controller was able to perform better in the fixed weight experiment than the experiment where it was free to adapt its own

motion strategy. In the fixed weight experiment it trained faster and reached a higher maximum average score. Similarly the Composite controller performed better when the weights controlling its motion strategy were fixed. Both of these controllers had inputs to their hidden layer from the six IR sensors. Since this data was the raw instantaneous sensor values, the patterns will have contained much more information than was available to the Ego-sensing controller. The Composite controller also had the ego-sensing data as inputs to its hidden layer and it scored the highest maximum average score of the three architectures.

In previous work by the author (McKibbin *et al*, submitted for review) a study of the nature of the information being fed to the hidden layer revealed that the fast changing sensor (exo-sensing) data can make it more difficult for the controller to learn slower changing temporal patterns. Conversely, the slower changing output (ego-sensing) data seemed to be more useful to the controller to be able to learn the temporal patterns more quickly. However each of the controller types was able to identify the target trajectory with similar success after training for 1000 iterations. The main difference between the architectures was the time taken to train. From the experiments in this work however, it is clear that when the motion strategy is fixed and not adaptable, the controllers perform differently. The controllers that had the fast changing sensor data available to them (Exo-sensning and Composite controllers) were more able to perform the identification task than the controller with only the slower changing output data (Ego-sensing controller). In the latter case, it seems to be that the restriction of the richness of the sensor information available to the controller combined with it not being able to invoke its own sensory-motor coordination strategy has inhibited it.

It should be noted that in the fixed weight experiment, every individual was initialised with the weights that exhibited a pre-trained following behaviour. This meant that every member of the population could begin to optimise the controller for the second part, the identification task. In the non-fixed weight experiment only the individuals who were able to complete the following part of the task could gain fitness in the identification task. It should be noted that even after 1000 iterations only 75% of the members had learnt the following task.

7 CONCLUSIONS

This paper has presented a study of the performance of three recurrent neural robot controllers in identifying environmental motion dynamics. Although all three can perform the task well, we have shown that there are significant differences in performance when sensory-motor coordination is eliminated from their motion strategy. We have highlighted the utility of DNNs as mobile robot controllers and suggest further investigation into the role of sensory-motor coordination in aiding complex robot tasks.

ACKNOWLEDGEMENTS

This work has been supported by the I-SWARM project, European FP6 Integrated Project, Project No. 507006.

REFERENCES

McKibbin, S.P., Amavasai, B., Selvan, A.N., Caparrelli, F., Othman, W.A.F.W, 2007. Recurrent Neural Robot Controllers: Feedback Mechanisms for identifying Environmental Motion Dynamics. Submitted.

Clerc, M., 1999. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the 1999 ICEC,* Washington DC, pp1951-1957.

Clerc, M., Kennedy, J., 2002. The particle swarm - explosion, stability, and convergence in amultidimensional complex space. In *IEEE Transactions on Evolutionary Computation*, vol. 6, issue 1.

Eberhart, R.C. Shi, Y., 2000. Comparing inertia weights and constriction factors in particleswarm optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation*.

Kennedy, J., Eberhart, R.C., 1995. Particle Swarm Optimisation. In *Proceedings of the IEEE Int. Conf. Neural Networks*.

Nolfi, S., 2002a. Evolving robots able to self-localize in the environment: The importance of viewing cognition as the result of processes occurring at different time scales. In *Connection Science, vol. 14 issue 3*.

Nolfi, S., Marocco D., 2002b. Active perception: A sensorimotor account of object categorization. In *From Animals to Animats*. In B. Hallam, D. Floreano, J. Hallam, G. Hayes, J-A. Meyer (eds.) *Proceedings of the VII International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press, pp. 266-271.

O'Regan, J. K., Noe, A., 2001. A sensorimotor account of vision and visual consciousness. In *Behavioral and Brain Sciences,* 24(5):939-73.

BEHAVIOR BASED DEPENDABILITY ESTIMATION
Estimating the Dependability of Autonomous Mobile Systems using Predictive Filter

Jan Rüdiger, Achim Wagner and Essam Badreddin
Automation Laboratory, University of Mannheim, B6, 23-29, Building B, EG, 68131 Mannheim, Germany
ruediger@uni-mannheim.de, a.wagner@ti.uni-mannheim.de, badreddin@ti.uni-mannheim.de

Keywords: Fault-tolerant systems, Autonomous systems, Behavioral systems.

Abstract: Dependability is getting a more important non-functional property of a system. Measuring and predicting the dependability is especially important for autonomous or semi-autonomous and safety-critical systems. Since, at least for (semi-) autonomous systems, those systems are usually described by their behavior, a definition for dependability based on the behavior of the system is evident. In this paper the behavioral based definition of dependability was used together with a particle filter to estimate the dependability of an autonomous mobile system.

1 INTRODUCTION

Non-functional properties reflect the overall quality of a system. Beside performance the dependability is getting a more important non-functional requirement of a system. The general, qualitative, definitions for *dependability* used in the literature so far are (in historical order):

> **Military Standard.** (Department of Defense, 1970) A measure of the degree to which an item is operable and capable of performing its required function at any (random) time during a specified mission profile, given item availability at the start of the mission.
>
> **Carter.** (Carter, 1982) A system is dependable if it is trustworthy enough that reliance can be placed on the service it delivers.
>
> **Laprie.** (Laprie, 1992) Dependability is that property of a computing system which allows reliance to be justifiably placed on the service it delivers.
>
> **Badreddin.** (Badreddin, 1999) Dependability in general is the capability of a system to successfully and safely fulfill its mission.
>
> **Dubrova.** (Dubrova, 2006) Dependability is the ability of a system to deliver its intended level of service to its users.

All definitions have in common that they define dependability on the service a system delivers and the trust that can be placed on that service. The service a system delivers, however, is the behavior as it is perceived by the user, which in our case will be called the mission of the system. They also have in common that they don't define a system independent way of how the measure or evaluate the dependability of a system. Comparing the dependability of different systems, even if a dependability measure for specific systems exists (see (Wilson et al., 2002; Kanoun et al., 2002; Brown et al., 2002; Rus et al., 2002; Cukier and Smidts, 2002; Mukherjee and Siewiorek, 1997; Arlat et al., 1990)), is almost impossible.
According to (Avizienis et al., 2004b; Avizienis et al., 2004a; Randell, 2000) dependability is understood as an integrated concept that further consists of different attributes, threads and means (see Fig. 1). This set of attributes is, however, application specific and thus not fix. In (Candea, 2003) and (Dewsbury et al., 2003) different sets of attributes for evaluating the dependability were proposed. In (Rüdiger et al., 2007b) a reduction of the dependability tree was proposed for the application of autonomous mobile systems. The reduced dependability tree is presented in Fig. 2.

This paper is outlined as follows: In Section 2 a short introduction to the framework of dynamic systems described by their behavior is presented leading to a definion for a system together with a mission, defined in this framework. The section concludes with a definition for a measure for the dependability of this system. Section 2.5 describes different methods of how to apply this definition to actually measure the

dependability. The results of a simulation using particle filter are then presented in Section 4. The paper ends with a discussion of the results in Section 5.

2 BEHAVIOR BASED DEPENDABILITY DEFINITION

2.1 System Definition

In the framework of Willems (see (Willems, 1991)) a system is defined in an universum $\mathbb{U}$. Elements of $\mathbb{U}$ are called outcomes of the system. A mathematical model of a system from a behavioral or black-box point of view claims that certain outcomes are possible, while others are not. The model thus defines a specific subset $\mathfrak{B} \subset \mathbb{U}$. This subset is called the *behavior* of the system.
A (deterministic) mathematical model of a system is then defined as:

Definition 1. *A* mathematical model *is a pair* $(\mathbb{U}, \mathfrak{B})$ *with the universum* $\mathbb{U}$ *- its elements are called* outcomes *- and* $\mathfrak{B}$ *the behavior.*

A dynamical system is a set of trajectories describing the behavior of the system during the time instants of interest in $\mathbb{W}$.
In contrast to the state space representation, like $\dot{x} = f \circ x$, Willems (see (Willems, 1991)) defines a dynamical system as:

Definition 2. *A* dynamical system $\sum$ *is a triple* $\sum = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ *with* $\mathbb{T} \subseteq \mathbb{R}$ *the time axis,* $\mathbb{W}$ *the signal space, and* $\mathfrak{B} \subseteq \mathbb{W}^{\mathbb{T}}$ *the behavior.*

Furthermore an autonomous system is defined as:

Definition 3. *(Autonomous System) Let* $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, $\mathbb{T} = \mathbb{Z}$ *or* $\mathbb{R}$*, be a time-invariant dynamical system.* Σ *is said to be* autonomous *if*

$$\left\{ w_1, w_2 \in \mathfrak{B} \text{ and } w_{1(t)} = w_{2(t)} \text{ for } t < 0 \right\} \Rightarrow \{ w_1 = w_2 \}$$

The definition of an autonomous systems states that the future behavior of the system is completely defined by its past trajectory.
This aspect is an important assumption for modeling the system later.

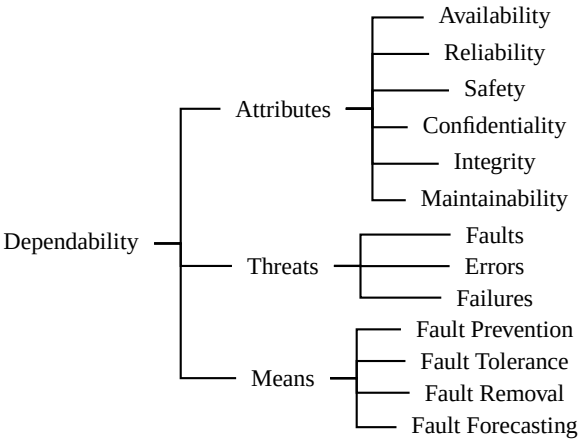


Figure 1: The dependability tree.

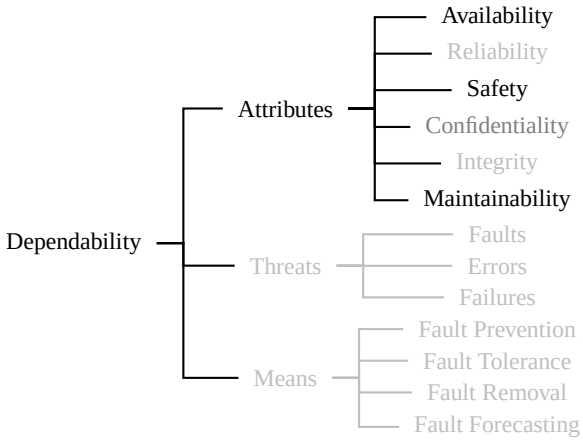


Figure 2: Reduced dependability tree for autonomous mobile systems.

2.2 Behavior and Mission of Autonomous System

To accomplish its task an autonomous system is usuallay given a set of behaviors. In (Rüdiger et al., 2007a) the behavior set of the system was defined as:

Definition 4. *(Behavior) Let* $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ *be a time-invariant dynamical system then* $B \subseteq \mathbb{W}^{\mathbb{T}}$ *is called the set of* basic behaviors $w_i(t) : \mathbb{T} \rightarrow \mathbb{W}$, $i = 1 \ldots n$ *and* $\mathbb{B}$ *the set of fused behaviors.*

Likewise the mission of the system was defined as:

Definition 5. *(Mission) Let* $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ *be a time-invariant dynamical system. We say the* mission w_m *of this system is the map* $w_m : \mathbb{T} \rightarrow \mathbb{W}$ *with* $w_m \in \mathfrak{B}$.

The mission, as defined in (Rüdiger et al., 2007a), is thus just a special trajectory or behavior in $\mathfrak{B}$. Having the system together with a mission mathematically defined is important for a definition of dependability.

2.3 Safe Area $\mathfrak{S}$

Before presenting a definition for dependability, at least the definition for the attribute safety in a behavioral context is needed.

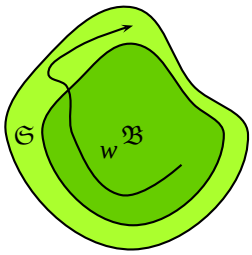


Figure 3: Safety: The system trajectory w leaves the set of admissible trajectories $\mathfrak{B}$ but is still considered to be safe since it remains inside $\mathfrak{S}$

While the other attributes of dependability, like availability, reliability etc., will only be indirectly included in the dependability definition (see Section 2.4 below), the attribute safety, since it is also included in the dependability definitions seen in Section 1, is directly included in the definition (see (Rüdiger et al., 2007a) for a definition of the remaining attributes in a behavioral context).
From a reliability point of view, all failures are equal. In case of safety, those failures are further divided into *fail-safe* and *fail-unsafe* ones. Safety is reliability with respect to failures that may cause catastrophic consequences. Therefore safety is unformaly defined as (see e.g. (Dubrova, 2006)):

> Safety $S(t)$ of a system is the probability that the system will either perform its function correctly or will discontinue its operation in a fail-safe manner.

For the formal definition of safety an area $\mathfrak{S}$ was introduced in (Rüdiger et al., 2007a) and further discussed in (Rüdiger et al., 2007b), which leads to catastrophic consequences when left. This safety area, however, must not be fully contained in the stability region of the system, but $\mathfrak{S}$ is defined to be around $\mathfrak{B}$ ($\mathfrak{B} \subset \mathfrak{S}$). This margin is, like $\mathfrak{B}$, highly system specific, but can be set equal to $\mathfrak{B}$ for a restrictive system.

Definition 6. *Let $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, $\mathbb{T} = \mathbb{Z}$ or $\mathbb{R}$, be a time-invariant dynamical system with a safe area $\mathfrak{S} \supseteq \mathfrak{B}$. The system is said to be* safe *if for all $t \in \mathbb{T}$ the system state $w(t) \in \mathfrak{S}$.*

The definition is illustrated in Fig. 3. Leaving the safe area $\mathfrak{S}$ does not necessary render the system un-operable for the rest of the mission. The above definition of safety permits that the systems trajectory returns to $\mathfrak{B}$ thus making the system fully operable again. This could be achieved be reconfiguration etc.

2.4 Dependability Definition

In (Rüdiger et al., 2007a) the dependability of a system was defined as:

Definition 7. *A time-invariant dynamical system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ with the behaviors $\mathbb{B}$ and a mission $w_m \in \mathfrak{B}$ is said to be (gradually)* dependable *in the period $T \in \mathbb{T}$ if, for all $t \in T$, the mission w_m can be (gradually) accomplished.*

To actually measure the dependability of a given system, this definition needs, however, to be further sophisticated. The main idea behind this definition is to look at the dependability as the difference between the mission trajectory w_m and the system trajectory w, which is the evolution of the system state. This, together with the distance to the safety area $\mathfrak{S}$ will be the main idea of a measure for the dependability.
After the system Σ has completed its mission the dependability $\mathfrak{D}$ of this system with this mission w_m can be defined to as:

$$\mathfrak{D}_{\mathfrak{m}} = 1 - \frac{1}{t^*} \int_0^{t^*} d(\tau) d\tau \tag{1}$$

for the continuous case and for the non-continuous case

$$\mathfrak{D}_{\mathfrak{m}} = 1 - \frac{1}{t^*} \sum_0^{t^*} d(\tau). \tag{2}$$

Where t^* is an appropriate normalizing faktor and d is an appropriate measure of the difference between the mission trajectory w_m and the system trajectory w and as such a combination of different distance measurements. Those distance measurements will be discussed in the following.

More important than knowing the dependability of a system after the completion of the mission is to know the dependability during the mission. For this the equation 1 and 2 is split up into a past and a future part. With this the dependability can be computed to be

$$\mathfrak{D}(t) = 1 - \left(\underbrace{\frac{1}{t^*} \int_0^{t} d(\tau) d\tau}_{\text{Past}} + \underbrace{\frac{1}{t^* + \delta} \int_t^{t+\delta} d(\tau) d\tau}_{\text{Future}} \right) \tag{3}$$

in the continuous case and for the non-continuous case

$$\mathfrak{D}(t) = 1 - \left(\underbrace{\frac{1}{t} \sum_0^{t} d(\tau)}_{\text{Past}} + \underbrace{\frac{1}{t_m - t} \sum_{t+\varepsilon}^{t_m} d(\tau)}_{\text{Future}} \right) \tag{4}$$

Computing the $d_i(t)$ is, of course, system and application specific. For the simulation only the distance between the mission trajectory and the system trajectory ($d_m(t)$) and the relative distance between the system trajectory and the safe area $d_{\mathfrak{S}}(t)$ were used, since these both will be used in most of the dependability measures.
The distance between the mission trajectory and the system trajectory was chosen to be the minimum euclidian distance between system state and the mission trajectory.

$$d_m(t) = 1 - e^{-a*\left(\frac{w(t)-w_m(t)}{w_m(t)}\right)^2} \quad (5)$$

The distance measure for safety $d_{\mathfrak{S}}(t)$ was chosen to be a reliable measure even when the mission trajectory w_m itself is close to the safe area $\mathfrak{S}$. The d_S is defined as follows:

$$d_S(t) = 1 - e^{\left(\frac{\min|\mathfrak{S}-w_m(t)|}{\min|\mathfrak{S}-w(t)|}\right)^2} \quad (6)$$

2.5 Measuring the Dependability

For computing the dependability of a system the actual state of the system and for adequate time horizon the future states must be available with sufficient accuracy. To achieve this different techniques are found throughout the literature, among them:

- Using a model of the system and its environment or
- probabilistic approaches like
  - Kalman Filter or
  - Particle Filter

If for the accomplishment of the mission a set of basic behaviors $\mathbb{B}$ rather than only one behavior is available, the minimum d of those behaviors needs to be taken and the future part of dependability thus computes to:

$$\underbrace{\int_{t+\varepsilon}^{t_m} \frac{\min(d(t))^2}{t_m} dt}_{\text{Future}} \quad (7)$$

If the System is further divided into sub-systems, the different measures of those sub-systems needs also to be joined according to the topology of the system.

3 DEPENDABILITY MONITORING AS RECURSIVE STATE ESTIMATION

The formulation of dependability presented above requires estimating the state of the autonomous mobile system and the environment as it changes over time. This information must then be compared to the mission trajectory w_m to compute the dependability of the system.

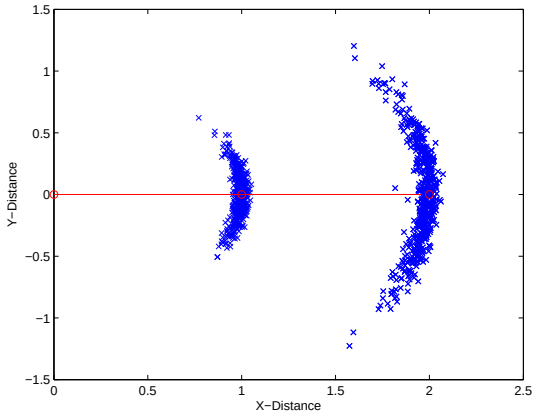


Figure 4: Prediction model of the robot for a translatory movement of 1m and 2m used to predict the dependability of the autonomous mobile system

3.1 Model based State Estimation

Using the mathematical model to compute the dependability of the system is the simplest way. This method, however, can only insufficient deal with changes in the system, which could happen due to system degeneration etc., or changes in the environment. Furthermore mathematical models usually focus on a specific aspect of the system and as thus aren't adequate for computing the dependability. A more sophisticated model of the robot and the environment could compensate this disadvantage with the cost of higher computation time.

3.2 Particle Filter based State Estimation

Since Kalman Filter restricts the state transition and the observation model to be linear functions of the system state, particle Filter are used here to track the state of the autonomous mobile system.
To be able to estimate the dependability with a particle filter, the system is modeled as Markovian, non linear, non-Gaussian. A Sample Importance Resampling Filter (SIR) (see e.g. (Arulampalam et al., 2002),(Chen, 2003)) was then used in a simulation described in the following section to estimate the system state.

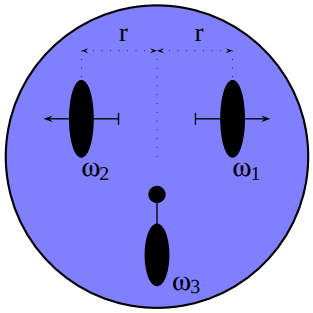


Figure 5: Drawing of the robot used in the simulation. Wheel ω_1 and ω_2 are two independently driven and measured conventional wheels. Wheel ω_3 is an undriven and unmeasured castor wheel.

4 SIMULATION RESULTS

The robot in the simulation has two degree of freedom (DOF) as shown in Fig. 5. For evaluating the dependability of this robot the state (pose)

$$x(t) = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} \tag{8}$$

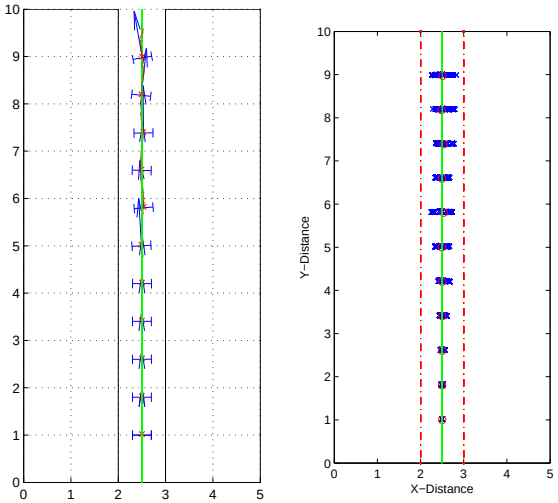
was estimated using a particle filter. The kinematic model of the robot presented in Eq. 9 was used to obtain the prediction model for the movement of the robot.

$$\begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \delta_s cos(\phi_{k-1}) \\ y_{k-1} + \delta_s sin(\phi_{k-1}) \\ \phi_{k-1} + \delta_\phi \end{bmatrix} \tag{9}$$

In this equation δ_s and δ_ϕ where computed using the movement of the wheels ω_1 and ω_2. A Gaussian noise modell is applied separately to each of the two types of motion because they are assumed to be independent. The resulting prediction model can be seen in Fig. 4 for a translatory movement of 1m and 2m.
The observation model used in the simulation is shown in the following equation.

$$y_k = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} \delta_X - r\delta_\phi \\ \delta_Y + r\delta_\phi \end{bmatrix}$$

Where r is the distance between the center of the robot and the contact point of the wheels (see Fig. 5) and δ_X, δ_Y are the motion of the robot in X and Y direction according to the robot coordinate system.
The mission w_m of the system in the simulation (light green line in Fig. 6) was to follow a hallway without colliding with the wall. Noise was added to the wheels to simulate slippage and/or actuator degeneration.



(a) (b)

Figure 6: Simulation Setup. Left image shows the mission trajectory (light green line) of the robot traveling down a hallway (red line). Right image shows the particles used for the state estimation for every 40th time step (blue stars) together with the safe area (dotted red line).

To compute the dependability the distance between the mission trajectory and the robot trajectory ($d_m(t)$) together with the distance between the robot trajectory and the safe area (red line in Fig. 6) relative to the distance between the mission trajectory and the safe area ($d_{\Theta}(t)$) was used to compute the dependability as proposed above. The resulting dependability can be seen in Fig 7. Since a diverge from the mission trajectory also always decreases the distance to the safe area both effects sum up.
In addition to just estimating the system state, the particle filter was also used to predict the future values of the system state and as thus the future dependability of the system. In this setup only the prediction for the next time step was used.

5 CONCLUSIONS

Dependability is of great importance for autonomous mobile systems. Not only for measuring the dependability, but also for comparing it to other missions of the same system or other systems aswell, a formal definition of dependability is important. The definition of dependability used in this paper is based on a mathematical description of the system and its behavior. This property was used in this paper to propose a method for estimating the dependability of an autonomous mobile system using a particle filter.

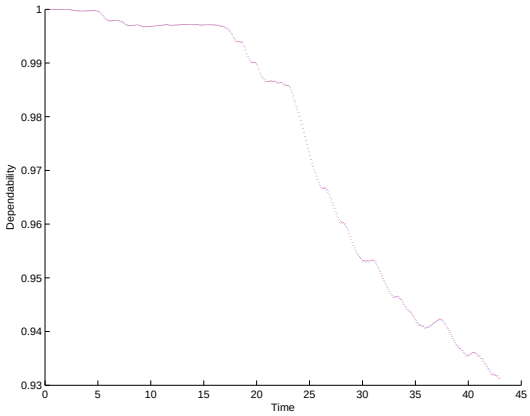


Figure 7: Measured (red) and predicted (blue) dependability of the autonomous mobile system.

REFERENCES

Arlat, J., Aguera, M., Amat, L., Crouzet, Y., Fabre, J.-C., Laprie, J.-C., Martins, E., and Powell, D. (1990). Fault injection for dependability validation: A methodology and some applications. *IEEE Transactions on Software Engineering*, 16(2):166–182.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 50(2):174–188.

Avizienis, A., Laprie, J.-C., and Randell, B. (2004a). Dependability and its threats: A taxonomy.

Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. (2004b). Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. on Dependable and Secure Computing*, 1(1):11–33.

Badreddin, E. (1999). Safety and dependability of mechatronics systems. In *Lecture Notes*. ETH Zürich.

Brown, A., Chung, L., and Patterson, D. A. (2002). Including the human factor in dependability benchmarks. *In Proceedings of the DSN Workshop on Dependability Benchmarking*.

Candea, G. (2003). The basics of dependability.

Carter, W. (1982). A time for reflection. In *Proc. 12th Int. Symp. on Fault Tolerant Computing (FTCS-12)*. FTCS-12) IEEE Computer Society Press Santa Monica.

Chen, Z. (2003). Bayesian filtering: From kalman filters to particle filters, and beyond. Technical report, McMaster University.

Cukier, M. and Smidts, C. S. (2002). Using bayesian theory for estimating dependability benchmark measures. *In Proceedings of the DSN Workshop on Dependability Benchmarking*.

Department of Defense, U. S. o. A. (1970). Military standard - definitions of terms for reliability and maintainability. Technical Report MIL-STD-721C.

Dewsbury, G., Sommerville, I., Clarke, K., and Rouncefield, M. (2003). A dependability model for domestic systems. In *SAFECOMP*, pages 103–115.

Dubrova, E. (2006). Fault tolerant design: An introduction. Draft.

Kanoun, K., Madeira, H., and Aria, J. (2002). A framework for dependability benchmarking. *In Proceedings of the DSN Workshop on Dependability Benchmarking*.

Laprie, J. C. (1992). *Dependability. Basic Concepts and Terminology.* Ed. Springer Verlag.

Mukherjee, A. and Siewiorek, D. P. (1997). Measuring software dependability by robustness benchmarking. *IEEE Trans. Softw. Eng.*, 23(6):366–378.

Randell, B. (2000). Turing Memorial Lecture: Facing up to faults. 43(2):95–106.

Rüdiger, J., Wagner, A., and Badreddin, E. (2007a). Behavior based definition of dependability for autonomous mobile systems. European Control Conference 2007. Kos, Greece.

Rüdiger, J., Wagner, A., and Badreddin, E. (2007b). Behavior based description of dependability - defining a minium set of attributes for a behavioral description of dependability. ICINCO.

Rus, I., Basili, V., Zelkowitz, M., and Boehm, B. (2002). Empirical evaluation of techniques and methods used for achieving and assessing software high dependability. *In Proceedings of the DSN Workshop on Dependability Benchmarking*.

Willems, J. (1991). Paradigms and puzzles in the theory of dynamical systems. *Automatic Control, IEEE Transactions on*, 36(3):259–294.

Wilson, D., Murphy, B., and Spainhower, L. (2002). Progress on defining standardized classes for comparing the dependability of computer systems.

CALCULATING SOFTWARE METRICS FOR LADDER LOGIC

Matthew Waters, Ken Young
International Manufacturing Centre, University of Warwick, Coventry, England, CV47AL, U.K.
Waters.matthew@gmail.com, young_k@wmgmail.wmg.warwick.ac.uk

Ira D. Baxter
Semantic Designs, Austin, Texas, U.S.A.
idbaxter@semanticdesigns.com

Keywords: Metrics, ladder logic, lexical analysis, parsing, attribute evaluation.

Abstract: Ladder logic is a graphical language widely used to program Programmable Logic Controllers (PLCs). PLCs are found at the heart of most industrial control systems used in automation because they are robust, they are relatively easy to program and because they are a proven technology. However there is currently no means to measure the intrinsic properties and qualities of the code produced. This paper details a method for creating tools to calculate software metrics for ladder logic, specifically Rockwell Automation's implementation of ladder logic for its ControlLogix family of PLCs, Import-Export language version 2.6. Results obtained from these tools are briefly discussed also.

1 INTRODUCTION

Ladder logic was originally designed as a method for programming PLCs. It was intended to resemble electrical relay schematic diagrams so that the engineers familiar with the existing hard-wired, relay based electrical control systems could easily adapt to the new technology. It was so successful in this regard that PLC programmers have typically been recruited on the strength of their engineering or technicians background, as opposed to their strength in computer science.

Although the basics of ladder logic have not changed much since that time, the language has evolved to help it meet the increasingly sophisticated needs of automation. Additional functionality has been added to the original relay specification language including: arithmetic operations, timers and counters, data comparison operations, data transfer commands, program control operations, ASCII operations, process control instructions and motion control instructions. Modern ladder logic now has much in common with more conventional programming languages (e.g. C and Java) in both functionality and in the way that the control strategies and algorithms can be implemented. Yet the shortage of software engineers in the field has meant that practices and techniques commonly used in computer science have been neglected in PLC programming languages.

2 SOFTWARE METRICS

The IEEE Standard Glossary of Software Engineering Terminology (IEEE Standard 610.12) defines software engineering as:
"The application of a systematic, disciplined, quantifiable approach to development, operation, and maintenance of software; that is, the application of engineering to software."

The fact that the IEEE considers that engineering software systems requires quantification of system properties is highly relevant. This means that the inherent properties of a given piece of code must be measured in order to be able to 'engineer' it and improve it.

The IEEE also defines a metric as "a quantitative measure of the degree to which a system, component or process possess a given attribute". Software metrics should therefore be an implicit part of the engineering process.

Various software metrics have existed since the 1970s, and they have even been used to assess the complexity of manufacturing control architectures based on software and information flow (Phukan

2005), but very few have been used to evaluate PLC code. There is a desire to see practical tools to achieve this (Frey 2002) and recently a Java program has been built that analyzes the metrics of an Instruction List PLC program (Younis 2007).

Metrics provide a way of analysing the code so the programmer can garner some information about its inherent properties. They are normally computed by static analysis techniques, which means that the program is analysed in an offline mode, as opposed to a dynamic analysis technique that analyses the program as it is running.

Although metrics have not been used to meaningfully analyse PLC code, they have been used extensively on other languages. This means that there is a lot known about the strengths of these tools and the advantages they can offer. These include:

- Wide acceptance of basic value of certain metrics. For example the cyclomatic complexity software metric (McCabe, 1976) is computed using a graph that describes the control flow of the program. The nodes of the graph correspond to the commands of a program. A directed edge connects two nodes if the second command might be executed immediately after the first command. This metric has been used extensively for the last thirty years and it is accepted that the higher the value of complexity, the harder the routine is to understand, test and maintain. Cyclomatic complexity is discussed further in section 3.3.1.2.
- Unbiased assessment of source code quality. Peer review can be used as a method of evaluating software code, but this is a biased assessment that could be affected by how the reviewer feels on the day. A computer program that analyses code will be unbiased.
- Repeatability of measurements. The difficulty in reliance on peer review is consistency in measurement. A programmer reviewing the same code two weeks apart is unlikely to give an identical response. Is not the case with deterministic static analysis.
- Ease of measurement. A metric measurement takes only a short time to collect and can be initiated by the programmer at their convenience; peer review takes far longer and involves more people.
- Ability to judge progress in enhancing quality by comparing before and after assessments.

The use of metrics allows organisation to set thresholds. If these thresholds are exceeded, action is recommended to inspect the code for problems, to reduce the measured values, either through modularisation or some alternate method. The initial coding effort might take longer, but it would in theory make it easier to program by fixing unnecessary complexity. It will also make it easier for a third party to understand the code. This would be a boon to any company, particularly for a modern factory where the demands of flexible automation and agile manufacturing mean that PLC code is changed more frequently than it ever was before.
Another benefit of metrics is that they can help an organisation to identify the software in its portfolio that is of the lowest quality. By being able to tell the difference between what is good code and bad code, steps can be taken to improve the software that is most likely to cause problems in the future.

An advantage of performing software metrics on PLC code is that code designed for similar functions (for example, interfaces for robots) from different manufacturers can be compared to help determine which equipment and software is the easiest to understand and work with. For example if the control interface for one robot manufacturer was found to have significantly less complexity than another company's robot interface, then a purchaser of these robots might be influenced by this information.

3 TOOL BUILDING INFRASTRUCTURE: DMS

Although some metrics are quite simple in theory, extracting them in practice is complex. A lexical analysis approach to industrial control logic analysis has been suggested in the past (Danielsson 2003) but dismissed as being too difficult to implement.

The tool chosen for this task was the "Design Maintenance System" (DMS®) by Semantic Designs.

The DMS Software Reengineering Toolkit is a set of tools for automating customized source program analysis, modification or translation or generation of software systems containing arbitrary mixtures of languages (Baxter 2004). The term "software" for DMS is very broad and covers any formal notation, including programming languages, markup languages, hardware description languages, design notations and data descriptions. It was for this

versatility that DMS was chosen to analyze Rockwell Automation's PLC code.

A very simple model of DMS is that of an extremely generalised compiler, having a parser generator capabilities for arbitrary parseable languages, a semantic analysis framework and a general program transformation engine. It is particularly important that the analyzer output can be used to choose the desired transforms. Unlike a conventional compiler, in which each component is specific to its task of translating one source language to one target machine language, each DMS component is highly parameterized, enabling a wide variety of effects. This means one can choose the input language, the analysis, the transforms, and the output form in arbitrary ways.

The computation of software metrics is based on the structure of the source code. This means metrics can be extracted from a parse of the program text. DMS® has the ability to parse large scale software systems based on the language definition modules used to drive DMS® for software reengineering tasks.

The language definition for PLC control programs is Rockwell Automation's Logix5000 Controllers Import/Export Format Version 2.6. This version was introduced when RSLogix5000 (Rockwell's development environment program) Version 15 was introduced. The language definition module is intended to be backwards compatible. Although many earlier examples of code have been parsed by the module, it has not been extensively tested for every prior version of the import/export language (which will from here-on be referred to as 'L5K', the file-type used by the import/export language.

Some other things to note about this language module are that

- The Motion Instruction set is included owing to the extensive use of these instructions in industry. Consequently, much of the example code used to test the parser made use of these instructions.
- Of the five IEC 61131 languages (ladder logic, sequential function charts, function block diagram, structured text and instruction list), the only one that the parser is presently designed process is ladder logic. Further expansion to extend the functionality to the remaining languages is possible and would be a logical continuation of this work.
- RSLogix5000 version 16 has subsequently been released along with version 2.7 of the import/export format language.

3.1 Lexical Analysis

The job of the lexer is to read in a source l5k program and to 'tokenize' it; that is to convert it from a stream of characters that make up the program body, to a sequence of lexemes. A lexeme is a single atomic unit of the language, for example a keyword. This sequence can then be input to the parser, which in turn will attach structure to the sequence and produce an abstract syntax tree.

3.1.1 Lexical Definition Macros

Macros are definitions of characters, character sets and other useful blocks of text that are made up from regular expressions. Macros may be defined to abstract lexical notions like blank or newline whitespace, case insensitive letters, digits, hexadecimal digits and floating point numbers.

3.1.2 Lexical Modes

DMS® supports the use of lexical modes to lex different source file sections that contain passages of distinct lexical vocabularies. Lexical modes used to lex L5K programs include:

- ModuleDeclarations. Lexes module declarations after module attributes have been collected.
- DataDeclarations. This lexes DATATYPE and TAG block contents.
- RLL. This lexes the PROGRAM section containing the body of ladder logic code.
- ParameterValue. Includes various types of structured values and unstructured strings. This is the lexical mode to collect attribute values including names, numbers etc.

Depending on where they are defined, macros are global (meaning they can be referred to from every lexical mode) or local (meaning that only the lexical mode in which the macro was defined can use that particular macro).

Lexical modes are stored in a stack. The mode that is at the top of the stack is the mode used to process the next token.

Operations performed on the mode stack are nearly always prompted by the occurrence of a specific token in a given lexical mode. Certain tokens encountered in one mode can trigger a change to another mode, reflected in a mode stack modification.

If there is no token found in the current lexical mode then the lexer can perform a last ditch error recovery. This is usually either a switch to another lexical mode, or popping the current lexical mode

from the stack. If the token is then found in a new lexical mode then the lexer carries on from there.

3.2 Parsing

The output from the lexer is a stream of terminal tokens stored within a metafile. This metafile is then used as the input to the parser.

The job of the parser is to attach a hierarchy of significance to list of tokens input to it from the lexer, constructing an abstract syntax representing the source program. The means through which this happens is by applying a set of context-free grammar rules to the token stream.

3.2.1 Context-Free Grammars

Context-free grammar rules are written in a Backus-Naur Form (BNF), which is an established method of describing a formal language. The form for declaring a context-free grammar rule is as follows:

$$NT \rightarrow T$$

where NT is a single non-terminal symbol and T is a sequence that can contain terminals i.e. tokens that have come from the lexer including literals, or non-terminals (which are formed from other grammar rule productions, i.e. are internal to the grammar). T can also be empty. The order of the components in T is critical to the formation of the rule. It is not enough that the appropriate terminals and non-terminals are present - if they are not in the correct order then the grammar rule has not been satisfied.

By substituting one grammar rule into another, each non-terminal can be expressed in terms of a sequence of terminal tokens. In other words, a non-terminal rule is a set of strings that make up part of a legal L5K program.

Multiple rules can be used to associate the same non-terminal symbol for different syntaxes, i.e. one NT may have many different flavours of T. For example, the grammar rule for the production called 'Conditions' is as follows:

```
Conditions = ;
Conditions = Conditions PrimitiveTest ;
```

What this means is that Conditions can be empty, or it can contain an arbitrary number of PrimitiveTests.

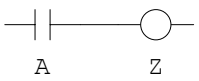


Figure 1: A simple ladder logic rung.

The Goal Non-Terminal. The topmost grammar rule is called the goal non-terminal. This is the first rule specified in the list of grammar rules that make up the formal language. It is unique because every other non-terminal will be used as a component part of another grammar rule, but the goal non-terminal has no 'parent' rule. The set of strings that the goal non-terminal can contain will be every possible legal L5K program, including the stream of tokens generated by lexing a legal program.

3.2.2 Abstract Syntax Trees

The data structure created by the parser that contains all the information about which grammar rules were used to parse a program can be represented diagrammatically by an abstract syntax tree (AST). In an AST, the nodes between branches represent non-terminal rules and the 'leaves' of the tree are terminal tokens. The goal non-terminal is the root node, and the branches show which non-terminal rules can be substituted in to each other.

Figure (2) below shows a small part of an AST, a sub-tree for the simple rung shown in figure (1).

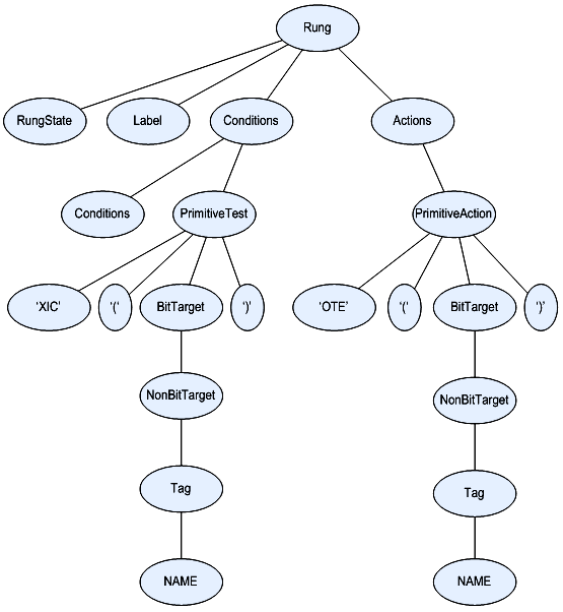


Figure 2: Sub-tree of a simple rung.

In this subtree, the lexed output are the literal instructions 'XIC' and 'OTE', the parentheses '(' and ')' and the terminal token for both instances of NAME, which in this case would be A and Z. Every other node is a non-terminal, and the way in which

these non-terminals are structures represents the form of the grammar rules.

3.3 Attribute Evaluation

Attribute evaluation entails attaching rules to grammar productions and terminals that compute certain interesting values over syntax trees. Computation of these values involves composing the attribute computations for the constituents of the tree with intermediate values passed up or down the tree depending on what is being calculated. Ultimately, calculated values are stored in hash trees associating attributes with specific AST nodes. Passing down from a parent to child node is known as 'inheriting' a value, and from a child to parent is called 'synthesis' of a value. The value passed is often then used in another rule associated with that particular production.

If a parent node needs a value passed up from its child to complete a calculation then it is imperative that the child rule is evaluated before the parent rule. The ordering can be further complicated by directives included by the user which force one rule to be executed before or after another. The partial order for the collection of these values and how they are calculated over a structure as large as an abstract syntax tree is critical.

3.3.1 Calculating Metrics

Attribute evaluation can be used to measure the inherent properties of a piece of code.

The following examples will show how to measure some basic metrics:

- The number of rungs of ladder logic in a program.
- The cyclomatic complexity of a ladder logic program.

Number of Rungs

Every time an AST node reflecting the grammar rule

```
RungList = RungList CommentedRung ;
```

is encountered, the following associated attribute evaluation rule is executed.

```
RungList[0].CommentedRungCount =
RungList[1].CommentedRungCount + 1 ;
```

The CommentRungCount value can then be passed up the AST to a higher level grammar rule by synthesis. The grammar production

```
RoutineDefinition = 'ROUTINE' NAME
RoutineAttributes RungList
'END_ROUTINE' ;
```

Has the associated rule

```
RoutineDefinition[0].CommentedRungCount
= RungList[1].CommentedRungCount ;
```

This hands the value of CommentedRungCount from the child node (RungList) to the parent (RoutineDefinition). Once the value has been passed higher up the tree, the value can be summed over the number of routines, and then over the number of programs to get the final value for the number of rungs in a file.

Cyclomatic Complexity

Calculating the cyclomatic complexity is more difficult. Knowing that

Cyclomatic Complexity = Number of Closed Loops + 1

(Watson, McCabe 1996) and the number of closed loops is essentially the number of 'If' statements makes this possible. But what constitutes an 'If' statement in ladder logic where no such construct is built in to the language?

The assertion made here is that if a non-trivial condition precedes an action in a rung then that is a closed loop. This simplest example of this is shown in figure (1) above. This can be thought of as an If statement in a conventional language; IF `A` is true THEN execute `Z`.

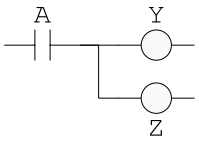


Figure 3.

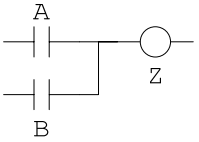


Figure 4.

Figure(3) and figure(4), like the example above still contain just one IF statement. In figure(3) IF `A` is true `THEN` execute `Y AND Z`; in figure(4) IF `A OR B` is true `THEN` execute `Z`.

Of course it is possible to have multiple IF statements contained within the same rung.

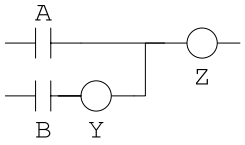


Figure 5.

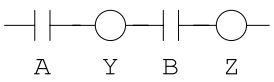


Figure 6.

Both figure(5) and figure(6) contain two `IF` statements. In figure(5), `IF B` true `THEN` execute `Y AND IF A OR B` true `THEN` execute `Z`; in figure(6) `IF A` true `THEN` execute `Y AND IF A AND B` are both true `THEN` execute `Z`.

Determining what is a non-trivial condition or action is achieved by passing around Boolean flags up and down the AST for each rung. These flags assess whether a condition or action is trivial or not. The attribute NonTrivialCondition is defined to be

- A non-empty Condition
- An Action that starts with a non-null condition

```
Conditions = ;
Conditions[0].NonTrivialCondition =
false ;

Conditions = Conditions
PrimitiveTest;Conditions[0].NonTrivialC
ondition = true ;
```

An additional flag is required for Actions (which can contain Conditions in them). NonTrivialTrailingCondition is true if an Action ends with a non-null Condition.

The number of If statements is then summed over all rungs within a routine, and to calculate the cyclomatic complexity one is added to that total.

An example of a grammar rule, the associated attributes and complexity calculation is shown below.

```
PrimitiveAction = '['
ParallelConditions ParallelActions ']';

PrimitiveAction[0].NonTrivialCondition
=
ParallelConditions[1].NonTrivialConditi
on /\
ParallelActions[1].NonTrivialCondition
;

PrimitiveAction[0].NonTrivialTrailingCo
ndition =
ParallelActions[1].NonTrivialTrailingCo
ndition ;

IF
ParallelConditions[1].NonTrivialConditi
on /\
~ParallelActions[1].NonTrivialCondition
THEN PrimitiveAction[0].IfCount =
ParallelActions[1].IfCount + 1 ;
ELSE PrimitiveAction[0].IfCount =
ParallelActions[1].IfCount ;
ENDIF;
```

3.3.2 Reporting Metrics

A family of metrics is calculated using this method. These include:

- Lines code without blank lines and comments
- Number of files *
- Number of Programs *
- Number of Routines *
- Number of Rungs *
- Number of Rungs with comments *
- Cyclomatic complexity
- Cyclomatic complexity of the largest rung *
- Mean Cyclomatic complexity per rung *
- PrimitiveTest Count *
- Maximum number of PrimitiveTests on a rung *
- Mean number PrimitiveTests per rung *
- Decision density
- Ladder instruction occurrence *
- Motion instruction occurrence *
- Halstead unique operators
- Halstead unique operands
- Halstead operator occurrence
- Halstead operand occurrence
- Halstead program length
- Halstead program vocabulary
- Halstead program volume
- Halstead program difficulty
- Halstead program effort
- Halstead bug prediction

These are standard software engineering metrics except the starred metrics which are ladder logic analogues. As well as these metrics, the location in the source code of the rung with the biggest cyclomatic complexity and also the largest number of PrimitiveTests is also reported.

The metrics reporting engine collects the required base information at different points in the hierarchy: routine, program (a collection of routines), controller (a collection of programs) and system level (a collection of files/controllers). The necessary calculations are then performed and the metrics are collated in to multiple reports of different formats (.txt and .xml files).

4 RESULTS

The metrics tool has been used to analyze production code from automotive OEMs. Just under 200000 lines of source code were analyzed – the tool took about 30 seconds to run. The results that it has produced have been very interesting. Some of the most notable are:

- Mean cyclomatic complexity per rung is in the range 1.1 – 1.6. This seems reasonable, averaging just slightly over 1 decision per rung.
- The maximum cyclomatic complexity found for any rung was 31. This level of complexity found on one rung means the program is harder to conceptualize than it needs to be and that to

enhance the readability of the routine for other programmers and users, this complicated rung should be split in to several smaller, simpler rungs.

- The mean number of PrimitiveTests per rung is in the range 2 – 4.
- The maximum number of PrimitiveTests found on a rung is 89. This is a very large value and makes navigating a program difficult as that amount of information cannot easily be fitted on to a standard monitor sized screen even when zoomed out. Action should certainly be taken to address this problem.
- For a set of 729 routines, the mean cyclomatic complexity was 31.37 but the median was 16, indicating that the majority of the routines are relatively small, but some of the bigger ones get quite large. The programmer should re-examine the more complex routines to see if a) there is a better way to implement the logic so that the functionality is equivalent but the routines less complex, b) that the complex routines can be split in to multiple simpler routines

A sample of some metrics report output can be found in the appendix.

5 CONCLUSIONS

There is a need to evaluate the quality of code produced for industrial control systems. Using the DMS® Software Engineering Toolkit, tools have been developed that use attribute evaluation over an abstract syntax tree to compute metrics that have been in common use in more conventional programming languages for nearly thirty years.

It is hoped that appropriate use of these tools will help programmers produce clear and concise code, will allow project managers to make better decisions concerning their code based upon the numerical evidence gleaned from the tool. Industrial partners willing to trial these tools within are being actively sought after.

ACKNOWLEDGEMENTS

My most sincere thanks to Dr. Ira Baxter for his guidance and patient tuition on how to use the DMS® toolkit.

Both the Engineering and Physical Sciences Research Council and Rockwell Automation for funding this research.

Dr. Vivek Hajarnavis and Larry Akers for their valuable encouragement and feedback.

REFERENCES

IEEE Standard 610.12, 1990, *IEEE Standard Glossary of Software Engineering Terminology*

A. Phukan, M. Kalava and V. Prabhu, 2005. *Complexity metrics for manufacturing control architectures based on software and information flow.* Computers & Industrial Engineering 49 1-20

G. Frey, 2002, *Software Quality in Logic Controller Programming,* Proceedings of the IEEE SMC

M. B. Younis and G. Frey, 2007. *Software Quality Measures to Determine the Diagnosability of PLC Applications*. Proceedings of the IEEE ETFA.

Thomas McCabe, 1976, *A Complexity Measure,* IEEE Transactions on Software Engineering, Volume 2, No 4, pp 308-320

F. Danielsson, P. Moore, P Eriksson, 2003, *Validation, off-line programming and optimization of industrial control logic,* Mechatronics 13 571-585

I. D. Baxter, C. Pidgeon, M. Mehlich, 2004, *DMS®: Program Transformations For Practical Scalable Software Evolution,* Proceedings of the IEEE International Conference on Software Engineering

A. Watson, T McCabe, 1996, *Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric,* NIST Special Publication 500-235

APPENDIX

```
FILE
C:/RSLogix5000/l5k_files/Z24_PLC1.L5K
55134 lines of source.
54783 lines of l5k code without blank
lines and comments.
20 programs.
232 routines.
7407 rungs.
Aggregate cyclomatic complexity: 7653
Mean cyclomatic complexity: 32.99
Median cyclomatic complexity: 21.00
Cyclomatic complexity of the largest
rung: 31
Position of rung with maximum
cyclomatic complexity, occurs @ line:
54494
Mean Cyclomatic complexity per rung:
1.03
PrimitiveTest Count: 19025
Maximum number of PrimitiveTests of any
rung in the Controller: 30
Decision density: 5.01
Halstead unique operators: 53
Halstead unique operands: 1865
```

```
Halstead operator occurrence: 37222
Halstead operand occurrence: 108478
Halstead program length: 145700
Halstead program vocabulary: 1918
Halstead program volume: 1588914.89
Halstead program difficulty: 1541.38
Halstead program effort: 2449115919.79
Halstead bug prediction: 529.64

PROGRAM MainProgram @ line 59508
 7027 lines of l5k code without blank
lines and comments.
  139 routines.
  Number of Rungs: 2713
  Aggregate cyclomatic complexity: 4535
  Mean cyclomatic complexity: 32.63
  Median cyclomatic complexity: 4.00
  Mean Cyclomatic complexity per rung:
1.67
  Cyclomatic complexity of the largest
rung: 26
  Position of rung with maximum
cyclomatic complexity, occurs @ line:
63750
  PrimitiveTest Count: 10989
  Mean number PrimitiveTests per rung:
4.05
  Maximum number of PrimitiveTests on a
rung: 81
  Position of rung with maximum number
of PrimitiveTests, occurs @ line: 59702
  Decision density: 0.65
  Halstead unique operators: 67
  Halstead unique operands: 955
  Halstead operator occurrence: 18365
  Halstead operand occurrence: 56269
  Halstead program length: 74634
  Halstead program vocabulary: 1022
  Halstead program volume: 746129.49
  Halstead program difficulty: 1973.83
  Halstead program effort:
1472735785.88
  Halstead bug prediction: 248.71

ROUTINE S_Dcu @ line 20278
   410 lines of l5k code without blank
lines and comments.
   Number of Rungs: 102
   Number of Rungs with comments: 102
   Cyclomatic complexity: 114
   Cyclomatic complexity of the largest
rung: 9
   Position of rung with maximum
cyclomatic complexity, occurs @ line:
20675
   Mean Cyclomatic complexity per rung:
1.12
   PrimitiveTest Count: 167
   Maximum number of PrimitiveTests on
a rung: 11
   Position of rung with maximum number
of PrimitiveTests, occurs @ line: 20675
   Mean number PrimitiveTests per rung:
1.64
   Decision density: 0.28
   Ladder instruction occurrence: 380
   Motion instruction occurrence: 10
   Halstead unique operators: 31
   Halstead unique operands: 183
   Halstead operator occurrence: 442
   Halstead operand occurrence: 1251
   Halstead program length: 1693
   Halstead program vocabulary: 214
   Halstead program volume: 13106.30
   Halstead program difficulty: 105.96
   Halstead program effort: 1388731.04
  Halstead bug prediction: 4.37
```

OPTIMISING A FLYING ROBOT
Controller Optimisation using a Genetic Algorithm on a Real-World Robot

Benjamin N. Passow
Institute of Creative Technologies, De Montfort University, The Gateway, LE1 9BH Leicester, U.K.
benpassow@dmu.ac.uk

Mario Gongora
Centre for Computational Intelligence, De Montfort University, The Gateway, LE1 9BH Leicester, U.K.
mgongora@dmu.ac.uk

Keywords: Genetic algorithm, robot, helicopter, PID, control.

Abstract: This work presents the optimisation of the heading controller of a small flying robot. A genetic algorithm (GA) has been used to tune the proportional, integral, and derivative (PID) parameters of the helicopter's controller. Instead of evaluating each individual's fitness in an artificial simulation, the actual flying robot has been used. The performance of a hand-tuned PID controller is compared to the GA-tuned controller. Tests on the helicopter confirm that the GA's solutions result in a better controller performance. Further more, results suggest that evaluating the GA's individuals on the real flying robot increases the controller's robustness.

1 INTRODUCTION

Flying Robots capable of vertical take off and landing (VTOL) including miniature flying robots (MFR), have gained a strong research interest within the last decade (Bouabdallah et al., 2007). The manoeuvrability of these robots presents new and exciting possibilities for research, industry, military, and search and rescue services.

Helicopters are very versatile in their manoeuvrability and have a number of advantages over other robotic platforms. However, one of the biggest disadvantages is the fact that they are nonlinear and highly unstable systems, very sensitive to external disturbances (Bagnell and Schneider, 2001), and are therefore difficult to control. A simple manoeuvre like hovering requires constant control feedback from the pilot, similar to what a human needs to do when standing and balancing on a ball. Because of the characteristics mentioned above, a controller for an autonomous helicopter must be fast in computing the control response. Active control is traditionally implemented using a combination of proportional, integral, and derivative (PID) control methods (Skogestad and Postlethwaite, 1996). Such a conventional controller is fast and works well for many control applications and therefore has been chosen for this work. Classical PID control techniques have been used before to stabilise an autonomous helicopter (Puntunan and Parnichkun, 2002; Sanchez et al., 2005). The difficulty in applying this method is the right choice of control parameters, and in our case the limited payload of the helicopter for the necessary processing hardware.

Genetic algorithms (GA) have been used before for identification and optimisation of PID control parameters (Perhinschi, 1997). A simulator of the corresponding system is very often used in order to evaluate the individual's fitness within a GA (De Moura Oliveira, 2005). This artificial model requires extensive knowledge about the system being controlled, or the system needs to be identified to create the model. Rather than optimising the controller using a simulation of the system we have explored the possibility of using the robot itself for the optimisation and evaluation of the controller. By doing so, the model identification becomes implicit and the system becomes more accurate, more realistic, thus overcoming the limitation of simulated optimisations where the quality of the solution is restricted by the quality of the simulator.

2 BACKGROUND

The growth of interest and investigations in UAVs includes vehicles capable of vertical take off and landing (VTOL) and miniature flying robots (MFR) (Bouabdallah et al., 2007). Such vehicles are very versatile in their manoeuvrability and thus present an advantageous robotics platform for research in new areas.

Ludington *et al.* use the GTMax helicopter platform for complex navigational tasks, pattern recognition and test runs involving searching for a certain pattern on a building and then identifying windows and doors (Ludington et al., 2006). The achievement of this work shows what can be done using a large autonomous helicopter platform. For a small and light weight indoor helicopter the restrictions demand other approaches.

In general, helicopters have 3 rotational degrees of freedom (DOF), called pitch, roll and yaw, as well as 3 translational DOF called up/down, left/right and forwards/backwards. These outputs are controlled by four inputs, the amount of lift with the speed and/or collective pitch of the rotor, the heading with the anti-torque system or the differential of two rotors, and the pitch and roll rotational angles which are in turn controlled by adjusting the rotor blades' plane angles. For more information the reader is referred to Coyle, 1996.

The helicopter used in this work is a Twister Bell 47 small indoor helicopter model. It is a coaxial rotor helicopter with twin counter rotating rotors with fixed collective pitch and 340 mm. span, driven by two high performance direct current motors and two servos to control rotor blades' plane angles. The weight of the helicopter in its original state is approximately 210 grams and it can lift up to 120 grams.

A helicopter's engine causes a torque effect on the helicopter. The two engines and rotors of this dual coaxial rotor helicopter turn in opposite directions, creating opposite torque effects that cancel each other out. If one rotor's speed is reduced whilst the other's speed is increased by an identical amount, the heading will change whilst the amount of lift is maintained.

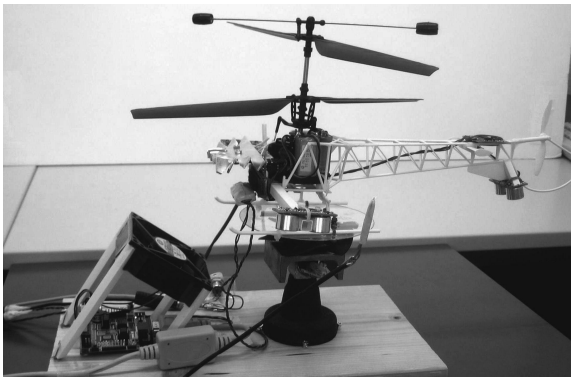


Figure 1: Experimental setup: helicopter strapped to turn table, ready for the GA tuning.

Helicopter controllers have been successfully implemented using a variety of methods, including classical PID control (Puntunan and Parnichkun, 2002). Puntunan and Parnichkun introduce a heading direction and floating height PD-controller for a single rotor helicopter. This confirms that classical control techniques can be used to control helicopter inputs.

The design and evaluation of the controller is often done on a mathematical model and simulation. Shim *et al.* present a study of three control design methodologies for an autonomous helicopter (Shim et al., 1998). The controllers are designed and evaluated on an artificial model based on aerodynamics models.

Sanchez *et al.* (Sanchez et al., 2005) introduce an unmanned helicopter control system combining a Mamdani type fuzzy logic controller (Mamdani, 1974) with PID controllers. The design and evaluation is done using a complex mathematical model. The system is only tested via simulation for hovering and slow velocities, but showed good performance.

Perhinschi (Perhinschi, 1997) used GAs to identify the gain parameters of linear differential equations which are used to stabilise and control a helicopters longitudinal channel. The GAs used a linearised model of a helicopter and the controller performance is not tested in simulation nor on a real helicopter.

3 CONTROL ARCHITECTURE

The model helicopter has four actuators to enable the control of all six DOF. There are two motors which independently control the speed of each rotor, giving combined control over altitude and yaw. Two servos control the lower rotor blades' plane angles for pitch and roll control. For autonomy, all of these actuators need to be controlled by a microprocessor. As only the heading controller is tuned in this work, only that part of the system is described in detail.

A digital compass is used to determine the helicopter's heading. The sensor is attached to the tail of the helicopter to increase the distance to the motors and thus reduce the interference they have on the sensor. The sensor is connected to a microchip PIC microcontroller. The microcontroller handles all onboard computation, sensor inputs, motor outputs, and

serial communication used to transfer information to and from the host computer on which the GA is running.

The control application running on the microcontroller reads all sensors, calculates all four PID control responses, one for each control input, and then sends the overall control responses to the actuators. In this system there are 13 control cycles executed each second.

The PID controller is implemented straightforwardly by first multiplying the proportional part with the proportional gain (PGain). Then the integral part is calculated, multiplying the integral gain (IGain) with the previous error which is limited by an positive and negative limit (IMax and IMin). The derivative part is computed multiplying the change of error with the derivative gain (DGain). Finally all three results are added up to get the final control response.

There are a number of existing techniques for tuning PID controllers (De Moura Oliveira, 2005). Even when tuning a PID controller by hand, a variety of strategies can be applied. The method used in this work is hand tuning and has been adapted from Smith, 1979. Using this method, the parameters shown in table 1 (Value) were identified.

Table 1: Hand tuned parameters and GA's parameter range.

Parameter	Short	Value	Range
Proportional gain	PGain	0.30	0 - 2.00
Integral gain	IGain	0.02	0 - 1.00
Integral state maximum	IMax	100	0 - 400
Integral state minimum	IMin	-100	0 - -400
Derivative gain	DGain	0.70	0 - 4.00

4 EXPERIMENTAL SETUP FOR THE GENETIC ALGORITHM

Genetic algorithms are widely used for search and optimisation purposes (Holland, 1975; Haupt and Haupt, 2004). GAs are very useful for optimising controllers (Fleming and Purshouse, 2002): this robust and flexible method can handle ill-behaved problem domains as well as noise and can be used for multi-objective optimisation.

Rather than using a simulator with the GA, we optimise the controller on the real flying robot. The system setup is shown in figure 1. It shows the dual rotor helicopter attached to a ball bearing supported turn table, restricted to turn up to 90^o and -90^o degrees from its middle position at 0^o. The fan is used for cooling down the helicopter's motors and the embedded system in between tests of individuals. Each test takes about 20 seconds with an additional 20 seconds to cool the system down. The fan is switched off while individuals are evaluated. Each individual is tested by automatically perturbing the helicopter to each side and analysing the controller's reaction. This setup in combination with the GA running on a host computer, enables the automatic implementation and evaluation of individuals and thus the execution of the GA on the real robot without any human intervention. The GA running on the host computer is configured as follows:

Solution Encoding. Based on the hand tuned controllers parameters (table 1), parameter ranges have been chosen and are shown in table 1. The chromosome consists of 5 integer values within that range. The gain parameters are encoded in steps of one hundredth, and the two integral state limits are encoded in 16 steps of 25.

Initial Population. The initial population is created in a random manner, choosing each chromosome randomly within the limits previously defined in table 1. The population size of 20 is chosen small enough to have a fast evaluation of each generation while providing enough individuals to maintain variety.

Evaluation Function. The evaluation function needs to evaluate each individual on the real helicopter. The phenotype of an individual is the real helicopter controller where the controller's parameters are based on the individual's chromosome. The evaluation function is shown in equation 1.

$$e = \sum_{t=1}^{189} (h_t - s)^2 \quad (1)$$

where e is the measure of error, t is time, 189 is the last control cycle of the evaluation, h_t is the heading at time interval t, and s is the setpoint. The fitness of an individual is inversely proportional to the measure of error. Squaring the error on each time interval increases selective pressure on large errors and helps find better solutions more quickly.

The GA program runs on the host computer. To evaluate an individual, its chromosome is sent to the helicopter's embedded system using a direct serial connection. The helicopter uses the received chromosome as the new control parameters, starts the motors and the controller reacts on the heading error based on these new parameters. While the controller and its evaluation are active, the heading sensor data is sent back to the host computer. In order to test the controller's performance on a given error, the helicopter is initially and automatically perturbed by 90^o to the set point, by driving the two rotors with different power levels. The helicopter turns but cannot go beyond 90^o as the experimental setup physically

Table 2: Min, mean, max, and standard dev. of measure of error of 12 tests of hand tuned and best GA individuals.

	Min	Mean	Max	StD
Hand tuned	95080	114351	134705	13556
GA tuned	45584	75850	88306	11233

blocks it. At this point the controller starts acting and is being evaluated. After 92 control cycles the evaluation and controller are paused and the helicopter is perturbed -90^o, into the opposite direction. Finally the controller and its evaluation are started again.

Selection. The selection is based on the fitness combined with random probability, similar to the roulette wheel strategy, but without the possibility of choosing an individual more than once. With this method in place, every individual could be chosen for the next generation although fitter individuals are more likely to be selected.

Genetic Operators. The individuals selected for the next generation are copied or changed using crossover or mutation operators. The best individual is always copied to the next generation. This process is also known as elitism. Altogether 20% of the old population are copied based on the probabilistic selection. These individuals are not changed at all.

Crossover is applied by taking the average of an individual's chromosome and the chromosome of the individual next in the probabilistic selection list. 40% of the next generation are the offspring of the previous generation's individuals.

The mutation operator is the source for new variety. It uses a probabilistic approach whereby the chance of a small mutation taking place is higher than that of big mutation taking place. This method has been applied to 40% of the individuals and on each of its loci.

Termination Criteria. The GA is terminated after the 30th generation. In order to investigate the GA's behaviour no minimum fitness has been defined on which the GA would terminate.

5 RESULTS AND ANALYSIS

The GA on the real helicopter platform has been run without the need for manual interaction with the system. Three complete runs have been conducted and the results are shown and discussed now.

With a population size of 20, running for 30 generations, the GA evaluates 600 individuals from a search space of over 2 billion. Each individual requires about 20 seconds for evaluation and 20 seconds for the helicopter to cool down, thus the overall time for one GA to finish is about 6 hours 40 minutes.

Table 3: Solutions of 3 best individuals of 3 independent GA runs plus measure of error.

PGain	IGain	IMin	IMax	DGain	Error
0.89	0.98	200	0	2.68	45237
0.83	0.43	75	0	3.13	59385
0.86	0.56	75	0	2.97	59886
0.93	0.34	75	0	3.67	52080
1.08	0.15	0	0	3.95	55174
1.05	0.49	25	0	3.73	56036
0.96	0.69	275	0	3.20	48092
0.93	0.00	0	0	3.15	51827
1.12	0.81	325	0	3.07	53210

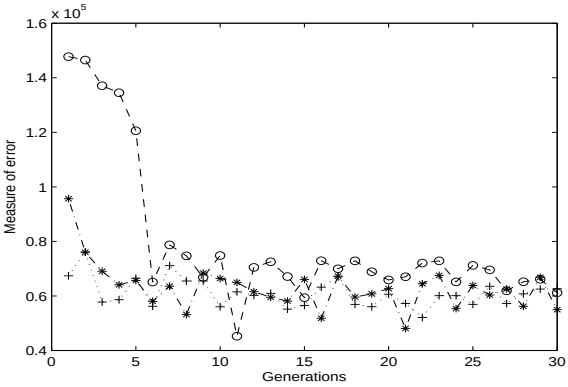


Figure 2: Best individual's measure of error of each generation in 3 independent GA runs.

Elitism is the method of always copying the best individual to the next generation. In simulations this often means a positive or at least zero increase in fitness. Every generation's best individual's measure of error from 3 GA runs is shown in figure 2. Although elitism is used in this work, the fitness of the best individuals of every generation fluctuates and sometimes even gets worse rather than showing a monotonic increase. The cause for this is the deviation when re-evaluating an individual. Retesting two individuals 12 times showed a standard deviation of 11233 and 13556 for the measure of error (table 2). This shows a critical and important difference between using simulated models with an unnatural consistency compared to working with the real system.

Usually, when there is a suitable solution found within a simulated system, such as in generation 11 in figure 2, it is not worth continuing the GA. Using the real world system, continuing the GA helps ensure the consistency of the final solutions. Noise and uncertainties in the system make the GA not converge to one specific solution but to a more robust solution.

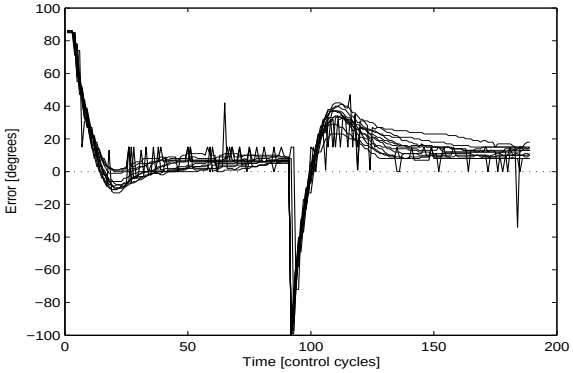


Figure 3: Hand tuned PID controller response to heading perturbed by 90^o at t=0 and -90^o at t=92. Composite of 12 individual tests.

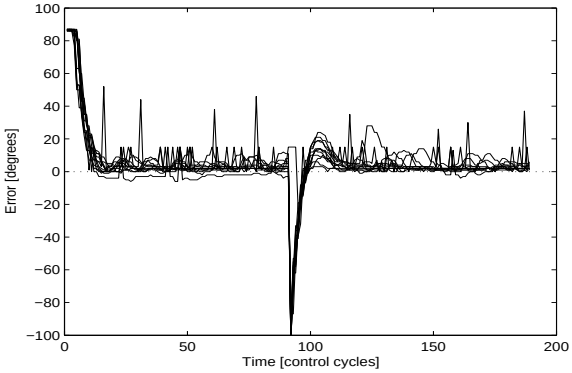


Figure 4: GA tuned PID controller response to heading perturbed by 90^o at t=0 and -90^o at t=92. Composite of 12 individual tests.

Table 3 lists the three best solutions of the three independent GA runs together with their measure of error. The proportional gain value is similar in all three GA runs, converging to a good value region. The small difference between the values indicates that this parameter has a high impact on the controller's performance. The derivative gain parameter did converge to a smaller region of values too, but not as restricted as the proportional gain. On the other hand, the integral gain together with the integral state minimum value seem not to be as important for the fitness as they are not forced into a specific region of values by the selective pressure. In contrast, it is quite obvious that the integral state maximum was forced to be zero. These results show that the system is not symmetrical. They show that the helicopter controller needs more gain in one direction than in the other. This could be caused by a variance in the motors' efficiency factors, helicopter asymmetry, or other causes.

Out of 3 GA runs we chose the one with the least fit final individual. From this GA we used the fittest individual of the last generation to compare it to the hand tuned controller. The individual has a PGain of 0.92, IGain of 0.12, IMin of 250, IMax of 0, and DGain of 3.64. Figure 3 depicts the response graph of the hand tuned PID controller perturbed by 90^o at $t = 0$ and -90^o at $t = 92$ as a composite of 12 individual tests. Figure 4 shows the response graph of the GA tuned PID controller. Figure 5 presents the plot of the mean of these 12 tests, for both the GA and hand tuned controllers.

These graphs confirm that the GA tuned controller performs better than the hand tuned controller. After the first and positive perturbation, in general the hand tuned controller does overshoot the setpoint slightly whereas the GA tuned controller does not. Furthermore, the GA tuned controller reaches the set point quicker and maintains it more accurately. For the

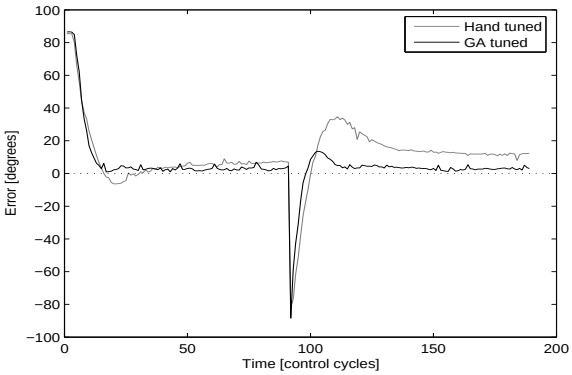


Figure 5: GA (black) and hand tuned (gray) PID controllers response to heading perturbed by 90^o at t=0 and -90^o at t=92. Mean of 12 individual tests for each controller.

second and negative perturbation, in general the GA tuned controller has less overshoot, approaches the set point quicker, and maintains it more accurately.

The measure of error from all 12 tests with the hand tuned and GA tuned controllers has been recorded and the results are presented in table 2. Over all 12 tests, based on the measure of error, the hand tuned controller performed in average 50% worse than the GA tuned controller.

It is quite normal in a real world system that running the same experiment again, evaluating the same individual again, results in a slightly different outcome. The standard deviation for the 12 hand tuned and the 12 GA tuned controller evaluations is 13556 and 11233 respectively (table 2). This variation of results is caused by noise everywhere within the system. Due to the natural uncertainties of real systems, the GA cannot converge to an absolute optimal solution; rather we have observed that it converges to a robust approximate optimum that can cope with uncertainties and noise.

6 CONCLUSIONS

In this work we introduced a controller optimisation methodology based on genetic algorithms running on a real flying robot. A GA has been used to tune the parameters of a PID controller using the real robot rather than a simulator to evaluate the individuals.

When the evaluation of GA individuals is done in a simulator, the performance of the final solution that the GA can find, can only be as good as the accuracy of the model used for the simulation permits. We proposed and presented a GA which evaluated the individuals on a real robot, which made implicit the formal model identification.

Furthermore, we presented the results of three GA runs indicating that its behaviour differs to the behaviour often seen on simulated systems. No monotonic increase in fitness is exhibited by the algorithm, although elitism was used. Additionally, re-evaluating the same GA tuned individual 12 times manifested a deviation, owing to the real system not being as artificially consistent as a simulator, and thus the GA converging to a more robust controller rather than to a particular optimal solution.

In order to confirm the GA's ability to optimise a controller on the complex real flying robot, the best control parameters the GA found have been compared with previously hand tuned control parameters. The performance of 12 individual tests with each controller was compared, and confirmed that the real execution GA found better and more robust solutions.

7 FUTURE WORK

In future work we will use the information from the GA to identify the robot's characteristics, which may be useful to create a very accurate model of it. Based on the individuals' parameters and using the sensor data from the robot we aim to identify the system formally.

With the formal model it may be possible to implement a simulator for the helicopter. At that point a comparison of two GAs can be conducted, one using a simulation and the other using the real robot for the evaluation of individuals.

ACKNOWLEDGEMENTS

We would like to thank Prof. Andrew Hugill, Director of the Institute of Creative Technologies, De Montfort University, for his support of this research project.

REFERENCES

Bagnell, J. and Schneider, J. (2001). Autonomous helicopter control using reinforcement learning policy search methods. volume 2.

Bouabdallah, S., Becker, M., and Siegwart, R. (2007). Autonomous miniature flying robots: coming soon!-research, development, and results. *Robotics & Automation Magazine, IEEE*, 14(3):88–98.

Coyle, S. (1996). *The art and science of flying helicopters*. Arnold, Hodder Headline Group.

De Moura Oliveira, P. (2005). Modern heuristics review for pid control systems optimization: A teaching experiment. In *Proceedings of the 5th International Conference on Control and Automation*, pages 828–833.

Fleming, P. and Purshouse, R. (2002). Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10(11):1223–1241.

Haupt, R. and Haupt, S. (2004). *Practical Genetic Algorithms*. Wiley-Interscience.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press.

Ludington, B., Johnson, E., and Vachtsevanos, G. (2006). Augmenting uav autonomy. *Robotics & Automation Magazine, IEEE*, 13(3):63–71.

Mamdani, E. H. (1974). Application of fuzzy algorithms for simple dynamic plant. *IEE Proceedings on Control Theory and Applications*, 121:1585 – 1588.

Perhinschi, M. (1997). A modified genetic algorithm for the design of autonomous helicopter control system. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pages 1111–1120.

Puntunan, S. and Parnichkun, M. (2002). Control of heading direction and floating height of a flying robot. In *Industrial Technology, 2002. IEEE ICIT'02. 2002 IEEE International Conference on*, volume 2, pages 690–693, Bangkok, Thailand.

Sanchez, E., Becerra, H., and Velez, C. (2005). Combining fuzzy and pid control for an unmanned helicopter. In *Annual Meeting of the North American Fuzzy Information Processing Society*, pages 235–240, Unidad Guadalajara, Mexico.

Shim, H., Koo, T., Hoffmann, F., and Sastry, S. (1998). A comprehensive study of control design for an autonomous helicopter. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 4, pages 3653–3658, Tampa, Florida, USA.

Skogestad, S. and Postlethwaite, I. (1996). *Multivariable Feedback Control: Analysis and Design*. Wiley.

Smith, C. (1979). *Fundamentals of Control Theory*, volume 86. Chemical Engineering.

AN EMBEDDED SYSTEM FOR SMALL-SCALED AUTONOMOUS VEHICLES

David Vissière and Nicolas Petit
Délégation Générale pour l'Armement, France
École Nationale Supérieure des Mines de Paris, France
david.vissiere@dga.defense.gouv.fr, nicolas.petit@cas.ensmp.fr

Keywords: Embedded systems, autonomous vehicles, UAVs.

Abstract: We consider the problem of designing a modular real-time embedded system for control applications with unmanned vehicles. We propose a simple and low-cost solution. Its computational power can be easily improved, depending on application requirements. To illustrate its performance, we report the implementation of a 75 Hz Extended Kalman Filter used for state estimation on a small-scaled helicopter.

1 INTRODUCTION

In this paper, we present some of our research effort in a current program aiming at proposing control strategies and a control architecture for a group of heterogeneous autonomous vehicles.

To conduct this research, we designed a versatile and simple real time embedded system, which can be easily used as real time guidance and navigation system on various platforms. Our work focuses on heterogenous vehicles, including small-scaled (typically less than 2 m wide) Vertical Take-Off and Landing aerial vehicles (VTOL as in (Castillo et al., 2005)) or fixed wing aircraft, and ground vehicles with tank like dynamics (as in (Morin and Samson, 2006; Vissière et al., 2007)). In the future, these vehicles will be asked to act cooperatively on the battlefield as pictured in Figure 1 (see also (Kaminer et al., 2004; Olfati-Saber, 2006) for other scenarios).

In practice, the aerial vehicles represent the most challenging applications in terms of navigation and guidance. The main reason for this, is that these vehicles can not easily go into any safe mode, as opposed to the ground vehicles which are, in comparison, slower and simpler. While it was proven that, with lowered performance expectations, it is possible to stabilize a fixed-wing Unmanned Air Vehicles (UAV) by directly closing the loop with signals from well-chosen sensors (e.g. in (Lee et al., 2003), the authors propose a solution to automatically control a fixed-wing UAV using only a single-antenna GPS receiver), it is considered by the vast majority of the UAV community that navigation systems require data fusion (Cheng et al., 2006). In facts, each sensor technology has its own flaws (among which are drift, noises, and possibly low resolution or low update frequency). Yet, large factors of accuracy can be gained by reconciliating their data.

Example of on-board data fusion applications are ubiquitous among autonomous vehicle control experiments. Reconciliating GPS and Inertial Measurement Unit (IMU) measurements is a classic case-study. In (Xiaokui and Jianping, 2002), results of data fusion from a BeeLine GPS receiver from Novatel^{TM} and a miniaturized IMU are presented. In (Cremean et al., 2005), high-speed data fusion systems have been developed in view of the DARPA Grand Challenge.

In this later experiment, several technological breakthroughs are presented using a high-end and powerful computer architecture. Software components communicate in a machine-independent fashion through a module management system.

Our experiments can not use such a high end setup, because the typical payload of our aerial vehicles does not exceed 5 kg.

Much smaller and lower-weighting systems can be considered though. In (Jung and Tsiotras, 2007), an embedded system is proposed which does not incorporate any powerful calculation board. A simple Rabbit Semiconductor RCM-3400^{TM} microcontroller is used to perform complementary filtering data fusion using a limited computational power. In the same spirit, in (Jung et al., 2005), a low-cost testbed for UAVs is presented. It is reported that the main advantage of designing such an autopilot from

Figure 1: Cooperative autonomous vehicles in a future battlefield.

scratch is that, by contrast to commercially available products (Cloud Cap Technologies, 2004; Micro Pilot, 2004), it provides full access to the internal control structures. We totally agree with this point of view.

In this paper, we present a solution lying in the middle of the two previously mentioned categories. Our system uses two processors. One processor is used to gather data from the sensors and to control the actuators. The other processor is used to perform the data fusion calculations (and possibly the control algorithms). The advantages of this structure are as follows: *i)* task scheduling is easily programmed, because only one of the two processors is in charge of handling the numerous devices and I/O; *ii)* the computations are performed as one single thread on a dedicated board (PC type); *iii)* depending on the computational requirements, the computation board can be easily upgraded without requiring any software changes or rising any concern about task scheduling; *iv*) finally, the overall system is quite low-cost, since it relies on off-the-shelf components and can be easily maintained.

The paper is organized as follows. In Section 2, we present our system architecture. We detail our hardware components and comment on their choices. In Section 3, we present as a test-case the embedding of our system into a small-scaled helicopter. Numerous details of implementation are provided. Finally, we conclude and give directions of future work.

2 SYSTEM ARCHITECTURE

Our primary goal was to develop an embedded system to test algorithms of various complexity on-board various small-scaled platforms. Early in the design process, one first constraint which appeared to us was the payload limitations of the considered flying machines. This lead us to focus on designing a low weight embedded system.

A second issue that was also raised early in the design stage was that the real-time requirements of a control system for such small UAVs are very strong. This is mostly due to the short time horizons instabilities. Yet, in the context of embedded systems, real-time scheduling of a number of sensing and computation tasks is known to be a difficult problem. More precisely, as exposed in (Caccamo et al., 2005), the problem of determining the feasibility of a periodic sequence of prioritized tasks is often (NP)-hard. Sufficient, but not necessary tests are pessimistic. Popular strategies such as the Rate-Monotonic policy (see again (Caccamo et al., 2005)), which consists of putting the highest priority on the shortest task can be proven to be unfeasible is the CPU load is too large[1]. While being troublesome on ground vehicles, such infeasibilities (and the induced inconsistencies in the embedded calculations) would represent a cause of potential major failure for our aerial platforms.

Keeping these two considerations in mind, we decided to develop a robust two-processors embedded system, running two distinct softwares and communicating through a simple two-ways protocol. The system specifications are as follows.

1. It performs the sensing and calculation tasks separately.
2. It is fast enough to run a typical 15 to 30 dimensional states EKF algorithm with a low latency (to eventually produce satisfactory closed-loop results).
3. Itis easy to upgrade.
4. It is versatile enough to handle various type of sensors and communication protocols.

As exposed in Figure 2 and Figure 3, this (modular) embedded system is composed of a micro-controller, which is in charge of gathering information from all the sensors, and a calculation board. These two elements are connected by a serial interface. The micro-controller also has a downlink to a ground station. We now present the details of the hardware components of our system.

2.1 Sensors

Considering both ground and aerial vehicles control applications, we listed a series of useful sensors that needed to be incorporated into our embedded system. Among these are: an IMU, a GPS receiver, a pressure sensor, an anemometer, magnetometers, and various switches. Other possibilities include odometers,

[1]the upper limit on admissible load is 69%, approximately.

LADARs (as used in (Cremean et al., 2005)), and sonars (as used in (Vissière et al., 2007)), or cameras (as used in (Hamel and Mahony, 2007)). In the context of our study, we only considered low-cost sensors. We now detail these. In each case, we specify the weight (in g), the cost (in USD), the dimensions (in mm), the update rate (f in Hz), and the protocol of communication (Comm.).

- **Inertial Measurement Unit (IMU).** Our IMU is a 3DM-GX1 from Microstrain™. It contains three angular rate gyroscopes, three orthogonal single-axis magnetometers, and three single-axis accelerometers, along with 16 bits A/D converters and a micro-controller. This IMU can deliver different messages, ranging from raw-data, to reconciliated measurements. In our setup, we ask the IMU to deliver only calibrated sensors data at a 75Hz rate.

Weight	Cost	Size	f	Comm.
30	1450	39,54,18	75	RS232

- **Global Positioning System (GPS).** Our GPS is a TIM-LS from μblox™. Through a proprietary binary protocol, it provides position and velocity information at a 4Hz rate. Position error is 2.5 m (Circular Error Probability CEP) and velocity error is 2 m CEP.

Weight	Cost	Size	f	Comm.
23	100	32,47,9.5	4	RS232

  The GPS receiver is not very tolerant against power supply voltage ripples. These can be kept below the 50 mV requirements thanks to a dedicated power supply regulator from TRACO™.

- **Barometer.** Our barometer is the MS-5534 from Intersema™. Using a SPI-type protocol, it gives calibrated digital pressure and temperature information. This device requires a 3 V power supply which is obtained through a fast response diode from the main 5 V power supply of the micro-controller.

Weight	Cost	Size	f	Comm.
2	14	5,4,2	20	SPI

- **Anemometer.** A PXLA sensor from ASensTec™ delivers a differential pressure analog signal, which can be read through a 10 bit A/D converter.

Weight	Cost	Size	f	Comm.
10	25	11,8,12	75	Analog

- **Magnetometer.** We use a HMR2300 three axis magnetometer from Honeywell™. Its range is $\pm$ 2 gauss and it has a 70 μgauss resolution.

Weight	Cost	Size	f	Comm.
28	230	75,30,20	154	RS232

- **Take-off and Landing Detector.** Being able to detect take-off and landing instants is necessary to properly initialize data fusion algorithms. In details, detection of the corresponding switches in the dynamics defines when the controls actually have an effect on the system. This is not the case when an UAV is on the ground. This detection is performed with on-off switches which can be located, e.g., on the landing gear. They deliver a logic signal which can be readily interpreted. To prevent electric arcs which might cause trouble to the connected micro-controller, we added a specific interfacing circuit. This switches can also be replaced by active switches which can be used to activate various devices such as digital cameras, or parachutes.

Weight	Cost	Size	f	Comm.
10	6	25,10,5	75	Boolean

Our system is data-driven by the IMU. The main reason behind this choice is that the IMU is considered as a critical sensor.

2.2 MPC555 Micro-controller

The micro-controller which serves as an interface for the sensors and actuators is a MPC555 Power PC from Motorola™. It runs a specific software we developed using the Phytec™ development kit. The reason for choosing this micro-controller are as follows. This device provides a double precision floating-point unit (64 bit) which is convenient for potential embedded algorithms (even if we do not use this possibility here since all computations are performed on the calculation board), it has a relatively fast 40 MHz clock, it has 32 bit architecture and 448Kbyte of Flash memory and 4 MBytes of RAM. Most importantly, among the family of 32 bits kits, the MPC555 has substantial computational capabilities and a large number of versatile and programmable Input/Output ports. In particular, we make an extensive use of its TPUs (Time Process Units), UARTs, A/D converters, SPIs, MPIOs (Modular I/O system) (see (Motorola, 2000)). Finally, it is small (credit card format) and has a low weight.

Weight	Cost	Size	f	Comm.
25	450	72,57,8	all	all

No operating system is used on the micro-controller. Rather the MPC555 runs a specific interrupts-driven software written in C. Information

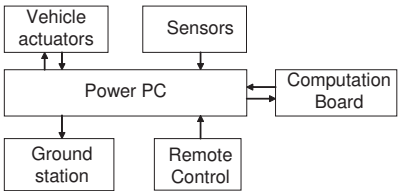


Figure 2: Sensors and computation board connections to the central micro-controller.

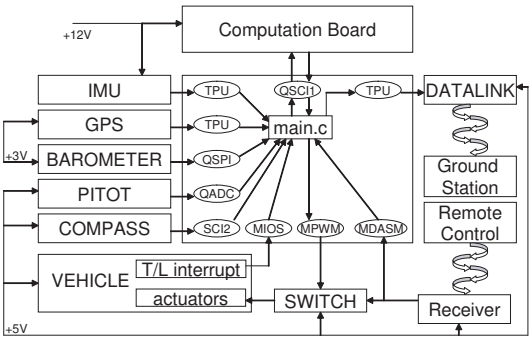


Figure 3: Embedded system internal connections.

from each sensors is transferred using a dedicated interrupt handler routine. Each external source or group of sources has its own interrupt level which avoids potential conflicts. Each data link is associated with a checksum to validate reception.

The data acquisition software running on the micro-controller is event-driven by the IMU messages which periodically sends 31 bytes of data. Once the message of the IMU is received and validated by the micro-controller, others sensors information are either directly read or picked in data buffers which are constantly fed with serial messages from the sensors through hardware interrupts. Information is gathered in a 116 bytes message containing all the onboard measurements. This message is sent to the calculation board through a high-speed serial port. Once the message is received and validated, the calculation board carries out one navigation loop consisting of a prediction equation and an estimation equation of a Kalman filter.

2.3 PC Computation Board

The computing board is a PC running the Knoppix 3.8.1 Linux distribution. The PC board was selected among numerous models (mostly mini-ITX, and PC104) based on computational power, energy consumption, toughness, and price. A fan-less board was considered as the most relevant choice, due to the often observed failures of fans in mechanically disturbed environments.

The chosen fan-less calculation board has a standard mini-ITX PC architecture. Its processor is a 1.2Ghz C7-M from VIA^{TM} designed for embedded applications. It can perform 1500 MIPS and has classic PC Input/Output ports such as a UART serial port (used as main data link with the micro-controller), an ethernet board (not used here), a VGA screen output (which can be used to monitor the system during debugging phases of the software and hardware development), a keyboard, and 4 USB ports (which can be considered for plugging future devices such has controllable cameras).

Experimental preliminary tests have shown us that multi-threading (one thread for message decoding and one thread for the main algorithm) presents two major drawbacks: some data can be lost, and the calculation cycle may end unexpectedly (slightly) late. For this reason, we decided to write our own UART driver using an interrupt handler in the kernel space. Further, we de-activated all hardware interruptions associated to unnecessary devices. As a result, only interrupts from the UART are enabled. Finally, we used a single computation thread.

The operating system is installed on a bootable 1 Gbyte Disk-On-Chip system which prevents all possible mechanical failure associated to hard-drives. This flash memory device is directly connected to the IDE port of the mother-board. The board is powered by a pico-PSU^{TM} power supply which provides various voltages ranging from 5V to 18V. The computation software are written in C and can be either be updated directly on the board via a `ssh` connection, or transferred, in a compiled form, from a remote PC. Custom scripts for compiling and distributing our executable code and configuration files are an efficient way to upgrade the software during development and testing.

Weight	Cost	Size	f	Comm.
800	350	170,180,40	1.5e6	RS232

3 FIELD EXPERIMENT

We now report some field experiment using our embedded system. To perform model identification tasks and design a real-time state observer in view of closed-loop control, we decided to load our embedded system into a small-scaled helicopter. In this section, we expose this work, and give numerous details about solutions to specific interfacing and vibrational issues.

3.1 Experimental Setup

We conducted all the experiments with the help of a human pilot which could, *at any time* remotely connect the inputs of the actuators of the helicopter to the outputs of our embedded system or directly to the outputs of the radio receiver (and thus take direct control of the helicopter). This is achieved thanks to a "remotely controlled switch board" which we designed. This switch is actuated from our Graupner^{TM} MC24 remote controller using one of the 12 available radio channels. This system provides a rapid transition between manual and autonomous flight.

During all flights, the pilot could see images from an on-board camera and read measurements from the embedded system received for the down datalink. A convenient Head-Up display system on the ground station was designed for that purpose. Our data link is a MaxStream^{TM} module operating at 2.4 GHz. It provides a RS232 serial data link at 9600 baud sending the information from the embedded system to the ground station.

Weight	Cost	Size	f	Comm.
250	230	40,68,9	30	RS232

3.2 Small-scaled Helicopter

We use a Benzin acrobatic helicopter from Vario^{TM}, with Graupner^{TM} electronics (C4441 servomotors which have high speed and high torque (8.5 Nm), a 16 channels receiver, and a yaw hobby gyroscope). It is a very reliable helicopter platform (we never observed any mechanical issue over mode than 100 flights). Its payload capacity (4Kg) is large enough for numerous reconnaissance applications. Its rotor is 1.9 m wide.

3.3 Interfacing and Vibrational Issues

Wiring the embedded system to the existing helicopter circuitry was achieved using some specific additional boards and connectors. To measure the pilot's orders in real-time, we used a 6 channels voltage follower circuit. Numerous LEDs were added to check the status of our system.

A central problem observed on-board helicopters is the 25 Hz vibrations induced by the main rotor blades. These vibrations generate a large amount of noise on the inertial sensors. In practice, these noises totally overwhelm the useful signals. Fortunately, it is possible to solve this issue by using well-chosen noise dampers. On our helicopter, we decided that the micro-controller and the sensors would all be located on a board which would be physically connected to the frame of the helicopter through four

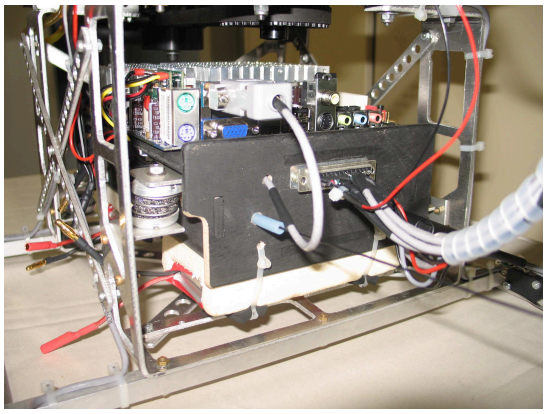


Figure 4: Our embedded system fitted into the (custom-built) landing gear of a small-scaled Vario Benzin helicopter. Springs and dampers are used to filter out vibrations from the main rotor blades.

spring-damper systems (see Figure 4). Experiments conducted on a vibrating table have shown that it was advantageous to keep the embedded system as compact and as rigid as possible. The total weight of the subsystem is about 600 g. We decided to attach some of the batteries to it to bring the weight close to 1.8 kg. This enabled us to use off-the-shelf dampers yielding appropriate cut-off frequencies. MV801-5CC dampers from Paulstra^{TM} were chosen for their ability to work with low masses vibrating at low frequencies. With these, we obtained a satisfactory vibration damping, with a cut-off frequency around 9 Hz. This is represented in Figure 5. Further, resonant frequencies due to the engine frequency (around 160 Hz), the tail rotor frequency (around 115 Hz), and the tail boom were removed using a digital notch filter. The presented solution attenuates high frequency vibration inputs down to negligible levels.

3.4 Experimental Identification

Preliminary model identification experiments need to be conducted before state estimation can be performed.

In particular, using our embedded system, we studied the actuators transfer functions and the yaw rate gyroscope.

In some works (see (Mettler, 2003) or (Mettler et al., 2000)), actuators (servomotors) are considered as first order systems with dead band. We identified such transfer functions for various Graupner^{TM} and Futaba^{TM} servomotors. Results of various tip-in and tip-out in reference signals were recorded to compute the time constant of the first order model.

On board our helicopter, a hobby gyroscope from

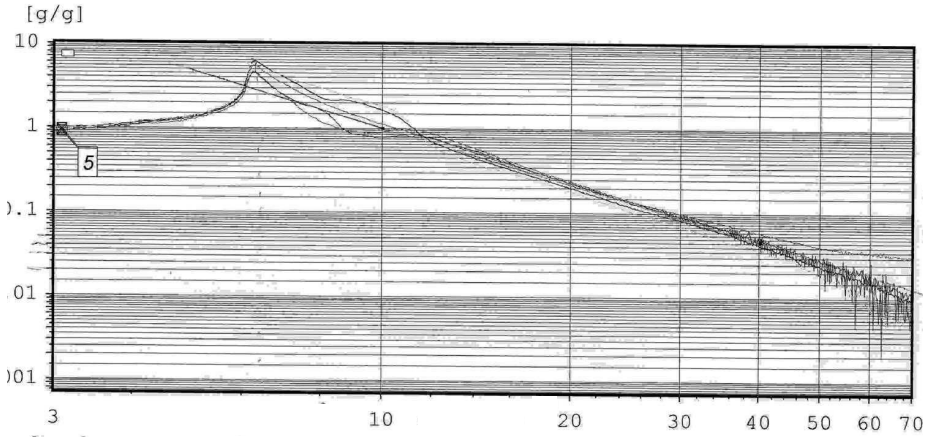


Figure 5: Bode diagram showing the resonance peak and the cut-off frequency of the mechanical structure equiped with the sensors, the micro-controller, and the spring-dampers suspension. The various plots are obtained on varying locations on the vibrating structure and show a good spatial uniformity of the vibration damping.

Graupner^{TM} is used to help the human pilot keeping the yaw rate as small as possible. Pilot orders are transferred from the R/C receiver to the tail actuator through this gyroscope.

To validate simple models of this transfer, we put our IMU under this gyroscope to measure the angular velocity. Simultaneously, we connected the gyroscope and recorded the gyroscope signals sent to the tail actuator. Surprisingly, it was discovered that the gyroscope feedback behaves as a 2 Hz low-pass filter on the pilot orders, and directly transmits the opposite of the received angular rate, without filtering it. This can be summarized under the form

$$\delta_{gyro} = N_r r_m + \frac{\delta_{pilot}}{1+\tau_{gyro}s}$$

3.5 State Estimation

The helicopter is a 6 degrees of freedom mechanical system with high bandwidth dynamic. Reconstructing the full state of this system from low-cost sensors only is quite a challenge. We use an Extended Kalman Filter (EKF, see e.g. (Simon, 2005)) to estimate the state of this system. In practice, the state of our EKF is composed of 23 variables including configuration states (13 variables using quaternions instead of Euler angles), and the aerodynamic and external forces and torques (10 variables including the harmonic expansion of the flapping phenomena as exposed in (Mettler et al., 2001)). We used equally valued tuning parameters for the 3 axis. These are chosen to capture fast dynamics (statistics data are $\sigma_{acceleration} = 8$ m.s^{-2}, and $\sigma_{torque} = 4$ rad.s^{-2}, respectively). Classically, discrete-time updates are implemented. As already discussed, updates are synchronized with the 75Hz measurements from the IMU.

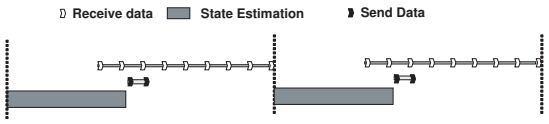


Figure 6: Succession of steps of data transfer and computation.

The succession of tasks performed by the computation board of our embedded system is described in Figure 6. As can be seen, it is necessary, due to a reception time being superior to the available time between two calculation cycles to simultaneously read or send data and perform computations. Some experimental results are presented in Figure 7. Position estimates around hovering are reported. In practice, they appear to be in great accordance with recorded videos.

4 CONCLUSIONS

Designing an embedded system which can qualify as a control system for various small-scaled air and ground vehicles is the subject of the research project presented in this paper. The embedded system we propose here has some interesting features. It is simple, low cost and, most of all, easy to upgrade. Its two processors architecture can incorporate various new

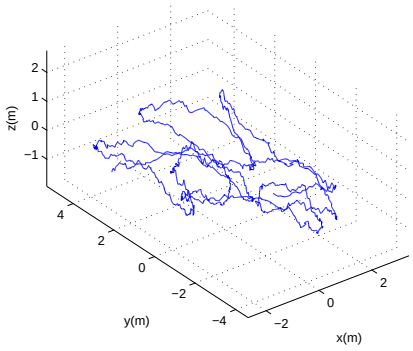


Figure 7: Position estimates during a hovering flight.

processors and sensors.

In our laboratory, this embedded system has been successfully used on a fixed wing aircraft (Rascal 110), a small-scaled helicopter (Vario^{TM} Benzin trainer), and ground vehicles (Pioneer 4 from Mobile Robots^{TM}).

Further developments focus on giving more autonomy, including path planning algorithm, following (Kogan and Murray, 2006; Murray et al., 2003), to respond to high-levels orders from a remote user. We will surely need more computational power, which can be obtained by simply upgrading the computational board. In parallel, we develop a new aerial vehicle for urban area applications. For this last project we consider using other sensors. In particular, for obstacle avoidance and navigation, we will connect LADARS, and ultrasonic sensors to our embedded system.

ACKNOWLEDGEMENTS

The authors are indebted to the numerous students, technicians and engineers who have been collaborating to the development of the presented technology and have brought their support during the conducted experiments.

REFERENCES

Caccamo, M., Baker, T., Burns, A., Buttazzo, G., and Sha, L. (2005). Real-time scheduling for embedded systems. In Hristu-Varsakelis, D. and Levine, W. S., editors, *Handbook of networked and embedded control systems*. Birkhäuser.

Castillo, P., Lozano, R., and Dzul, A. E. (2005). *Modelling and control of mini-flying machines*. Advances in industrial control. Springer.

Cheng, J., Lu, Y., Thomas, E. R., and Farrell, J. A. (2006). Data fusion via Kalman filter: GPS and INS. In Ge, S. S. and Lewis, F. L., editors, *Autonomous mobile robots*, Control engineering series. Taylor and Francis.

Cloud Cap Technologies (2004). http://www.cloudcaptech.com. Hood River, OR, USA.

Cremean, L. B., Foote, T. B., Gillula, J. H., Hines, G. H., Kogan, D., Kriechbaum, K. L., Lamb, J. C., Leibs, J., Lindzey, L., Rasmussen, C. E., Stewart, A. D., Burdick, J. W., and Murray, R. M. (2005). Alice: An information-rich autonomous vehicle for high-speed desert navigation. *Journal of Field Robotics*, (9).

Hamel, T. and Mahony, R. (2007). Image based visual servo-control for a class of aerial robotic systems. 43(11):1975–1983.

Jung, D., Levy, E. J., Zhou, D., Fink, R., Moshe, J., Earl, A., and Tsiotras, P. (2005). Design and development of a low-cost test-bed for undergraduate education in UAVs. In *Proc. of the* 44*th IEEE Conf. on Decision and Control, and the European Control Conference 2005*.

Jung, D. and Tsiotras, P. (2007). Inertial attitude and position reference system development for a small UAV. In *AIAA Infotech at Aerospace*.

Kaminer, I., Yakimenko, O., Dobrokhodov, V., Lizaraga, M., and Pascoal, A. (2004). Cooperative control of small uavs for naval applications. In *Proc. of the* 43*rd IEEE Conf. on Decision and Control*.

Kogan, D. and Murray, R. M. (2006). Optimization-based navigation for the DARPA Grand Challenge. In *Proc. of the* 45*th IEEE Conf. on Decision and Control*.

Lee, S., Lee, T., Park, S., and Kee, C. (2003). Flight test results of UAV automatic control using a single-antenna GPS receiver. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Mettler, B. (2003). *Identification Modeling and Characteristics of Miniature Rotorcraft*. Boston : Kluwer Academic Publishers.

Mettler, B., Tischler, M. B., and Kanade, T. (2000). System identification of a model-scale helicopter. Technical Report CMU-RI-TR-00-03, Carnegie Mellon University.

Mettler, B., Tischler, M. B., and Kanade, T. (2001). System identification modeling of a small-scale unmanned helicopter. *Journal of the American Helicopter Society*.

Micro Pilot (2004). *http://www.micropilot.com*. Stony Mountain, Canada.

Morin, P. and Samson, C. (2006). Trajectory tracking for nonholonomic vehicles. In Kozlowski, K., editor, *Robot Motion Control: Recent Developments*. Springer.

Motorola (2000). MPC555 / MPC556 user's manual. User's manual, Motorola.

Murray, R. M., Hauser, J., Jadbabaie, A., Milam, M. B., Petit, N., Dunbar, W. B., and Franz, R. (2003). Online control customization via optimization-based control. In Samad, T. and Balas, G., editors, *Software-Enabled Control, Information technology for dynamical systems*, pages 149–174. Wiley-Interscience.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Automat. Control*, 51(3):401–420.

Simon, D. (2005). *Optimal state estimation.* Wiley.

Vissière, D., Chang, D. E., and Petit, N. (2007). Experiments of trajectory generation and obstacle avoidance for a UGV. In *Proc. of the 2007 American Control Conference*.

Xiaokui, Y. and Jianping, Y. (2002). Study on low-cost GPS/DMU integrated navigation system. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*.

AN OPTIMIZATION PROCEDURE TO RECONSTRUCT THE AUTOMOBILE INGRESS MOVEMENT

Ait El Menceur M. O., P. Pudlo, F.-X. Lepoutre
LAMIH UMR CNRS 8530, University of Valenciennes, Valenciennes, France
{Mohandouidir.Aitelmenceur, Philippe.Pudlo, Francois-Xavier.Lepoutre}@univ-valenciennes.fr

P. Gorce
HANDIBIO-LESP EA 31-62, University of Sud Toulon-Var, La Garde, France
philippe.gorce@univ-tln.fr

Keywords: Multi objective optimization, Modelling and simulation, Automobile ingress movement reconstruction.

Abstract: To simulate the automobile ingress movement, joint angles are needed. The joint angles are computed from the experimental data issued from an optoelectronic motion capture system. As these systems are often corrupted by problems related either to the system or to the experimentation, the computed angles are biased. Lempereur et al. (2003) proposed an optimization procedure to remedy to this problem. However, their method gives good results only on the end effectors' trajectories, while the other bodies' trajectories are not considered by their method. That degrades the positions of these parts and causes their eventual collisions with the vehicle's parts. On the other hand the corrected angles present some vibrations causing unrealistic simulation. In this paper we present a multi objective optimization based procedure to correct the joint articulation angles in automobile ingress movement. Our method minimizes the distance between all reconstructed trajectories with the real ones at each step of time. Our method follows a compromise between all trajectories of the model. Our method gives better global results. Correction of the joint angles allows a realistic simulation.

1 INTRODUCTION

Automobile accessibility is a serious problem for elderly and/or disabled people that can lead them to stop driving definitely (Cappelaere et al., 1991). To avoid losing these customers, who are in a continuous increasing in the industrialized countries (Brutel, 2002), car manufactures show their interest in this population and in its behaviour during the accessibility movement. The new vehicles tend to be adapted so that to give less discomfort for drivers. However, vehicles can not be modified without considering the driver. Traditionally car manufactures use physical mock ups to test new vehicle prototypes. However, this procedure is very expensive and less reliable (Verriest, 2000). To remedy to its disadvantages, car manufactures had the recourse to the using of numerical simulation (Porter et al., 1993), (Tessier, 2000). HANDIMAN project goes in this direction and aims to integrate the accessibility discomfort evaluation in the first stage of the new vehicle conception for elderly and/or disabled people (Ait El Menceur et al., 2007).

This evaluation is done on the basis of the accessibility movement simulation. Simulation requires modelling of the system to simulate. In our case the system is the human being.

Modelling requires knowledge of the system. The knowledge is acquired from the experimentation on the human being. In HANDIMAN project we used an optoelectronic motion capture system to capture the ingress/egress movements of elderly and/or disabled people (Ait El Menceur et al., 2007). Even though they are among the most reliable movement studying systems, the use of optoelectronic systems encounters some problems (Cappozzo et al., 1996). These problems are due either to the system itself or to the experimental protocol. One of the problems met in the reconstruction, and caused by the experimental protocol, is the computation of joint articulation angles from non rigid bodies (human body) and the integration of these angles in a rigid body structure (humanoid model). The resulted angles are biased and their using in the movement reconstruction induces false trajectories of the humanoid. These

trajectories can provoke the humanoid's collision with the different vehicle's parts (collision of the head with the roof, knee with the steering wheel, penetration of the humanoid inside the seat…).

Lempereur et al. (2003) proposed an optimization based approach to remedy to this problem. This procedure minimizes at each step of time the distance between the measured trajectory (desired trajectory) of the end effector and the trajectory of the humanoid reconstructed from the computation of the joint angles. This procedure gives good results on the end effortor's trajectories. However, as it does not consider other body parts' trajectories (like knees, ankles) these lasts can have erroneous trajectories as they can enter in collision with the vehicle's parts.

In our study we propose an optimization method based on a multi objective function that considers the trajectories of many body parts. Our method aims to minimize at each step of time the distance between the measured trajectories and the trajectories of some body parts (feet, ankles, knees, hips, trunk, neck and head).

Our paper is organized as follows: Section 2 presents the method. Section 3 will detail our optimization procedure. The results will be shown in section 4. The paper is concluded in section 5.

2 METHOD

2.1 Experimentation

The experiments were conducted as part of the French HANDIMAN (RNTS 2004) project. This project aims at integrating the ingress/egress discomfort for elderly and/or disabled persons in first stages of new vehicle conception for these populations. This project considers several trials of ingress and egress movement of 41 test subjects on four vehicles representative of a large part of vehicles present in the trade (Ait El Menceur et al., 2007). In the present study only the ingress trial of one subject on one vehicle is considered, the other trials of other subjects are similar. The trial is performed on a minivan vehicle (see figure 1).

An optoelectronic motion capture system Vicon® 612 at sampling rate of 60 Hz is used. The system is equipped with 8 CCD cameras.

Fifty three anatomical markers are set on the different body segments of the subject to capture the movements during the different acquisitions (Ait El Menceur et al., 2007).

The joint angles are computed for each joint. The ISB recommendation is adopted (Wu et al., 2002).

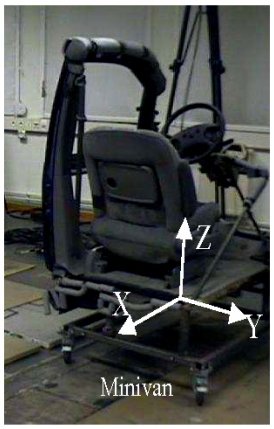


Figure 1: The stripped vehicle used in the study.

2.2 Humanoid Model

To reconstruct the ingress movement, we propose a three dimensional model of 20 DOF considering the two lower limbs and the trunk with the head.

The 20 DOF of the Humanoid model are partitioned as follows: 3 DOF for each hip, 3 DOF for the joint linking the two bodies of the trunk (T10 vertebra according to (Lempereur et al., 2005)), 3 DOF for the joint linking the head to the upper trunk, 2 DOF for each knee and ankle. The humanoid's articulations are rotoid. The convention of Denavit and Hartenberg (Denavit and Hartenberg, 1955) was adopted in the humanoid modeling process. The humanoid model is represented in figure 2.

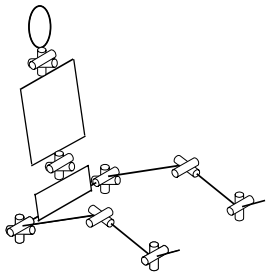


Figure 2: Humanoid model.

3 OPTIMIZATION PROCEDURE

Our method uses a multi objective optimization procedure. It minimizes the sum of the distances between all reconstructed trajectories of different body parts of our humanoid with the measured ones. The mathematical formulation of our optimization problem is given by the following expression.

$$\min \sum_{i=1}^{N} \left\| X_{di} - X_i \right\| = \min \sum_{i=1}^{N} \left\| X_{di} - f_i(q_j) \right\| \quad (1)$$

With N is the number of the body parts whose the positions are to be corrected. $\|.\|$: is the Euclidian distance (norm). q_j with j=1…7 (for the lower limbs chain) and j=1…6 (for the trunk chain) are the calculated joint angles. f_j are parts of the forward geometric function giving the positions of each body part.

The final expression of our optimization procedure is given as follows:

$$\begin{cases} \min \sum_{i=1}^{N} \left\| X_{di} - f_i(q_j) \right\| \\ \text{SC.} \quad q_{i\,min} < q_i < q_{i\,max} \\ \text{SC.} \quad \left| q_i - q_{ical} \right| < \varepsilon_i \\ \text{SC.} \quad \left| q_i - q_{i-1} \right| < V_i \end{cases} \quad (2)$$

The first constraint concerns the physiological angles limitation (Kapandji, 1974).

The second constraint imposes to the optimized angles to not diverge too much from the calculated angles. ε_i are found after several trial and error tests. For the lower limbs chains we kept same values of ε_i proposed by Lempereur et al. (2003). For the trunk chain we identified the corresponding ε_i :

Table 1: Values of ε_i for the trunk chain.

Joints	ε_i
Joint linking the two parts of the trunk	
- Flexion/Extension	19°
- Abduction/Adduction	17°
-Lateral/Medial rotation	13°
Joint linking the head to the upper trunk	
- Flexion/Extension	23°
- Abduction/Adduction	17°
-Lateral/Medial rotation	13°

The third constraint is the constraint of continuity. V_i is the maximum variation allowed between two successive optimized angles.

The maximum variations of the different joints depend on the movement and on the population.

Like in (Lempereur et al., 2003) our optimization procedure is achieved by using the Matlab® Optimization toolbox which uses the sequential quadratic program.

4 RESULTS

We have applied the method of Lempereur et al. (2003) and our method on different kinematic chains of our model. In general the method of Lempereur et al. (2003) gives good results on the end effectors trajectories. However it degrades the positions of other parts. On the other hand, the angles corrected by the method of Lempereur et al. (2003) show many vibrations and these lasts influence negatively on the reconstruction and rendering it “unrealistic”.

For an illustration purpose we present the optimization results of the method of Lempereur et al. (2003) and our method on a bit of the right ankle trajectory. Our method allows good correction of the right ankle trajectory. On the other hand this last is smoothed.

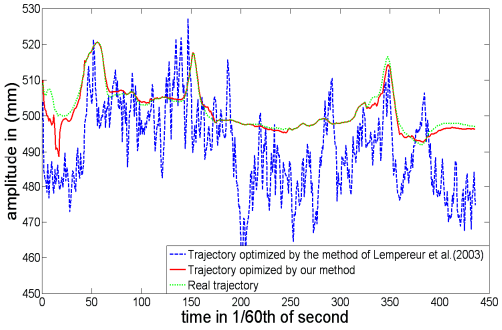


Figure 3: Right ankle different trajectories.

The three dimensional RMS between, respectively, reconstructed trajectories, trajectories optimized by Lempereur et al. (2003) method, our method and the real trajectories are represented in table 2.

Table 2: RMS (in mm) between, respectively, reconstructed trajectories, trajectories optimized by Lempereur et al. (2003) method, our method and the real trajectories.

	Model	Method of Lempereur et al. (2003)	Our method
Right knee	19.0	19.4	18.5
Right ankle	19.3	6.4	3.7
Right foot	20.1	0.0	3.3
Left knee	23.5	32.6	25.4
Left ankle	23.3	13.0	5.6
Left foot	24.4	0.3	5.0
C7	4.3	56.8	34.6
Head	69.6	5.7	16.6

The method of Lempereur et al. (2003) gives good results on the end effectors' trajectories which can even lead to their superposition with the measured trajectories.

However we can see some degradations on other body parts trajectories (like knees and ankles) and that induces to eventual collisions of these parts with the vehicle's parts, like collision of the knee with the steering wheel or penetration of the legs inside the seat.

5 CONCLUSIONS

We have proposed another optimization method based on a multi objective function to correct the joint articulation angles in automobile ingress movement. Our method minimizes at each step of time the distances between all humanoid body parts' trajectories and their real trajectories. We kept same constraints defined by Lempereur et al. (2003) for the lower limbs.

Unlike the method of Lempereur et al. (2003), our method gives good global results on all trajectories. This is due to the fact that it follows a good compromise between all trajectories and at every step of time.

The correction of the joint angles will allow a realistic simulation.

Our method presents slight degradations of some body parts' trajectories. That can be enhanced by integrating some weighting factors on some trajectories.

ACKNOWLEDGEMENTS

This research was supported financially by the European Community, the Délégation Régionale à la Recherche et à la Technologie, the Ministère de l'Education Nationale, de la Recherche et de la Technologie, the Région Nord Pas de Calais and the Centre National de la Recherche Scientifique. The authors gratefully acknowledge the support of these institutions.

The authors would like to thank those at SMPR, INRETS, and Renault who participated in the HANDIMAN project.

REFERENCES

Ait El Menceur, M-O., Pudlo, P., Découfour, N., Bassement, M., Gillet, C., Chateauroux, E., Gorce, P., Lepoutre, F-X., 2007. Towards dynamic studying of the car ingress/egress movement for elderly and disabled population. *in proceeding with IEEE HUMAN'07, Human Machine iNteraction Conference, Timimoun, Algeria, march 12-14th. pp. 350-355.*

Brutel C., 2002. La population de la France métropolitaine en 2050 : un vieillissement inéluctable. *Economie et Statistique*, N°355-356, pp 57-71.

Cappelaere, A., Thévenon, A., Delcambre, B., 1991. Polyarthrite rhumaoïde et conduite automobile. Essais contrôlés d'un véhicule de série. Ann. *Réadaptation Méd. Phys,* 34, pp 239-244.

Cappozzo, A., Catani, F., Leardini, A., Benedetti, M. G., Croce, U. D., 1996. Position and orientation in space of bones during movement : experimental artifacts, *Clinical Biomechanics*, Vol. 11, No. 2, pp. 90-100.

Denavit, J., and Hartenberg, R. S., 1955. A kinematic notation for lower pair mechanism based on matrices, *Journal of Applied Mechanics*, Vol. 22, pp. 215-221.

Kapandji, I. A., 1974. Physiologie articulaire, Fascicule II, Membre inférieur, *Librairie Maloine* S. A., Paris.

Lempereur, M., Pudlo, P., Gorce, P., Lepoutre, F-X., 2003b. Optimization approach for the simulation of car accessibility movement. *Proc. of the IEEE International Conference on Systems Man and Cybernetics, october 5-8, Washington DC, USA*, octobre, ISBN 0-7803-7953-5.

Lempereur, M., Pudlo, P., Gorce, P., Lepoutre, F-X., 2005. Mannequin virtuel adapté à la simulation du mouvement d'entrée-sortie au véhicule automobile. *Journal Européen des Systèmes Automatisés*, 38 (7-8), pp. 959-976.

Porter, J. M., Case, K., Freer, M.T., Bonney, M., 1993.Computer-Aided Ergonomics Design of Automobiles. *In Automotive Ergonomics, Eds. Peacock and Karwonski, Taylor & Francis*, 43- 78. ISBN: 0748400052, 1993.

Tessier, Y., 2000. Vers des mannequins numériques integrés dans la conception de produits . *Les modèles numériques de l'homme pour la conception de produits*, Lyon-Bron.

Verriest, J. P., 2000. Les mannequins numériques dans la conception de produits. *Les modèles numériques de l'homme pour la conception de produits* , Mars 2000, Institut National de Recherche sur les Transports et leur Sécurité.

Wu G., Siegler S., Allard P., Kirtley C., Leardini A., Rosenbaum D., Whittle M., D'Lima D. D.,
Cristofolini L., Witte H., 2002. ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion--part I: ankle, hip, and spine. Journal of Biomechanics, 35(4), 543-54.

A DISTRIBUTED HARDWARE-SOFTWARE ARCHITECTURE FOR CONTROL AN AUTONOMOUS MOBILE ROBOT

Ricardo S. Britto, Andre M. Santana, Anderson A. S. Souza
Adelardo A. D. Medeiros and Pablo J. Alsina
Department of Computating Engineering and Automation, UFRN, Natal, Brazil
{*rbritto, andremacedo, abner, adelardo, pablo*}*@dca.ufrn.br*

Keywords: Control Architecture, Autonomous Mobile Robot, Hybrid Deliberative-Reactive Paradigm, CAN Bus.

Abstract: In this paper, we introduce a hardware-software architecture for controlling the autonomous mobile robot Kapeck. The Kapeck robot is composed of a set of sensors and actuators organized in a CAN bus. Two embedded computers and eigth microcontroller-based boards are used in the system. One of the computers hosts the vision system, due to the significant processing needs of this kind of system. The other computer is used to coordinate and access the CAN bus and to accomplish the other activities of the robot. The microcontroller-based boards are used with the sensors and actuators. The robot has this distributed configuration in order to exhibit a good real-time behavior, where the response time and the temporal predictability of the system is important. We adopted the hybrid deliberative-reactive paradigm in the proposed architecture to conciliate the reactive behavior of the sensors-actuators net and the deliberative activities required to accomplish more complex tasks.

1 INTRODUCTION

The control architecture of an autonomous mobile robot aims qualify the vehicle for operating in its environment using its computational and physical resources. An instance of control architecture must guarantee the accomplishment of its own tasks in a robust-stable way. There are many proposed architectures, but there is not a definitive paradigm that implementates all the functionalities required (Medeiros, 1998).

The more representative paradigms are the hierarchic, reactive and hybrid (deliberative-reactive) paradigms. In the literature, we can find several exemples of classical control architectures: NASREM, Subsumption, AuRa, SFX, Saphira and TCA (Murphy, 2000), PyramidNet (Roisenberg et al., 2004), SHA (Kim et al., 2003), hybrid architecture proposed by Adouane(Adouane and Le Fort-Piat, 2004), CO-HBRA (Heinen, 2002), Ly architecture (Ly et al., 2004) and LAAS architecture (Alami et al., 1993).

A well projected architecture is essential for implementing complex robots. It must have different kinds of hardware and software modules. Moreover, architecture must lead the implementation and the relationship of the modules.

This work firstly specify a hardware-software architecture for the Kapeck robot. In a second moment, this architecture will be used to implement a control organization. Finally, an experiment was made to demonstrate the proposed architecture.

2 HARDWARE ARCHITECTURE

The Kapeck (Figure 1) is a wheeled non-holonomic multi-task robot. The robot has a differential locomotion system. It has 1 stereo head with 2 cameras and 1 sonar belt with 8 sonars. Moreover, Kapeck has 2 manipulators with 5 degrees of liberty, but in the current stage of the project, the manipulators modules are not included in the proposed architecture, but the modular feature of the architecture makes easy the inclusion of these modules in the future.

The differential locomotion system has 2 independent wheels with 1 DC motor each one. In addition to the motor wheels, there are 2 free wheels, without motors. Each motor wheel has 1 attached optical encoder. These optical encoders, as all encoders used in Kapeck, generate 1024 pulses per revolution.

The stereo head has 5 motors with attached optical encoders. There are 2 CCD cameras. These cameras have 1024X768 per pixel max resolution. They cap-

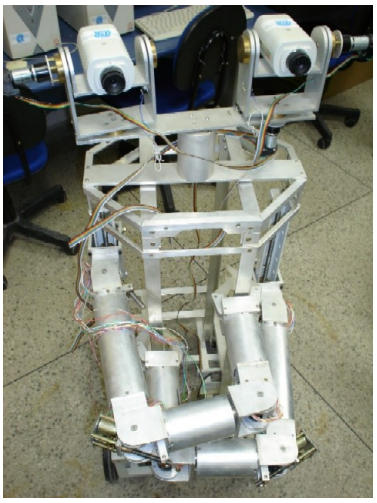


Figure 1: Robot Kapeck.

ture 30 frames per second.

All the sensors and actuators (just not cameras) were attached to a CAN bus (Bosch, 1991). The CAN protocol was developed in the Bosch Laboratories for the automobilist industry, but nowadays this protocol is used in other areas, as robotics. The CAN is very suitable for systems with many sensors and actuators and time constraints. This protocol can perform with multiple masters or in master-slave mode. It supports velocity of 1 Mbit/s in distances up 30 meters. The maximum distance supported by CAN is 5 kilometers. When used in this mode, the maximum velocity reached in a CAN bus is 10 Kbit/s.

Due to the short sampling periods needed to access the sensors and actuators, the presence of real parallelism in the execution of the robot process is needful. Thus, the proposed hardware architecture of Kapeck (Figure 2) was developed to allow distributed processing, using the robot 10 processors. This architecture is modular, which means the inclusion of new hardware modules, like the software ones, is possible and easy.

The hardware architecture is composed of 1 desktop computer, 2 embedded computers and 8 microcontrolled boards. The embedded computers and microcontrolled boards are inside robot. This hardware configuration permits the real parallel execution of the software modules of robot.

The `monitor` computer (Figure 2) stays outside Kapeck and communicates with the robot through a wireless net (IEEE 802.11b). This computer has the user interaction software module. This module watches and interacts with Kapeck.

The `kapeck1` computer (Figure 2) contains the software responsible for the deliberation of the robot.

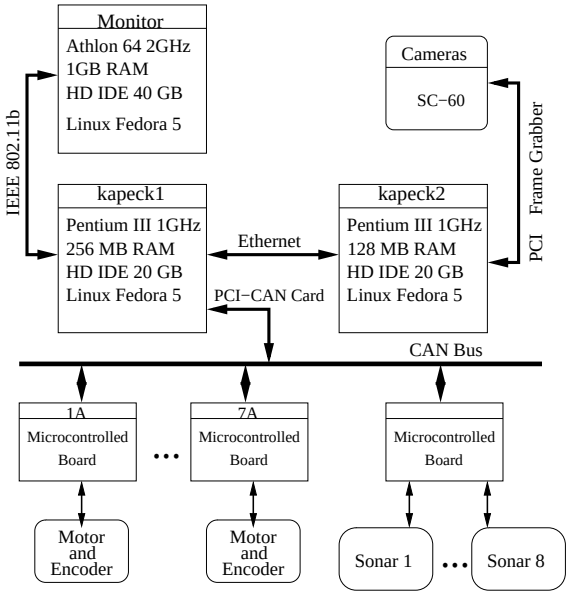


Figure 2: Hardware Architecture.

It is responsible for the interface with the `monitor` computer, passing to user interaction software module information about the robotic system. Moreover, `kapeck1` can execute commands passed by `monitor`.

The `kapeck2` computer (Figure 2) contains the vision system (`visonSis`). The vision system was placed in a separated computer due to its great computational requirement. This computer communicates with the `kapeck1` using a TCP/IP Ethernet peer-to-peer net (crossover cable).

The microcontrolled boards are responsible for the lower control of the sensors and actuators of Kapeck. In this way, there is a performance gain of the robotic system due to the dedicated processing of these boards, releasing the embedded computers to other tasks. These boards are connected to a CAN bus configured in master-slave mode. The CAN bus is connected to the `kapeck1` computer. Only the master of the CAN bus, contained inside `kapeck1`, can permit the access of sensors and actuators to the bus. All the boards have a PIC 18F258. This microprocessor was chosen due to its embedded support to CAN protocol . It also has PWM (Pulse Width Modulation) generator, timers and IO ports, needful to the applications. There are 2 kinds of boards, one called motor controller board, and another called sonar controller board.

The motor controller board is responsible to generate the PWM signal for one motor through the PID (Proportional Integral Derivative) control algorithm, receiving from the high level controller the position references or the velocity references. This board also

has the function of acquiring encoder measurement. When the board is attached to a stereo head motor, the board receives a limit switch signal. The measurements are returned to `kapeck1` computer. The embedded PID control algorithm uses the measurements of encoder and limit switch. The sampling period experimentally chosen to the control cycle of the embedded PID controller is 10 ms.

The sonar controller board is responsible for the sonar functioning. It collects asynchronously sonar measurement when `kapeck1` requires such measurements. The board returns to `kapeck1` the range measurement to the nearest obstacle.

3 SOFTWARE ARCHITECTURE

To take advantage of the distributed hardware of Kapeck, the software modules of robot must be able to execute in parallel way. To make it possible, this work proposes two mechanism that permits parallel execution of software modules. One of them is the blackboard. The blackboards are shared memory area that permits the interaction between software modules (Hayes-Roth, 1985). The other mechanism is a logic representation of the CAN bus called procCAN.

3.1 Blackboards

The blackboards are implemented by 3 `C++` class called `bBoard`, `bBoardAtt`, `bBoardRe` and the software module called `bBoardServer`.

The `bBoard` class is used to create and destroy the blackboards. This class also manages the access to the blackboard created in mutual exclusion, using semaphores. It is used by the system initiator modules, responsible to initiate the other software modules and create all blackboards and semaphores used to exchange information inside the robot. All blackboard, when it is created, receives an unique identification key (ID).

The `bBoardAtt` class is used to attach to existing blackboards. This class provides the read-write primitives of blackboard.

The `bBoardRe` class has the same methods and it functions like the `bBoardAtt`, but it is used to attach to a remote blackboard, created in another Kapeck computer. The utilization of local and remote blackboards is transparent to the client program, because both classes have the same read-write methods.

The `bBoardServer` is a software module implemented in both embedded computers. It uses instances of `bBoardAtt` to accomplish the read-write operations of local blackboards in the name of the remote clients.

An example of architecture that uses blackboards is the COHBRA architecture (Heinen, 2002). COHBRA has a central blackboard where the software modules share information. This approach is different of the one used in this work, where there are many distributed blackboards.

3.2 procCAN

This class makes transparent to the software modules the hardware features of the robot. The existence of this class makes easy hardware alterations of the Kapeck, since it disconnects the software modules implementation from the hardware modules. The software modules operate on the logic CAN bus, that encapsulates the real CAN bus.

The CAN protocol has a native mechanism that controls the simultaneous access to the bus. It uses the CSMA/CA protocol (Carrier Sense - Multiple Access with Collision Avoidance) (Bosch, 1991) to do that. However, Kapeck has a node attached to the CAN bus (`kapeck1` computer) that contains many process competing internally for the access to the CAN bus. In this situation, the existence of a mechanism that guarantee the mutual exclusion access of the bus is needed, since the absence of this kind of mechanism can create message collisions before these messages reach the bus.

The mutual exclusion of the bus is guaranteed by using the CAN bus in master-slave mode and using a bus arbiter. All the methods implemented by the procCAN class internally require the access to the real CAN bus to the bus arbiter. If the bus is busy and a software module requires the permition to access the bus, the requiring module will be blocked until the arbiter concedes the access permition. When the access permition is conceded, the requiring module must execute the desired actions and after it must signal to the bus arbiter that the bus is free to be used to other modules.

An example of architecture that uses the CAN protocol is the architecture present in Coronel work (Coronel et al., 2005). In this work is used high-level communication protocol, on the CAN protocol, called SCoCAN (Shared Channel on CAN) to control bus access.

4 CONTROL ORGANIZATION

The control organization developed for Kapeck (Figure 3) was based in hybrid deliberative-reactive

paradigm. We adopted this paradigm to conciliate the reactive behavior of the sensors-actuators net and the deliberative activities required to accomplish more complex tasks. Thus, the control organization is composed of the following software modules: `Perception`, `Localizer`, `Cartographer`, `actionPl`, `pathPl`, `refRoGen`, `avoiderObs`, `posCon`, `headCon`, `refHeGen` and `visionSis`.

The `Perception` module is responsible for adquires all the sensor measurements. It requires measurements of sonars and encoders, in a fixed sampling period. This module writes in blackboards the received measurements. `Perception` writes in 3 different blackboards: one to write angular displacement of wheels, other to write angular displacement of head parts and a third blackboard to write the sonars measurements.

The `Localizer` module was designed to localize the robot in the current map. It reads data from blackboards where `Perception` writes. This module calculates the robot position using odometry. The calculated robot position is exported to a different blackboard.

The `Cartographer` maps the environment using data exported by `Perception`. The map can be constructed in real-time or, in known environment, be a priori defined. The generated or a priori known map is exported to a blackboard. In the experiment presented in this work, we used a priori known topologic map with metric information, implemented as a graph.

The `actionPl` reads data from blackboards where `Localizer` and `Cartographer` export data. Knowing the global task of the robot, this module determines the action to execute. In the experiment presented in this work, the robot global task is to explore a building with many environments until find a known mark. In this way, there are 4 possible actions:

1. Go to the center of a non-explored environment;
2. Explore a non-explored environment using the stereo head;
3. Go to the mark, if and when it is discovered;
4. Go to another non-explored environment.

The topologic map of building is a priori known. The exploration is done using depth search in the graph that represents the connection topology of the building.

In the case of action 1, an action corresponds to go to the center of environment where the robot is. An action 2 corresponds to execute specifc moviments with the stereo head. These movements are executed when robot reaches the center of the environment. Such actions have as objective to provide to `visionSis` a better pose to capture and process the images.

The action 3 corresponds to go to the mark position, discovered by `visionSis`. This action is triggered only when the robot accomplishes its global task. When this action happens, the building exploration is finalized.

In the case of action 4, an action means go to a non-explored environment. The path from the current environment to the nearest non-explored environment is determined by Dijkstra algorithm (Cormen et al., 2002). All the possible paths are pre-calculated at the initialization of the robot, because the map is a priori known and it has little dimensions (in the case of the experiment of this work). Thus, the `actionPl` is responsible for the execution of the depth search on the map and for the election of the correct action in each moment of robot execution.

The actions are synchronously passed to the `pathPl` module. If the `actionPl` triggers an action 1, the `pathPl` exports the center position of current environment. If the `actionPl` triggers an action 2, the `pathPl` exports the positions that the stereo head parts must reach. If the `actionPl` triggers an action 3, the `pathPl` exports the position of discovered mark. If the `actionPl` triggers an action 4, the `pathPl` generates a geometric path from the current environment to the final desired one, passing through the environments between them. Two blackboards are used by the `pathPl` module: One to export the data needed to move the robot from an environment to another; other blackboard to export data needed to move the stereo head parts.

The refHoGen module reads the blackboard where `pathPl` writes. It generates the reference positions in each sampling period and exports these references in a different blackboard.

The `posCon` is a reactive module that generates velocities percentages for wheel motors using PID algorithm to move the robot to a position read from the blackboard where refHoGen exports data. These percentages are sent to motor controller board, providing the reference to the embedded controller of them. This module also reads the data exported by `Localizer`, needed by the PID controller.

The `avoiderObs` reads sonar measurement acquired by `Perception` module. The `avoiderObs` monitors read data in order to identify obstacles in the robot path. If an obstacle is detected, a safety position is generated and exported to the blackboard where refHoGen exports.

The `headCon` functions like `posCon` module, but it reads references from blackboard where `refHeGen` module writes data.

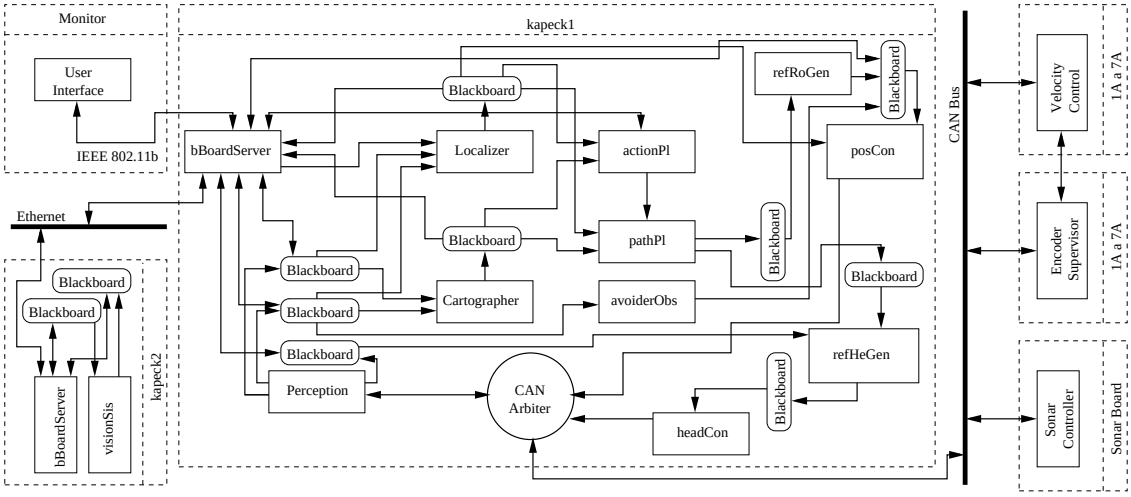


Figure 3: Control Organization.

The `refHeGen` function in to possibles behaviors: Standard behavior and Exploring behavior. In the standard behavior, this module generates and exports to a blackboard specific positions possible to reach by the stereo head parts. In the exploring behavior (started when `actionPl` triggers an action 2 and finished when the action 2 finishes), `refHeGen` starts to read and export the positions generated by `pathPl`.

The `visionSis` processes the camera images to find the desired mark. When the mark is discovered, `visionSis` signals to `actionPl`. After this, `actionPl` triggers an action 3. Together with the signal, `visionSis` exports the robot position to a blackboard.

5 EXPERIMENTAL RESULTS

The experiment developed in this work to demonstrate the proposed architecture consists in a robot that explores a priori known map to find a mark. When the robot finds the mark, it goes to its position. This experiment contains the following software modules of proposed control organization: `Perception`, `Localizer`, `Cartographer`, `actionPl`, `pathPl`, `refRoGen`, `posCon`, `headCon`, `refHeGen` e `visionSis`.

In the Figure 4 we present the experimental results. In the moments T0, T3, T6 and T10 the `actionPl` triggers an action 1. In the moments T1, T4, T7 and T11 the `actionPl` triggers an action 2. In the moments T2, T5, T8 and T12 the `actionPl` triggers an action 4. In the moment T13 `visionSis` identifies the mark, signaling to `actionPl`. So, `actionPl` triggers a action 3. At this moment, the robot discards the rest of path generated in moment T12 and goes to the mark, using the coordinates provided by `visionSis`. In the moment T14 the robot reaches the mark and accomplishes the global task.

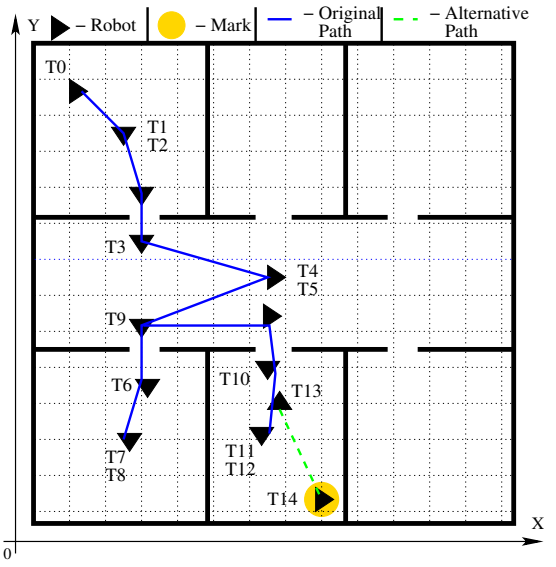


Figure 4: Experimental Results.

6 CONCLUSIONS

This work presented the main ideas of a hardware-software architecture for an autonomous mobile robot. In particular, we propose the using of blackboards to implement the interection between software modules. We also presented the CAN bus access mode, using a logical representation of the real CAN

bus. The proposed architecture tries to manage not only the organization software aspects, but also the hardware organization and the interaction between hardware and software modules. The concepts presented were applied to Kapeck. We pretend in next works improve the software modules of the control organization and develop hardware and software modules to control the manipulators. In the current stage of the work, we use a simple mechanism to coordinate the access to the CAN bus. An evolution of the work could be the utilization of a high-level protocol on CAN.

REFERENCES

Adouane, L. and Le Fort-Piat, N. (2004). Hybrid behavioral control architecture for the cooperation of minimalist mobile robots. In *Proceedings of ICRA'04*, pages 3735–3740, New Orleans,USA.

Alami, R., Chantila, R., and Espiau, B. (1993). Designing an inteligent control archtecture for autonomous mobile robots. In *Proceedings of ICAR'93*, pages 435–440, Tokyo, Japan.

Bosch, R. (1991). Bosch CAN specification. Technical report, Bosch Gmbh.

Cormen, T. H., Leiserson, C. E., Stein, C., and Rivest, R. L. (2002). *Algoritmos: Teoria e Prtica*. Campus, 1 edition.

Coronel, J. O., Benet, G., Sim, J. E., Prez, P., and Albero, M. (2005). CAN-based control architecture using the SCoCAN communication protocol. In *Proceedings of ETFA'05*, Catania, Italy.

Hayes-Roth, B. (1985). A blackboard architecture for control. In *Artificial Inteligence 26*.

Heinen, F. J. (2002). Sistema de controle hbrido para robs mveis autnomos. Master's thesis, Centro de Cincias Exatas e Tecnolgicas, UNISINOS, San Leopoldo, RS.

Kim, J.-O., Im, C.-J., Shin, H.-J., Yi, K. Y., and Lee, H. G. (2003). A new task-based control architecture for personal robots. In *Proceedings of IROS'03*, pages 1481–1486, Las Vegas, USA.

Ly, D. N., Asfour, T., and Dillmann, R. (2004). A modular and embedded control architecture for humanoid robots. In *Proceedings of IROS'04*, pages 2775–2780, Sendai, Japan.

Medeiros, A. A. D. (1998). A survey of control architectures for autonomous mobile robots. *Journal of Brazilian Computer Society*.

Murphy, R. R. (2000). *Introduction to AI Robotics*. MIT Press, Cambridge, Massachusetts 02142, 2 edition.

Roisenberg, M., Barreto, J. M., Silva, F. d. A., Vieira, R. C., and Coelho, D. K. (2004). Pyramidnet: A modular and hierarchical neural network architecture for behavior-based robotics. In *Proceedings of ISRA'04*, pages 32–37, Queretero, Mexico.

ENVIRONMENT FOR DESIGNING AND SIMULATING CONTROL NETWORKS AT DIGITAL HOME

Jorge Azorín-López, Rafael J. Valdivieso-Sarabia, Andrés Fuster-Guilló
and Juan M. García-Chamizo
Dpto. Tecnología Informática y Computación, University of Alicante, Ctra. San Vicente del Raspeig s/n, Alicante, Spain
jazorin@dtic.ua.es, rvaldivieso@dtic.ua.es, fuster@dtic.ua.es, juanma@dtic.ua.es

Keywords: Simulation, control network design, automation, digital home.

Abstract: A design and simulation environment for control network is presented. Control network design could be complex task because many technologies are involved. Each network technology uses its own design and configuration software. Also, it is necessary realize network installation in order to validate its correct operation. This situation introduces high temporal and economical costs in the network life cycle. Simulation design methodology as a task allows detect errors prematurely. System validations are made to a high level of abstraction. This paper proposes a design and simulation environment of digital home control network. It is based on a design independent from technology and postpone the technology choice and incorporates a simulation task that allows simulate the network.

1 INTRODUCTION

In this paper a digital home control networks design environment is proposed. The set of technologies that make the applications viable are diverse: X10 (Fuster and Azorín, 2005), KNX/EIB (Haenselmann et al., 2007), LonWorks (Ming et al., 2007), CAN (Jung et al., 2005), for example. It turns out complex to integrate them in a common system. However, sometimes system requirements need to communicate them in order to provide higher services. Discovery protocols facilitate devices connection and the services negotiation like: Universal Plug and Play (UPnP) (Rhee et al., 2004), Jini Network Technology, Open Service Gateway initiative (OSGi) (Kawamura and Maeomichi, 2004). In spite of standardization attempt, the integration is complex. As example, tools used by technologies considered as automation networks standard are: European Installation Bus Tool Software (ETS) for Konnex (KNX), and LonMaker Integration Tool (LonMaker) for Lonworks. Both tools have the same purpose, but they have different design methodology. Moreover these tools do not allow realizing simulations. A consequence is that the correct functioning will not be verified until the system will be implemented. These facts introduce high temporal and economical costs. Simulation brings advantages in the design of control installation. It allows detect errors prematurely in the design phase (Denning et al., 1989). Therefore simulation is seen as a test case (Norton and Suppe, 2001), where checks are made to a higher level of abstraction. There are some control network simulators like DOMOSIM (Bravo et al., 2006) and VISIR (González et al. 2001) that are orientated to teaching methodology of facilities design. These tools present negative aspects like: are oriented to one technology or are not valid for professional environment.

The proposal gathered in this paper is to provide an environment to provide control network architectures in the digital home. The objective is that these architectures can be valid for any technology. We are going to pay special attention to network simulation task.

2 MODELING CONTROL SYSTEMS

The technologies and methods used for modelling home automated systems are based on the use of very low level technologies (Muñoz et al., 2004). The great diversity of control technologies causes

that the system creation, following bottom-up methodologies turns out to be a complex task.

The top-down methodologies are characterized essentially abstract. They require that design network is conceived before any consideration imposed by the technologies. A model based on Model Driven Architecture (MDA) (Mellor et al., 2004) and Services Oriented Architectures (SOA) (Newcomer and Lomow, 2005) is proposed. It consists in three layers called: functional, structural and technological. Each layer is defined by the corresponding question:

What features I want to offer (functional). The design will provide features, and will be composed by a set of services.

How should behave services (structural). How relate them to provide the features.

What technologies will implement the system (technological). The implementation technology will be chosen among the available technologies.

Functional is the most abstraction layer. It describes installation features. It avoids thinking about how to do this and technology implementation. In this layer, the control installation, *CI*, is defined by a set of services, *Si*, (1). Services, *Si*, need to satisfy a set of tasks, *ti* (2).

$$CI = \{S_1, S_2, ..., S_n\} \quad (1)$$

$$S_i = \{t_{i1}, t_{i2}, ..., t_{in}\} \quad (2)$$

The next level of abstraction is called structural. It is focuses on the structure and generic devices behaviour. Since the structural layer, the control installation, *CI*, is composed of a set of generic resources, *Rs*, and a wide range of connections, *C*, which are established between resources (3).

$$\begin{aligned} &CI = CI(Rs, C) / \\ &Rs = \{Rs_1, Rs_2, ..., Rs_n\}, \\ &C = \{C_1, C_2, ..., C_q\} \end{aligned} \quad (3)$$

A resource represents the entity that provides some tasks, *tij*. Resources are represented as a series of tasks, *tij*, offered to other entities (4).

$$Rs_i = Rs_i\{t_{i1}, t_{i2}, ... t_{im}\} \quad (4)$$

Connections are seen as associations between two resources, *Rsi* and *Rso* (5).

$$C_i = C_i(Rs_i, Rs_o) \quad (5)$$

Lower abstraction layer is technological. It is an instance of structural layer using a specific technology. Resources are defined by the set of tasks, *Ti,* and by set of characteristics, *CAi*, (6).

$$\begin{aligned} &R_i = R_i(T_i, CA_i) / \\ &CA_i = \{ca_{i1}, ca_{i2}, ..., ca_{ip}\} \end{aligned} \quad (6)$$

In order to design specific system implantation is necessary to make the transition from the functional to structural and finally, to technological. To achieve the first transition, the services should have all their tasks matched with some of the tasks that provide the resources of the technological layer. Therefore should there some task, *txy*, belong to any resource, *Rsx*, which is equivalent to the task, tij, required for service *Si*. This must be met for all tasks, *tij*, required by the service, *Si* (7).

$$\begin{aligned} &\exists t_{xy} \in Rs_x / \\ &t_{xy} \equiv t_{ij}, \forall t_{ij} \in S_i, Rs_x \in Rs \end{aligned} \quad (7)$$

Structural technological transition needs to match generic resources *Rs*, with technological resources *Ri*. Therefore all tasks, *txy*, from all generic resources, *Rsx*, should be matched with task, *tij*, from technological resources *Ri*. This matching must be among equivalent tasks, (8).

$$\begin{aligned} &\forall Rs_x \in Rs, \exists R_i \in R / \\ &t_{xy} \equiv t_{ji}, \forall t_{xy} \in Rs_x, t_{ji} \in R_i \end{aligned} \quad (8)$$

3 TRASGU: DESIGN ENVIRONMENT

Trasgu is a design environment based on the previously model defined. The installation is designed from a higher level of abstraction and the specific implementation aspects are chosen in last steps. It provides robustness design and reduction of development time, because transitions from the highest abstraction level to the lowest are progressive and simple.

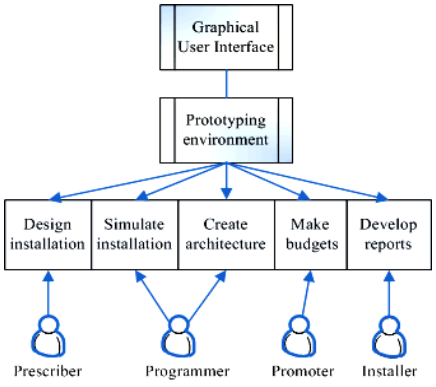


Figure 1: Environment features and their possible users.

Environment provides the following features: designing a control installation independently any technology; perform simulations to verify and validate that our design meets users specifications; create architecture in agreement to the chosen technology for implementation; making budgets with each technology; develop different types of reports. These features will be used by their respective actors who are involved in a control installation: designers, integrators, property developers, engineer... (See figure 1).

The installation design is the first task. First of all, services (2) should be added then input and output resources (4) should be added and connected with their corresponding services.

Network design is simulated in order to guarantee that design satisfies specification. This simulation will be explained in next points. The architecture creation task allows architecture generation that will implement the design in a real installation. This architecture is based on a middleware (Valdivieso et al., 2007) (Fuster et al., 2005) that provides communication among different control technologies. The budget creation task generates budgets from the technological design. The reports development task allows generate some types of reports: wired connections, devices situation, etc.

4 SIMULATION

The environment is designed to perform simulations from different abstraction levels. Simulations are from functional layer (1), (2), structural layer (3), (4) and (5) and technological layer (6). The functional simulation is responsible for simulating the behaviour of generic control network. With this validation the generic device configuration and connections can be ensured. The structural simulation includes the functional, but adds new features, like the real resources position and installation regulation. The technological simulation determines the real behaviour that the implemented control network provides.

Simulation calculations are realized by an external module that has been developed in Matlab / Simulink. The communication between the simulation calculations module and environment has been done through a Java library called jMatLink (Müller and Waller, 1999). It is responsible for sending commands to Matlab. A library called jSCA has been developed. It is responsible for encapsulating Matlab commands for facilitate the communication. The jSCA features are implemented as methods which allow: create Matlab instances; create, configure, and remove blocks in Simulink; configure parameters simulation, and obtain the simulation results. This library is a level above jMatLink. The environment calls jSCA library and this library calls jMatlink library.

Figure 2: Architecture that achieves communication among Java application and Simulink.

The architecture that achieves the communication among Java application and Simulink is showed at figure 2. The first layer is the own Trasgu that communicates with jSCA library in order to use jMatlink library that allows communication with matlab engine. It is daemon that is listening matlab commands thrown by jMatlink and uses own Simulink library for simulate.

4.1 Implementation

The environment has been implemented in Java and uses an information system implemented in XML. These XML files reflect information of the resources that can be used in the installation and the information about the current network. This information is reflected from abstraction layers defined at the model.

Environment presents distinct areas. Services that the user can add to the network are at the top of the window. Generic resources are on the left side of the window and can be used in the design. Building plant is on the centre. Services and resources can be added to this area. Services (2) define their behaviour based on inputs from input resources (4) and provide outputs to output resources (4). These functions are logical, arithmetical and comparison operators. This specification is done in a form that contains inputs, output and operators that allows interconnect them. Figure 3 shows control network designed at environment.

Simulation requires a parallel model from the functional layer in Simulink. This model is equivalent to the XML files defined in the information system used by the environment. Each resource defined in the environment is corresponded by another defined in Simulink library created for control network simulation. The Simulink model consists in a block set formed by input blocks, output blocks, input-output blocks, and connections in order to establish a logic circuit. For this purpose

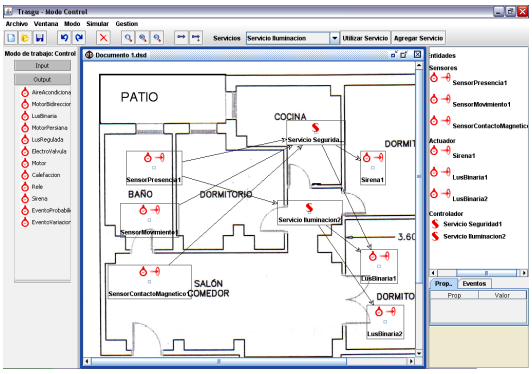


Figure 3: Network designed at environment.

there are two block types: discrete and continuous. The discrete blocks are for digital devices, and continuous are for analogic devices.

The simulation starts creating an equivalent control network in Simulink. The corresponding network blocks must be added to create the mdl file. Parameters of each block are configured and connections among blocks are created. The last step is to add probability or schedules events for activate sensors. Finally the mdl file is created and Simulink executes the simulation and the environment reads values from all Simulink network blocks and represents them at environment.

5 CONCLUSIONS

This paper presents a design and simulation environment of control networks in digital home. The objective is to facilitate the tasks of designing and validating control networks. A top-down methodology is proposed, where technology implementation choice is left to the end phase of designing process. The environment kernel is the control network model around different abstraction layers: functional, structural and technological.

Simulation is presented as a new phase in methodology of design control network. It reduces costs and helps us to design effective and efficient control networks.

The future work is aimed to deepening in aspects of results generalization: new models for structural and technological layers, simulation of technological layer and interactive simulation module.

REFERENCES

Bravo, C., Redondo, M., Ortega, M.., and Verdejo, M.F., 2006. *Collaborative environments for the learning of design: a model and a case study in Domotics.* Computers & Education, Vol. 46, No. 2, 152-173.

Denning, P. J., Comer, D. E., Gries, D., Mulder, C., Tucker, A., Turner, A. J., Young, P. R., 1989. *Computing as a discipline.* Communications of the ACM, Vol.32 No.1, 9-23

Fuster, A. and Azorín, J., 2005. Hogar digital. El camino de la domótica a los ambientes inteligentes.I Encuentro Interdisciplinar de Domótica 2005, pp 45-54

Fuster, A., de Miguel, G. and Azorín, J., 2005. *Tecnología Middleware para pasarelas residenciales.* Hogar digital. El camino de la domótica a los ambientes inteligentes.I Encuentro Interdisciplinar de Domótica 2005, 87-102.

González, V. M., Mateos, F., López, A.M., Enguita, J.M., García M.and Olaiz, R., 2001. *Visir, a simulation software for domotics installations to improve laboratory training*, Frontiers in Education Conference, 31st Annual Vol. 3, F4C-6-11.

Haenselmann, T., King, T., Busse, B., Effelsberg, W. and Markus Fuchs, 2007. *Scriptable Sensor Network Based Home-Automation*. Emerging Directions in Embedded and Ubiquitous Computing, 579-591.

Jung, J., Park, K. and Cha J., 2005. *Implementation of a Network-Based Distributed System Using the CAN* Protocol. Knowledge-Based Intelligent Information and Engineering Systems. Vol. 3681.

Kawamura, R. and Maeomichi, 2004. *Standardization Activity of OSGi (Open Services Gateway Initiative)*, NTT Technical Review, Vol. 2, No. 1, 94-97.

Mellor, S., Scott K., Uhl, A. and Weise, D., 2004. *MDA Distilled, Principles of Model Driven Architecture*, Addison-Wesley Professional.

Min, W., Hong, Z., Guo-ping, L. and Jin-hua, S., 2007. *Networked control and supervision system based on LonWorks fieldbus and Intranet/Internet.* Journal of Central South University of Technology, Vol. 14, No.2, 260-265.

Müller, S. and Waller, H., 1999. *Efficient Integration of Real-Time Hardware and Web Based Services Into MATLAB*. 11th European Simulation Symposium and Exhibition, 26-28.

Muñoz, J., Fons, J., Pelechano, V. and Pastor, O., 2003. *Hacia el Modelado Conceptual de Sistemas Domóticos*, VIII Jornadas de Ingeniería del Software y Bases de Datos. Universidad de Alicante, 369-378

Newcomer, E. and Lomow, G., 2005. *Understanding SOA with Web Services*. Addison Wesley.

Norton S. and Suppe F., 2001. *Why atmospheric modeling is good science*. MIT Press. p. 88-133.

Rhee, S., Yang, S., Park, S., Chun, J.and Park, J., 2004. *UPnP Home Networking-Based IEEE1394 Digital Home Appliances Control.* Advanced Web Technologies and Applications, Vol. 3007, 457-466.

Valdivieso, R.J., Sánchez, J.G., Azorín, J. and Fuster, A., 2007. *Entorno para el desarrollo y simulación de arquitecturas de redes de control en el hogar.* II Internacional Simposium Ubiquitous Computing and Ambient Intelligence.

MULTI-LEVEL CONTROL OF AN INTELLIGENT WHEELCHAIR IN A HOSPITAL ENVIRONMENT USING A CYBER-MOUSE SIMULATION SYSTEM

Rodrigo A. M. Braga, Marcelo Petry, Eugenio Oliveira and Luis Paulo Reis
Artificial Intelligence and Computer Science Lab-LIACC, 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, eco@fe.up.pt, lpreis@fe.up.pt

Keywords: Intelligent Wheelchair, Intelligent Robotics, Strips Planning, Path Planning, A* Algorithm.

Abstract: The development of intelligent wheelchairs is 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 the integration of a robotic simulator with our intelligent wheelchair shared control and planning modules. An adapted version of the free Cyber-Mouse robotic simulator was used to simulate the movement of the intelligent wheelchair in a hospital environment. Adaptations of the subsumption architecture, Strips Planning and A* Algorithms were employed and integrated to allow wheelchair intelligent behavior. The experimental results have demonstrated the success of the integration of these algorithms in our simulator allowing very safe motion of the intelligent wheelchair in the simulated hospital environment. Also, the adapted Cyber-Mouse simulator proved its capability and robustness in simulating the hospital environment and wheelchair physic characteristics.

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 safety and comfortable Intelligent Wheelchairs (IW) to practical uses. The main functions of IWs are (Faria, 2007a)(Faria, 2007b)(Jia, 2005):

•Interaction with the user, including hand based control, such as, joystick, keyboard, mouse, touch screen; voice based control, such as audio; vision based control, such as camera; and other sensor based control, such as pressure sensors.

•Afford Services, for instance autonomous navigation (with safety, flexibility and robust obstacle avoidance), communication with other devices (like automatic doors).

This paper discusses the application of a Cyber-Mouse simulator, developed at the Univ.Aveiro in Portugal (Lau, 2002) in the study, development and test of shared control and high-level planning algorithms applied in an IW operating in a hospital environment.

A shared control algorithm was tested, allowing IW automatically avoids danger situations. Also, typical algorithms used in most intelligent robotics applications were applied in the control of the IW and simulated in the hospital scenery. Blended with the control, a motion planner was developed capable of generating the behavior/path commands according to an a-priori 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).

Cyber-Mouse is a competition among virtual robots, which takes place in a simulated environment running in a computer network. The simulation system creates a virtual arena with a starting grid, a target area, signalled by a beacon, and populated with obstacles. It also creates the virtual bodies of the robots. The simulator estimates sensor measures which are sent to the agents. Reversely, receives and apply actuating orders coming from agents (Lau, 2002).

The paper is subdivided in 5 different sections: section 2 presents some brief definitions of the control algorithms applied in this research; section 3 contains a description of the system developed; section 4 provides experimental tests and result discussion and section 5 presents the final conclusions and points out some future research topics.

2 INTELLIGENT WHEELCHAIR CONTROL ALGORITHMS

To test the simulator, the algorithms used to implement the control system and algorithms used for the high-level planning system were:

- ***Adapted Subsumption Architecture.*** (Brooks, 1991) (Russell, 2002)(Ferber, 1991) for basic moving;
- ***Knowledge Representation using Grids.*** Spatial knowledge for robotic movement, navigation and planning (Borenstein, 1991)(Thrun, 1996).
- ***A* Search Algorithm for Trajectory Generation.*** used to find a path from a given initial node to a given goal node (Shapiro, 2000) (Hart, 1968) (Barr, 1986).
- ***Adapted Strips Planning Algorithm.*** Planning is the task of searching for an action sequence to achieve a given goal (Russell, 2002)(Weld, 1999)(Fikes, 1971) (Bonet, 2001).

3 SYSTEM DESCRIPTION

The hospital environment was created modifying the Cyber-Mouse labyrinth definition XML file that contains positions of a set of walls of arbitrary shape and beacons that emit signals. Figure 1 shows our hospital modified floor plan.

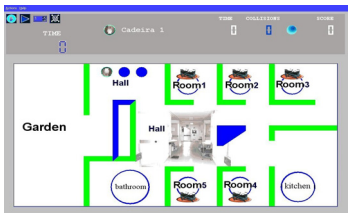


Figure 1: Hospital environment represented using the Cyber-Mouse simulator.

The hospital representation is composed of five rooms: bathroom, kitchen, hall and garden and a set of walls of different shapes. IWs are represented by modified robotic agents. UDP Sockets are used for communication between the robotics agents and the simulator. The communication is based in dispatching XML messages of five types: request register; refuse reply; acceptation reply; sensor data, action order.

Figure 2 presents the developed agent control software interface. A communication configuration panel can be seen in left side of the interface, the sensor values are depicted in the middle of the interface. In the right side, there are three function modules. The first module has the following functions: semi-automatic control, revolve in a point direction, move to a point, move through planning path.

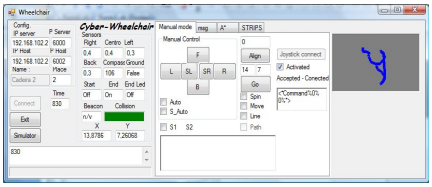


Figure 2: Intelligent Wheelchair software interface.

The semi-automatic control, also denoted shared control, was constructed with fusion between user command and subsumption architecture. This way it achieves safe movement allowing automatic obstacle avoidance. Figure 3 presents the implemented architecture.

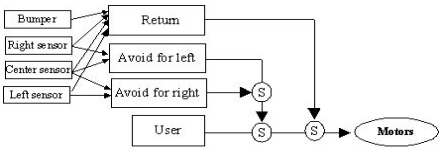


Figure 3: Subsumption architecture used in shared control.

Figure 4 shows the path generation interface module. The path generation was implemented using the A* algorithm. The state world information is loaded into the system using a XML file. A grid of 56x28 cells representing the world- state, for this development, was considered as the world state environment. Also, object extension techniques were used, in such manner that a robot may be represented as a point in the system without the risk of collision with the walls.

The path generation module searches for a path, starting in the actual point and finishing in the user selected point. The path generated can be seen in the interface provided.

The last module (see Figure 5) is an automatic planning module developed based in Strips Planning Algorithm with inverse chaining. Initially, we developed five general high-level objectives: *"go-bathroom", "go-garden", "go-room", "go-kitchen", "go-consulting_room"*. These general objectives generate different final objectives, depending on the patient, wheelchair and location set for the actions. For example, the action *patient 1 with wheelchair 2 go-room for Room 1* results in the objective: *patient 1 in Room 1, patient 1 without wheelchair, wheelchair 2 free and situated in hall*. In this way, it is allowed to set many different final objectives and since the plan depends on the initial

state, it will have different sequences and number of actions for arriving the final objective.

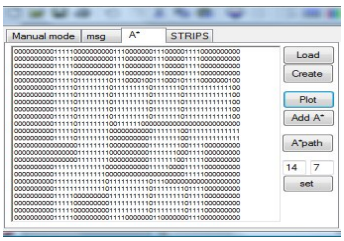


Figure 4: Path generation module interface.

After plan generation, we can execute it in the simulated environment. Now, the system uses the path generation module for the generation the paths for movement actions and automatic move control module to execute the movement through the path generated with collision avoidance. This module executes step by step the actions of the plan using the other modules whenever it is necessary.

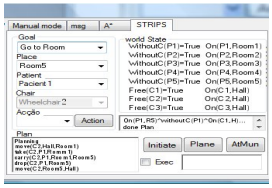


Figure 5: Planning module interface.

4 EXPERIMENTS AND RESULTS

This section presents a simple operation example of our robotic agent and results gathered in this experiment. After initiating the simulator, the configuration, map and grid file are loaded in the simulator, through a XML file.

As focus is primarily testing the simulation system and the implementation of an ordinary multi level control, sensor and motor noise were disable. Another simplification is the robot data position, which is provided by the simulator GPS system that represents a robust odometry system.

The simulator configuration is presented in Table 1. The simulator allows inserting noise in the robots sensors and actuators. However in this stage of development, the sensor and actuators noise was not used.

The first test was the use of the A* algorithm for searching the path from point (x=9,5 and y=13,2) to point (x=14 and y=7). The plan devised is presented in Table 3. The plan is presented in inverse order, i.e., the first point is in the table bottom and last point is in the table top.

Table 1: XML file to configure simulator.

```
<Parameters SimTime="18000" CycleTime="80" CompassNoise="0"
BeaconNoise="0"  ObstacleNoise="0" MotorsNoise="0"
RunningTimeout="1350" GPS="On" ShowActions="False" Nbeacons="5"
Lab="C:/CiberToolsWindows-1.2.0/Labs/My2Lab.xml"
Grid="C:/CiberToolsWindows1.2.0/Labs/MyGrid.xml" />
```

The resulting movements are shown in Figure 6. The black letters represent before of move, the blue line represent plan path and the red line represent the path travelled. The number '0' represents free space, '1' represents the expanded wall. In this test, the move was free collision and with minimum trajectory error.

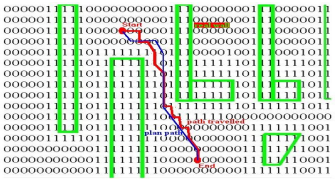


Figure 6: Movement result.

Next step was to generate a comprehensive plan using the implemented path generator. The goal in this case was to pick up patient 1 in bedroom 1; carry him to bedroom 5, to make company to patient 5; and finally go back to the lobby. In this example we had as objective: "Go to the bedroom", as Place: 5 and Patient: 1. This parameters result as final objective: *On(P1,R5) ^ WithoutC(P1) ^ On(C2,H).* The world state before the action, the resulting plan and the following state are represented in Figure 7.

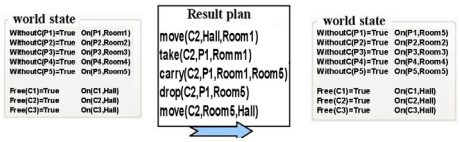


Figure 7: World state s, plan and world state s'.

The grid containing the final state and the travelled path, based on the plan mentioned before, can be observed in Figure 8. This grid contains the four basic movements that the IW needed to perform to achieve the final objective: from the initial point until the lobby, from lobby to bedroom 1 (to pick up patient 1), then to the bedroom 5 and after leave patient 1 going back (empty) to the lobby. The time to conclude movement in automatic control was 982 second and for manual control was 870 seconds.

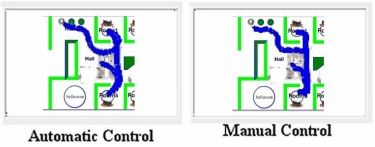


Figure 8: Final movement result.

5 CONCLUSIONS

This paper presented the implementation of a robotic Intelligent Wheelchair, simulating its behavior in an adapted Cyber-Mouse simulator. This agent resulted of a modular system composed of three modules. The first one is a shared control that merges user commands with the information received by the IW sensors. This advantage prevents the collision with objects in its way becoming movement system more reliable and safe. The algorithms developed use a subsumption architecture: once there are objects close to the wheelchair, the user commands are disabled and deflect commands are sent to the motors, ensuring a safe motion.

A module was develop using the A* algorithm as path generator to calculate the shortest path from the robot's actual position to the objective point. The third module was implemented to plan some usual tasks in a hospital environment, using Strips planning to solve these interactions. In most of the cases Strips proved to be efficient, delivering complete plans, with exceptions presented under Sussman anomaly effects, where one operation cancels other in the plan.

It was also demonstrated that Cyber-Mouse is a friendly tool to test control algorithms, IW navigation and its interaction with hospital environment.

Future research directions include the improvement to non-linear planning and upgrade from A* to D* algorithm, once it's preferable in such a dynamic environment. To be fully intelligent, it's not enough the wheelchair to plan its own path or share its control. It is also necessary to communicate with other intelligent wheelchairs and devices like doors activation systems, elevators and lights. Due to this, it is intended to perform an in depth study in proper methodologies to implement these capacities in the wheelchair, and this way, implement some functions as cooperative behavior among a group of IW and collaboration among the user and the system. In Cyber-Mouse it is necessary to increase its present simulation capacities, from its actual three IW, enabling hybrid systems test, where real and virtual IW 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 wouldn't be necessary a large number of real IW. Still, in Cyber-Mouse, it is necessary to implement noise treatment in the motors and sensors to have actions in the simulated wheelchair closer to those of real wheelchairs.

REFERENCES

Barr, A. Cohen, P. Feigenbaum, E., 1986. Handbook of Artificial Intelligence. Addison Wesley. Vol. 1.

Bonet, B. and Geffner, H., 2001. Heuristic search planner 2.0. The AI Magazine, 22(1), pp.77–80.

Borenstein, J., Koren, Y., 1991. The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots, IEEE Journal of Robotic and Automation, v. 7, n. 3, pp. 278-288.

Brooks, R. A., 1991. Intelligence Without Reason. Proc. 12th Int. Joint Conference on Artificial Intelligence - IJCAI-91, Sydney, Australia. pp.569-595.

Faria, P. M., Braga, R A. M., Valgôde, E., Reis L. P., 2007a. Platform to Drive an IW using Facial Expressions. Proc. 9th Int. Conf. on Enterprise Information Systems, HCI, ICEIS'2007. pp.164-169. Funchal, Portugal. June 12–16.

Faria, P. M., Braga, R. A. M., Valgôde, E., Reis, L. P., 2007b. Interface Framework to Drive an Intelligent Wheelchair Using Facial Expressions. IEEE Int. Symp. on Industrial Electronics, pp. 1791-1796, June 4-7.

Ferber, J.; Gasser, L., 1991. Intelligence artificielle distribuée. In: Int. WorkShop on Expert Systems & Their Applications, 10., Avignon. Cours n. 9. France.

Fikes, R., Nilsson, N. J., 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. IJCAI 1971. pp.608-620.

Hart, P. E., Nilsson, N. J. and Bertram R., 1968, A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2):pp.100–107.

Jia, P. and Hu, H., 2005. Head Gesture based Control of an Intelligent Wheelchair. CACSUK-11th Ann. Conf. Chinese Aut. Comp. Society in the UK, Sheffield, UK.

Lau, N., Pereira, A., Melo, A., Neves, A. and Figueiredo, J., 2002. Ciber-Rato: Um Ambiente de Simulação de Robots Móveis e Autónomos. Rev. DETUA, Vol. 3, Nr. 7, pp.647-650, Sept.

Luo, R. C., Chen, T. M. and Lin, M. H., 1999. Automatic Guided IW System Using Hierarchical Grey-Fuzzy Motion Decision-Making Algorithms, Proc. IEEVRSJ 1999 Int. Conf. on Intelligent Robots and Systems.

Russell, S. and Norvig, P., 2002. Artificial Intelligence: A Modern Approach. Prentice Hall, E.Cliffs, NJ, 2nd ed.

Shapiro S.C., 2000. Encyclopedia of Artificial Intelligence. Wiley-Interscience. May.

Thrun, S. and Bücken, A., 1996. Integrating Grid-Based and Topological Maps for Mobile Robot Navigation, In: Proc. 13th National Conference on Artificial Intelligence AAAI, Portland, Oregon, pp.944-950.

Weld, D. S., 1999. Recent advances in AI planning. AI Magazine, pp.93–123

ROBUST CONTROL OF THE C5 PARALLEL ROBOT

B. Achili*+, B. Daachi*, A. Ali-Cherif+ and Y. Amirat*
+*Laboratoire d'Informatique Avancée de Saint Denis, 2 rue de la liberté 93526 Saint Denis Cedex, France*
**Laboratoire Images, Signaux et Systèmes intelligents, 122-124 rue Paul Armangot, 94400 Vitry/seine, France*
achili@ai.univ-paris8.fr

Keywords: Parallel robot, Robust control, Stability analysis.

Abstract: This paper deals with the dynamic control of a parallel robot with C5 joints. Computed torque control and robust control have been studied and implemented. For this purpose, we have used the inverse dynamic model whose parameters have been experimentally identified. The closed loop stability has been studied using the Lyapunov principle. The addition of a robustness term based on sliding mode technique ensures good tracking performances. The experimental results show the effectiveness of the robust control.

1 INTRODUCTION

Parallel manipulator is a closed-loop mechanism in which the end-effector (mobile platform) is connected to the base by at least two independent kinematic chains. Compared to the serial ones, the parallel architectures have potential advantages in terms of stiffness, accuracy, high speed and payload. They are widely applied to the following fields, like the Pick and Place operation in food, medicine, electronic industry, etc.

Due to their complex architecture, precise and robust control of parallel machines is a hard and open problem which has been widely addressed in the litterature. When the task requires fast motion of robot and high precision, it is very important to design a controller with good performances in order to match the mechanism. In literature, various control methods are proposed such as proportional, integration, derivative (PID) control, computed torque control (Middleton and Goodwin, 88) adaptive control (Slotine and Li, 88), neural networks control (Miller et al, 87), fuzzy control, fuzzy adaptive control.

Computed torque control has been proposed in the litterature. The latter requires the exact knowledge of inverse dynamic model. In theory, it ensures the decoupling and the linearization of equations of robot motion, resulting in a uniform response for any robot configuration . This technique is more efficient in term of precision for high moving than PD and PID linear control. However it is sensitive to the parameter variations and external disturbances of the system (Zhiong et al, 07). In practice, the dynamic model of robot can not be exactly known. Therefore in order to circumvent the problem of dynamic model uncertainties, an adaptive technique is needed.

In literature, several works have been published in the field of intelligent control methods such as Fuzzy control, neural network control (Miller et al, 87). Thus, a fuzzy neural network hybrid control (FNN) is proposed. In this control technique the hybrid control system, combines the computed torque controller, the FNN uncertainty observer and a compensated controller to control the position of a slider of the motor–toggle servomechanism. Recently, a new approach, combining the computed torque control with fuzzy control has been proposed in literature. The latter is used to approximate lumped uncertainty due to parameters variations. Among other recent work, an on-line updated PID algorithm is proposed (Chen and Huang, 2004).

In this paper, we have addressed the robust control of a parallel robot with C5 joints. This type of control allows us to improve the trajectory tracking for fast motions. This approach is based on sliding mode technique. It consist to add a compensation term in the control law in order to compensate the identification and modeling errors.

This paper is organized in five sections. The first one describes the mechanical architecture of the C5 parallel robot. The second section presents the dynamic model and its properties. Section 3 is devoted to the control law synthesis. Section 4 is dedicated to the presentation and analysis of the experimental re-

sults. Finally, a conclusion and some perspectives are given in the last section.

2 DESCRIPTION OF THE C5 PARALLEL ROBOT

The C5 parallel robot consists of a static part and a mobile part connected together by six actuated links. Each segment is embedded to the static part at point A_i and linked to the mobile part through a spherical joint attached to two crossed sliding plates at point B_i (Fig. 1)

Theoretical study concerning this architecture has been presented in the literature. The C5 links parallel robot is equipped with six linear actuators; each of them is driven by a DC motor. Each motor drives a ball and screw arrangement. The position measurements are obtained from six incremental encoders, which are tied to the DC motors.

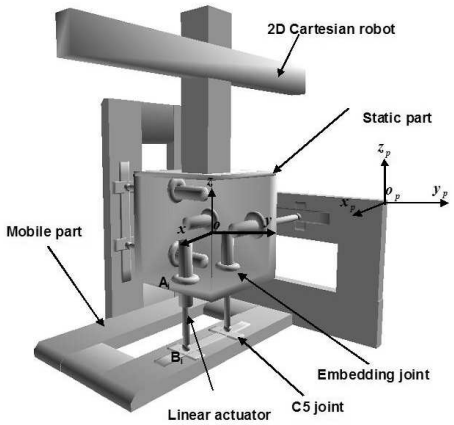


Figure 1: Parallel robot.

3 DYNAMIC MODEL AND PROPERTIES

The dynamic model of the considered parallel robot is given by the following equation :

$$\Gamma = M(q)\ddot{q} + H(q,\dot{q}) \tag{1}$$

with

- q the (6×1) vector of joint positions
- $\dot{q}$ the (6×1) vector of joint velocities
- $\ddot{q}$ the (6×1) vector of joint accelerations
- $M(q)$ the (6×6) inertia matrix
- $H(q,\dot{q})$ the(6×1) vector of gravitational forces, frictions, Coriolis centripetal forces and other dynamics.
- Γ the vector of the torques.

The robot dynamics (1) have physical properties that can be used in the control law synthesis :

Property 1. The matrix M is Symmetric Positive Definite (*SPD*).

Property 2. The matrix C can be chosen so that $\dot{M} - 2C$ is skew symmetric.

Matrices M and H are identified by using least squares method. We note by $\hat{M}$ and $\hat{H}$ the estimation of M and H respectively. The detail of this identification method is given in (Janot et al, 07).

4 CONTROL LAW SYNTHESIS

Assuming that the dynamic model is exactly identified (case of negligible identification errors), we can use the following control law :

$$\Gamma = \hat{M}(q)\ddot{q}_r + \hat{H}(q,\dot{q}) \tag{2}$$

with

$$\ddot{q}_r = \ddot{q}^d + k_v\dot{e} + k_p e \tag{3}$$

$$e = (q^d - q) \tag{4}$$

$$\dot{e} = (\dot{q}^d - \dot{q}) \tag{5}$$

where

- e is the trajectory tracking error vector
- $q^d, \dot{q}^d, \ddot{q}^d$ are respectively, desired joint positions, velocities and accelerations
- k_p and k_v are respectively, diagonal positive definite matrices that represent the proportional and derivatives gains.

The combination of equations (2 and (3) gives the following equation:

$$\ddot{e} + k_v\dot{e} + k_p e = 0 \tag{6}$$

The solution of equation (6) is globaly exponentially stable. In our case, the functions of the dynamic model (matrices M and H) are estimated by least squares method. The parameters of these functions are fixed. The robot carries out different tasks and generally the identification error is never close to zero. It is then imperative to take into account these identification errors. For this purpose, we introduce in the control law a term of robustness δu based on

the sliding mode technique. The control law is then given by:

$$\Gamma = \hat{M}(q)\ddot{q}_r + \hat{H}(q,\dot{q}) + \delta u \tag{7}$$

In closed loop, the equation of the dynamic error is given by :

$$\ddot{e} + k_v\dot{e} + k_p e = \hat{M}^{-1}\left[(M-\hat{M})\ddot{q} + (H-\hat{H}) - \delta u\right] \tag{8}$$

We consider the state form of the equation (9) :

$$\dot{\mathbf{e}} = A\,\mathbf{e} + B\hat{M}^{-1}\left[(M-\hat{M})\ddot{q} + (H-\hat{H}) - \delta u\right] \tag{9}$$

with

- $\mathbf{e} = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$
- $A = \begin{bmatrix} 0 & I \\ -k_p & -k_v \end{bmatrix}$
- $B = \begin{bmatrix} 0 \\ I \end{bmatrix}$
- 0 and I are the$(n \times n)$ zero and identity matrix respectively.

Consider a new signal error s:

$$s = C\mathbf{e} \tag{10}$$

with

$$C = \begin{bmatrix} \Lambda & I \end{bmatrix} \tag{11}$$

Λ is a positive diagonal matrix, such that the transfer matrix $[C(pI-A)^{-1}B]$ is strictly positive real (SPR). For the purpose of the stability analysis we use the formulation given in (Meddah, 98).

We choose δu inspired from sliding-mode theory as follows :

$$\delta u = \beta sign(s) \tag{12}$$

where β is the sliding gain.

For stability study, we use the following Lyapunov function V :

$$V(t) = \frac{1}{2}\mathbf{e}^T P\mathbf{e} \tag{13}$$

where P is a symmetric positive definite matrix solution of the Lyapunov equation:

$$A^T P + AP = -Q$$
$$PB = C^T$$

Q a symmetric positive definite matrix.

The time derivative of the Lyapunov function (13) is expressed by the following equation:

$$\begin{aligned}\dot{V}(t) &= -\frac{1}{2}e^T Q\mathbf{e} \\ &\quad + s^T\hat{M}^{-1}\left[(M-\hat{M})\ddot{q} + (H-\hat{H}) - \beta sign(s)\right]\end{aligned} \tag{14}$$

Consequently $\dot{V} \leq 0$ when the following inequality is satisfied :

$$\beta \geq \frac{\lambda_{\hat{M}}^{\max}}{\lambda_{\hat{M}}^{\min}}\left\|(M-\hat{M})\ddot{q} + (H-\hat{H})\right\| \tag{15}$$

where $\lambda_{\hat{M}}^{\max}$ and $\lambda_{\hat{M}}^{\min}$ are respectively the greatest eigen value and the smallest eigen value of $\hat{M}^{-1}$.

If we choose the gain β, according to(15) we obtain :

$$\dot{V}(t) \leq -\frac{1}{2}\mathbf{e}^T Q\mathbf{e}$$

for any $t \geq 0$, thus s is bounded.

To prove that $s \to 0$ when $t \to \infty$, we can apply a Barbalat lemma to the following non negative function :

$$V_1(t) = V(t) - \int_0^t \left(\dot{V}(t) + \frac{1}{2}\mathbf{e}^T Q\mathbf{e}\right) d\tau \tag{16}$$

$$\dot{V}_1(t) = -\frac{1}{2}\mathbf{e}^T Q\mathbf{e} \tag{17}$$

$\dot{V}_1(t)$ is uniformly continuous.

According to Barbalat lemma we can conclude that $\dot{V}_1(t) \to 0$ and consequently $\mathbf{e} \to 0$ and $s \to 0$. Therefore the system represented by equation (9) is asymptotically stable.

Even if the value of the gain β is determined from condition (15), it is difficult to find this value as matrices M and H are unknown. Thus it is not possible to obtain the exact value of $((M-\hat{M})\ddot{q} + (H-\hat{H}))$. In practice, β is chosen heuristically. Note that the term *sign* used in (14) produces the chattering phenomenon in the control input. In order to avoid this drawback, Slotine and Li (Slotine and Li, 91) propose to replace the function *sign* by the function *sat*. (saturation)

5 EXPERIMENTAL RESULTS

In this section we present the experimental results of the application of the control laws described in section 4. These control approach is compared to a PID control. A chirp function is used as a desired trajectory, the frequency varies between $0.1hz$ and $0.3hz$. The trajectory tracking error concerning the first axis (filtered by the 4th order Butterworth) is shown in Fig. 2. Concerning other axis, we obtained the same performance as the first one.

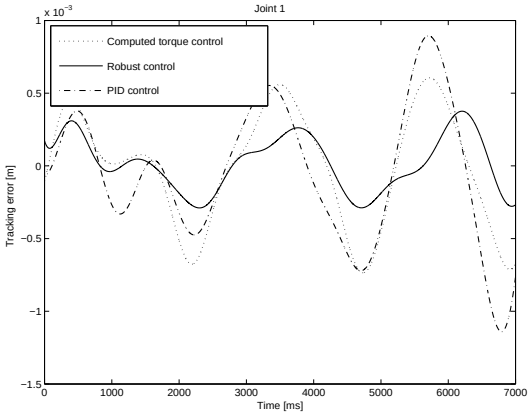


Figure 2: Tracking error of computed torque control, robust control and PID control.

5.1 Discussion

Figure 2 show that computed torque approach clearly improve the tracking performances compared to the PID control, but it remains insufficient because the precise values of M and H are difficult to obtain due to measuring errors, environment and parameters variations. Therefore, we can conclude that computed torque method exhibits good performances only when robot dynamics model is precisely identified.

However, we obtained good tracking performances when the robust control (control law with compensation term δu) is used. Besides, we also note that the errors increase for fast motions due to PID control. For robust control, these errors remain small with respect to PID control and Computed torque control errors.

6 CONCLUSIONS

In this paper, we implemented a sliding mode approach for the robust control of the C5 parallel robot. The stability of the system in closed loop with the control law including a compensation term is guaranteed. Thereafter a comparative study of above approaches show that the control using a robust term ensures a good trajectory tracking compared to the others presented approaches. The experimental results show clearly that the robustness term, based on the sliding mode method, reduces the effect of the identification errors. In our short term project, we propose a new control law such that the identification and modeling errors will be compensated in an adaptive way.

REFERENCES

Zhiyong Y., Jiang W., Jiangping M., 2007. *Motor-mecanism dynamic model based neural network optimized computed torque control of a high speed parallel manipulator*, Mechatronics 17 381-390.

Song Z., Yi J., Zhao D., Li X., 2005. *A computed torque controller for uncertain robotic manipulator system: Fuzzy approach*, Fuzzy Sets and Systems 154 208-226.

Chen J., Huang T.C., 2004 *Applying neural networks to on-line updated PID controllers for nonlinear process control*, J Process Control; 14:211–30.

Middleton R.H., Goodwin G.C., 1988. *Adaptive computed torque control for rigid link manipulators*, System Control Lett. 10 9–16.

Slotine J.J.E., Li W., 1988. *Adaptive manipulator control: a case study*, IEEE Trans. Automat. Control 33 995–1003.

Janot A., Bidard C., Gosselin F., Gautier M., Keller D., Perrot Y., 2007, *Modelingand Identifi cation of a 3 DOF Haptic Interface*,IEEE International Conference on Robotics and Automation Roma, Italy, 10-14 pp4949-4955.

Meddah Y. D., 1998. *Identification et commande neuronales de systèmes non linéaires : Application aux systèmes Robotisés*. Thèse de doctorat de l'université Pierre et Marie Curie.

Miller W.T., Glanz F.H., Kraft L.G., 1987, *Application of a general learning algorithm to the control of robotic manipulators*, Int. J. Robot. Res. 6 84–98.

LOCALIZATION OF A MOBILE ROBOT BASED IN ODOMETRY AND NATURAL LANDMARKS USING EXTENDED KALMAN FILTER

Andre M. Santana, Anderson A. S. Sousa, Ricardo S. Britto, Pablo J. Alsina
and Adelardo A. D. Medeiros
Federal University of Rio Grande do Norte, Natal-RN, Brazil
{*andremacedo, abner, rbritto, pablo, adelardo*}*@dca.ufrn.br*

Keywords: Robot Localization, Kalman Filter, Sensor Fusion.

Abstract: This work proposes a localization system for mobile robots using the Extended Kalman Filter. The robot navigates in an known environment where the lines of the floor are used as natural landmarks and identifiqued by using the Hough transform.The prediction phase of the Kalman Filter is implemented using the odometry model of the robot. The update phase directly uses the parameters of the lines detected by the Hough algorithm to correct the robot's pose.

1 INTRODUCTION

Borenstein et al. have classified the localization methods in two great categories: relative localization methods, which give the robot's pose relative to the initial one, and absolute localization methods, which indicate the global pose of the robot and do not need previously calculated poses (Borenstein et al., 1997).

As what concerns wheel robots, it is common the use of encoders linked to wheel rotation axes, a technique which is known as odometry (Borenstein et al., 1997). However, the basic idea of odometry is the integration of the mobile information in a determined period of time, what leads to the accumulation of errors (Park et al., 1998).

The techniques of absolute localization use landmarks to locate the robot. These landmarks can be artificial ones, when introduced in the environment aiming at assisting at the localization of the robot, or natural ones, when they can be found in the proper environment. It is important to underline that even the techniques of absolute localization are inaccurate due to noises produced by the manipulated sensors.

Literature shows works using distance measures to natural landmarks (walls, for example) to locate the robot. The obtaining of these measures is generally made with the help of sonar, laser and computational vision (Lizzaralde et al., 2003; Kim and Kim, 2004; Pres et al., 1999).

Bezerra used in his work the lines of the floor composing the environment as natural landmarks (Bezerra, 2004). Kiriy and Buehler, have used extended Kalman Filter to follow a number of artificial landmarks placed in a non-structured way (Kiriy and Buehler, 2002). Launay et al. employed ceiling lamps of a corridor to locate the robot (Launay et al., 2002).

This paper proposes a system enabling to locate a mobile robot in an environment in which the lines of the floor form a bi-dimensional grid. To turn it possible, the lines are identified as natural landmarks and its characteristics, as well as the odometry model of the robot, are incorporated in a Kalman Filter in order to get its pose.

2 THE KALMAN FILTER

The modeling of the Discrete Kalman Filter - DKF presupposes that the system is linear and described by the model of the equations of the system (1):

$$\begin{cases} \mathbf{s}_t = \mathbf{A}_t \mathbf{s}_{t-1} + \mathbf{B}_t \mathbf{u}_{t-1} + \gamma_t \\ \mathbf{z}_t = \mathbf{C}_t \mathbf{s}_t + \delta_t \end{cases} \quad (1)$$

in which $\mathbf{s} \in R^n$ is the vector of the states; $\mathbf{u} \in R^l$ is the vector of the control entrances; $\mathbf{z} \in R^m$ is the vector of measurements; the matrix $n \times n$, $\mathbf{A}$, is the transition

matrix of the states; **B**, $n \times l$, is the coefficient matrix on entry; matrix **C**, $m \times n$, is the observation matrix; $\gamma \in R^n$ represents the vector of the noises to the process and $\delta \in R^m$ the vector of measurement errors. Indexes t and $t-1$ represent the present and the previous instants of time.

The Filter operates in prediction-actualization mode, taking into account the statistical proprieties of noise. An internal model of the system is used to updating, while a retro-alimentation scheme accomplishes the measurements. The phases of prediction and actualization to DKF can be described by the systems (2) and (3) respectively.

$$\begin{cases} \bar{\mu}_t = \mathbf{A}_t \mu_{t-1} + \mathbf{B}_t \mathbf{u}_{t-1} \\ \bar{\Sigma}_t = \mathbf{A}_t \Sigma_{t-1} \mathbf{A}_t^T + \mathbf{R}_t \end{cases} \tag{2}$$

$$\begin{cases} \mathbf{K}_t = \bar{\Sigma}_t \mathbf{C}_t^T (\mathbf{C}_t \bar{\Sigma}_t \mathbf{C}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - \mathbf{C}_t \bar{\mu}_t) \\ \Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \bar{\Sigma}_t \end{cases} \tag{3}$$

The Kalman Filter represents the vector of the states $\mathbf{s}_t$ by its mean μ_t and co-variance Σ_t. Matrixes **R**, $n \times n$, and **Q**, $l \times l$, are the matrixes of the covariance of the noises of the process (γ) and measurement (δ) respectively, and matrix **K**, $n \times m$, represents the prot of the system.

A derivation of the Kalman Filter applied to nonlinear systems is the Extended Kalman Filter - EKF. The idea of the EKF is to linearize the functions around the current estimation using the partial derivatives of the process and of the measuring functions to calculate the estimations, even in the face of nonlinear relations. The model of the system to EKF is given by the system (4):

$$\begin{cases} \mathbf{s}_t = g(\mathbf{u}_{t-1}, \mathbf{s}_{t-1}) + \gamma_t \\ \mathbf{z}_t = h(\mathbf{s}_t) + \delta_t \end{cases} \tag{4}$$

in which $g(\mathbf{u}_{t-1}, \mathbf{s}_{t-1})$ is a non-linear function representing the model of the system, and $h(\mathbf{s}_t)$ is a nonlinear function representing the model of the measurements.

Their prediction and actualization phases can be obtained by the systems of equations (5) and (6) respectively.

$$\begin{cases} \bar{\mu}_t = g(\mathbf{u}_{t-1}, \mu_{t-1}) \\ \bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{R}_t \end{cases} \tag{5}$$

$$\begin{cases} \mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - h(\bar{\mu}_t)) \\ \Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t \end{cases} \tag{6}$$

The matrix **G**, $n \times n$, is the jacobian term linearizes the model and **H**, $l \times n$ is the jacobian term linearizes the measuring vector. Such matrixes are defined by the equations (7) e (8).

$$\mathbf{G}_t = \frac{\partial g(\mathbf{u}_{t-1}, \mathbf{s}_{t-1})}{\partial \mathbf{s}_{t-1}} \tag{7}$$

$$\mathbf{H}_t = \frac{\partial h(\mathbf{s}_t)}{\partial \mathbf{s}_t} \tag{8}$$

Next we will describe the modeling of the problem, as well as the definition of the matrixes which will be employed in the Kalman Filter.

3 MODELING

3.1 Prediction Phase: Odometer Model of the Robots Movement

A classic method used to calculate the pose of a robot is the odometry. This method uses sensors, optical encoders, for example, which measure the rotation of the robot's wheels. Using the cinematic model of the robot, its pose is calculated by means of the integration of its movements from a referential axis.

As encoders are sensors, normally their reading would be implemented in the actualization phase of the Kalman Filter, not in the prediction phase. Thrun et al. propose that odometer information does not function as sensorial measurements; rather they suggest incorporating them to the robot's model (Thrun et al., 2005).

In order that this proposal is implemented, one must use a robot's cinematic model considering the angular displacements of the wheels as signal that the system is entering in the prediction phase of the Kalman Filter.

Consider a robot with diferential drive in which the control signals applied and its actuators are not tension, instead angular displacement, according to Figure 1.

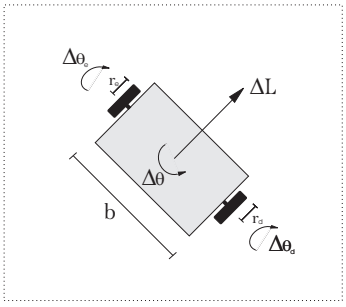


Figure 1: Variables of the geometric model.

With this idea, and supposing that speeds are constant in the sampling period, one can determine the geometric model of the robot's movement(system 9).

$$\begin{cases} x_t = x_{t-1} + \\ \quad \dfrac{\Delta L}{\Delta\theta}[\sin(\theta_{t-1}+\Delta\theta) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} - \\ \quad \dfrac{\Delta L}{\Delta\theta}[\cos(\theta_{t-1}+\Delta\theta) - \cos(\theta_{t-1})] \\ \theta_t = \theta_{t-1} + \Delta\theta \end{cases} \tag{9}$$

The turn easier the readability of the system (9) representing the odometry model of the robot, two auxiliary variables have been employed ΔL and $\Delta\theta$

$$\begin{cases} \Delta L = (\Delta\theta_d r_d + \Delta\theta_e r_e)/2 \\ \Delta\theta = (\Delta\theta_d r_d - \Delta\theta_e r_e)/b \end{cases} \tag{10}$$

in which $\Delta\theta_d$ is the reading of the right encoder and functions relatively the robot by means of the angular displacement of the right wheel; $\Delta\theta_e$ is the reading of the left encoder and functions as a displacement applied to the left wheel; b represents the distance from wheel to wheel of the robot; r_d and r_e are the spokes of the right and the left wheels respectively.

It is important to emphasize that in real applications the angular displacement effectively realized by the right wheel differs of that measured by the encoder. Besides that, the supposition that the speeds are constant in the sampling period, which has been used to obtain the model 9, is not always true. Hence, there are differences between the " angular displacements of the wheels ($\Delta\hat{\theta}_d$ e $\Delta\hat{\theta}_e$) and those ones measured by the encoders ($\Delta\theta_d$ e $\Delta\theta_e$). This difference will be modeled by a Gaussian noise, according to system (11).

$$\begin{cases} \Delta\hat{\theta}_d = \Delta\theta_d + \varepsilon_d \\ \Delta\hat{\theta}_e = \Delta\theta_e + \varepsilon_e \end{cases} \tag{11}$$

It is known that odometry possesses accumulative error. Therefore, the noises ε_d and ε_e do not possess constant variance. It is presumed that these noises present a proportional standard deviation to the module of the measured displacement.

With these new considerations, system (9) is now represented by system (12):

$$\begin{cases} x_t = x_{t-1} + \\ \quad \dfrac{\Delta\hat{L}}{\Delta\hat{\theta}}[\sin(\theta_{t-1}+\Delta\hat{\theta}) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} - \\ \quad \dfrac{\Delta\hat{L}}{\Delta\hat{\theta}}[\cos(\theta_{t-1}+\Delta\hat{\theta}) - \cos(\theta_{t-1})] \\ \theta_t = \theta_{t-1} + \Delta\hat{\theta} \end{cases} \tag{12}$$

in which

$$\begin{cases} \Delta\hat{L} = (\Delta\hat{\theta}_d r_d + \Delta\hat{\theta}_e r_e)/2 \\ \Delta\hat{\theta} = (\Delta\hat{\theta}_d r_d - \Delta\hat{\theta}_e r_e)/b \end{cases} \tag{13}$$

One should observe that this model can not be used when $\Delta\hat{\theta} = 0$. When it occurs, one uses an odometry module simpler than a robot (system 14), obtained from the limit of system 12 when $\Delta\hat{\theta} \to 0$.

$$\begin{cases} x_t = x_{t-1} + \Delta\hat{L}\cos(\theta_{t-1}) \\ y_t = y_{t-1} + \Delta\hat{L}\sin(\theta_{t-1}) \\ \theta_t = \theta_{t-1} \end{cases} \tag{14}$$

Thrun's idea implies a difference as what concerns system (4), because the noise is not audible; rather, it is incorporated to the function which describes the model, as system (15) shows:

$$\begin{cases} \mathbf{s}_t = p(\mathbf{u}_{t-1}, \mathbf{s}_{t-1}, \varepsilon_t) \\ \mathbf{z}_t = h(\mathbf{s}_t) + \delta_t \end{cases} \tag{15}$$

in which $\varepsilon_t = [\varepsilon_d \ \varepsilon_e]^T$ is the noise vector connected to odometry.

It is necessary, however, to bring about a change in the prediction phase of the system (6) resulting in the system (16) equations:

$$\begin{cases} \bar{\mu}_t = \mu_{t-1} + p(\mathbf{u}_{t-1}, \mu_{t-1}, 0) \\ \bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{V}_t \mathbf{M}_t \mathbf{V}_t^T \end{cases} \tag{16}$$

in which $\mathbf{M}$, $l \times l$, is the co-variance matrix of the noise sensors (ε) and $\mathbf{V}$, $n \times m$, is the jacobian mapping the sensor noise to the space of state. Matrix V is defined by the equation (17).

$$\mathbf{V}_t = \frac{\partial p(\mathbf{u}_{t-1}, \mathbf{s}_{t-1}, 0)}{\partial \mathbf{u}_{t-1}} \tag{17}$$

Making use of the odometry model of the robot described in this section and the definitions of the matrixes used by the Kalman Filter, we have:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & g_{13} \\ 0 & 1 & g_{23} \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{onde} \tag{18}$$

(19)

$$g_{13} = \frac{\Delta\hat{L}}{\Delta\hat{\theta}}[\cos(\theta_{t-1} + \Delta\hat{\theta}) - \cos(\theta_{t-1})]$$
$$g_{23} = \frac{\Delta\hat{L}}{\Delta\hat{\theta}}[\sin(\theta_{t-1} + \Delta\hat{\theta}) - \sin(\theta_{t-1})]$$

$$\mathbf{V} = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ r_d/b & -r_e/b \end{pmatrix}, \quad \text{onde} \tag{20}$$

(21)

$$v_{11} = k1\cos(k2) - k3[\sin(k2) - sin(\theta_{t-1})]$$
$$v_{12} = -k1\cos(k2) + k3[\sin(k2) - sin(\theta_{t-1})]$$
$$v_{21} = k1\sin(k2) - k3[-\cos(k2) + cos(\theta_{t-1})]$$
$$v_{22} = -k1\sin(k2) + k3[-\cos(k2) + cos(\theta_{t-1})]$$

$$\mathbf{M} = \begin{pmatrix} (\alpha_1|\Delta\hat{\theta}_d|)^2 & 0 \\ 0 & (\alpha_2|\Delta\hat{\theta}_e|)^2 \end{pmatrix} \tag{22}$$

Elements m_{11} and m_{22} in the equation (22) represent the fact that the standard deviations of ε_d and ε_e are proportional to the module of the angular displacement. The variables $k1$, $k2$ and $k3$ are given by system (23), considering $r_d = r_e = r$.

$$\begin{cases} k1 = \dfrac{r(\Delta\hat{\theta}_d + \Delta\hat{\theta}_e)}{b(\Delta\hat{\theta}_d - \Delta\hat{\theta}_e)} \\ k2 = \theta_{t-1} + \dfrac{r(\Delta\hat{\theta}_d - \Delta\hat{\theta}_e)}{b} \\ k3 = \dfrac{b\Delta\hat{\theta}_e}{2(r(\Delta\hat{\theta}_d - \Delta\hat{\theta}_e)/b)^2} \end{cases} \tag{23}$$

3.2 Update Phase: Sensor Model for Detecting Natural Landmarks

In this work we will use as natural landmarks a set of straight lines formed by the grooves of the floor in the environment where the robot will navigate because, besides being already available in the referred environment, they are also very common in the real world.

Due to the choice of the straight lines as landmarks, the technique adopted to identify them was the Hough transform. This kind of transform is a method employed to identify inside a digital image a class of geometric forms which can be represented by a parametric curve (Gonzales, 2000). As what concerns the straight lines, a mapping is provided between the Cartesian space (X,Y) and the space of the parameters (ρ,α) where the straight line is defined.

Hough defines a straight line using its common representation, as equation (24) shows, in which parameter ρ represents the length of the vector and α the angle this vector forms with axis X. Figure 2 shows the geometric representation of these parameters.

$$\rho = x\cos(\alpha) + y\sin(\alpha) \tag{24}$$

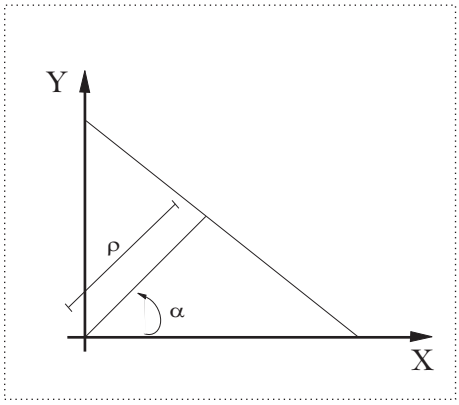


Figure 2: Parameters of the Hough.

The system discussed in this paper is based in a robot with differential drive, possessing a rm and stable camera embedded in its structure, as in Figure 3. The idea is to use information obtained directly on the image processing (ρ,α) in the actualization phase of an Extended Kalman Filter to calculate the robot's pose. Thereof, one must deduct the sensor model (that is, the image processor) in function of the variables of state.

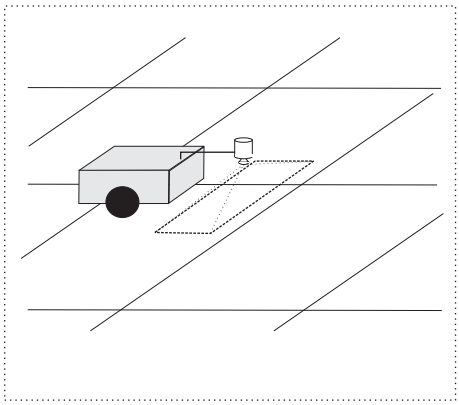


Figure 3: Robotic system.

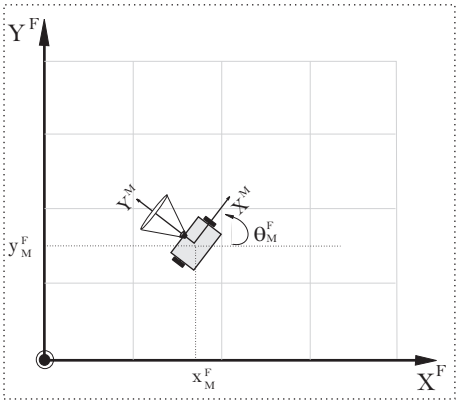


Figure 4: System of Coordinates.

Robot navigates an environment in which the position of the straight lines of the world (α^F, ρ^F) is known and, at each step, it identifies the descriptors of the straight lines contained in the image (α^M, ρ^M) using image processing and the parameters of the camera gauging.

Figure 4 illustrates the systems of steady $\{F\}$ and mobile $\{M\}$ coordination used to mathematic deduction of the sensor model. The point (x_M^F, y_M^F) is the coordinate of origin of the mobile system mapped in the system of steady and mobile coordinates, while variable θ_M^F represents the rotation angle of the system of mobile coordinates.

Our point of departure is a simple change mapping a point in the system de mobile $\{M\}$ coordinates for the system of steady $\{F\}$ coordinates, as in system (25).

$$\begin{cases} x^F = \cos(\theta_F^M)x^M - \sin(\theta_F^M)y^M + x_M^F \\ y^F = \sin(\theta_F^M)x^M + \cos(\theta_F^M)y^M + y_M^F \end{cases} \quad (25)$$

Using an equation (24) and considering the system of steady $\{F\}$ coordinates, we have:

$$\rho^F = x^F\cos(\alpha^F) + y^F\sin(\alpha^F) \quad (26)$$

Also using the definition of system (24), but now considering the system of mobile $\{M\}$ coordinates, we have:

$$\rho^M = x^M\cos(\alpha^M) + y^M\sin(\alpha^M) \quad (27)$$

Replacing (25) in (26) and doing the necessary equivalences with system (27), we can obtain system (28), which represents the sensor module to be used in the filter.

$$\begin{cases} \alpha^M = \alpha^F - \theta_M^F \\ \rho^M = \rho^F - x_M^F\cos(\alpha^F) - y_M^F\sin(\alpha^F) \end{cases} \quad (28)$$

In this system, α^F and ρ^F are given, because they represent the description of the map landmark, which is supposedly known. The equations express the relations among the returned information (α^F, ρ^F) and the height that one wants to estimate $(x_F^M, y_F^M, \theta_F^M)$.

One should note that there is a straight relation among these variables $(x_F^M, y_F^M, \theta_F^M)$ and the robot's pose (x_R, y_R, θ_R), which is given by system 29.

$$\begin{cases} x_R = x_F^M \\ y_R = y_F^M \\ \theta_R = \theta_F^M + \dfrac{\pi}{2} \end{cases} \quad (29)$$

The model of system 28 is incorporated to the Kalman Filter through matrix H (equation 8), given by equation 30.

$$H = \begin{pmatrix} -\cos(\theta_M^F) & -\sin(\theta_M^F) & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (30)$$

4 RESULTS

The situations presented here have been obtained by simulation. We tried to use the noise measure of the sensors consistent to reality. For that, it has been embedded to encoders a noise which standard deviation is proportional to the amount of read pulses. In the identification of the parameters of the straight lines ρ and α, the standard deviation of noise also obeys a proportion which is ruled by the size that the straight line occupies in the image.

In the Figures, the hatched rectangle represents the robot's real pose, while the continuous rectangle, the calculated pose.

Figure 5 presents the result of the localization system us- ing only odometry.

Another localization system largely used has also been implemented: the localization system using geometric correction. In this system, at each step the straight lines are identified and used to calculate the robot's pose using trigonometry. When there are no straight lines, the correction is made by odometry(Figure 6).

Finally, in Figure 7 is shown the result of the pose calculation using the fusion of the data of odometry and of the landmark detection by EKF.

A particular situation has been implemented to test the robustness of the localization systems. For that, it has been introduced to the system a perturbation whenever the robot gets near position (6.5,4.5). The results coming from the use of geometry and Kalman are exhibited in Figures 8 and 9.

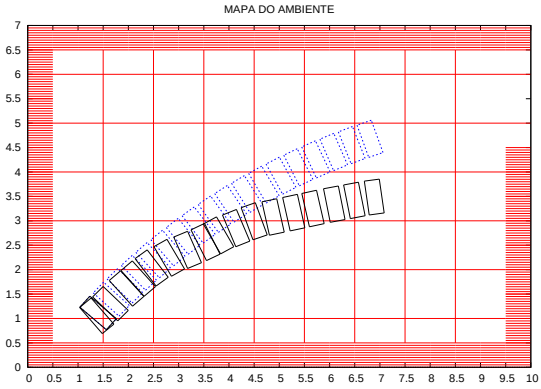


Figure 5: Localization by odometry.

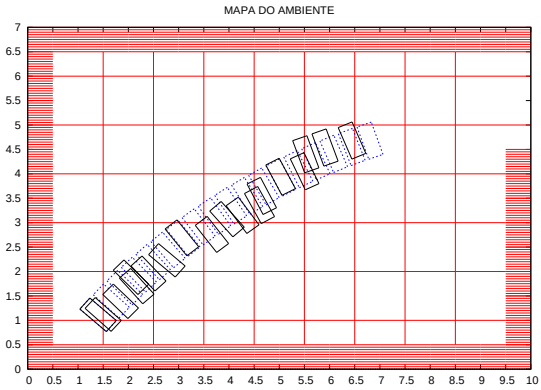


Figure 6: Localization by odometry and geometric correction.

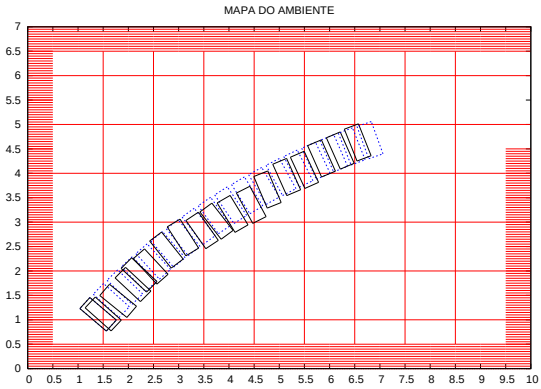


Figure 7: Localization using Extended Kalman Filter.

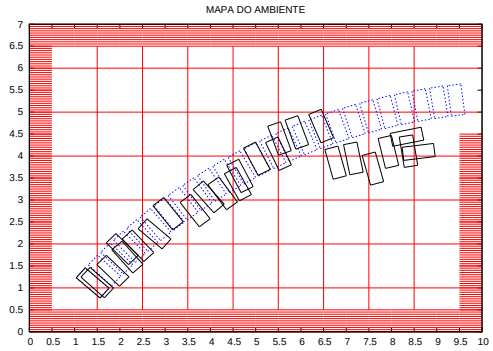


Figure 8: Effect of perturbation: Localization by odometry and geometric correction.

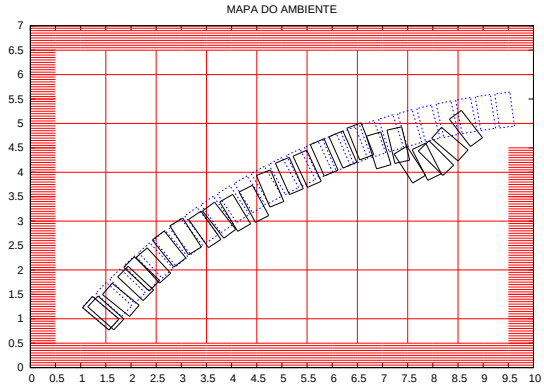


Figure 9: Effect of perturbation:Localization using Extended Kalman Filter.

5 CONCLUSIONS AND PERSPECTIVES

This paper has proposed a localization system for mobile robots using the Extended Kalman Filter. The main contribution is the modeling of the optic sensor made such a way that it permits using the parameters obtained in the image processing directly to equations of the Kalman Filter, without passing any intermediate phase of pose calculation, or of distance, only with the available usual information.

When analyzing Figures 5, 6 and 7 one can perceive that the behavior of the localization system using the Kalman Filter has proved more satisfactory than those using odometry and geometric corrections.

As what concerns perturbation rejections (Figures 8 e 9) the system based in Kalman also proved efficient, for it tends to return to real pose, while the system based in geometric correction did not exhibit the same performance.

As future works, we intend to: Implement other formulations of the Kalman Filter. For example, the Kalman Filter with Partial Observations; Replace the Kalman Filter by a Filter of Particles, having in view that the latter incorporates more easily the non-linearities of the problem, besides leading with non-Gaussian noises; Develop this strategy of localization to a proposal of SLAM (Simultaneous Localization and Mapping), so much that robot is able of doing its localization without a previous knowledge of the map

and, simultaneously, mapping the environment it navigates.

ACKNOWLEDGEMENTS

We thanks CAPES and CNPq by the financial support.

REFERENCES

Bezerra, C. G. (2004). Localizao de um rob mvel usando odometria e marcos naturais. Master's thesis, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN.

Borenstein, J., Everett, H., Feng, L., and Wehe, D. (1997). Mobile robot positioning: Sensors and techniques. Journal of Robotic Systems, 14(4):231–249.

Gonzales, R. C. (2000). *Processamento de Imagens Digitais*. Edgard Blucher.

Kim, S. and Kim, Y. (2004). Robot localization using ultrasonic sensors. Proccedings of the 2004 IEEE/RSJ International Conference on Inteligent Robots and Systems, Sendal, Japan.

Kiriy, E. and Buehler, M. (2002). Three-state extended kalman filter for mobile robot localization. Report Centre for Intelligent Machines - CIM,McGill University.

Launay, F., Ohya, A., and Yuta, S. (2002). A corridors lights based navigation system including path definition using a topologically corrected map for indor mobile robots. Proceedings IEEE International Conference on Robotics and Automation, pp.3918-3923.

Lizzaralde, F., Nunes, E., Hsu, L., and J.T., W. (2003). Mobile robot navigation using sensor fusion. Proccedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan.

Park, K. C., Chung, D., Chung, H., and Lee, J. G. (1998). Dead reckoning navigation mobile robot using an indirect kalman filter. Conference on Multi-sensor fusion and Integration for Intelliget Systems, 9(3):107-118.

Pres, J., Catellanos, J., Montiel, J., Neira, J., and Tards, J. (1999). Continuous mobile robot localization: Vision vs. laser. Proccedings of the 1999 IEEE International Conference on Robotic and Automation.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, 01 edition.

MODELING AND SIMULATION OF A NEW PARALLEL ROBOT USED IN MINIMALLY INVASIVE SURGERY

Doina Pisla, Calin Vaida, Nicolae Plitea
Technical University of Cluj-Napoca, Department of Mechanics and Computer Programming
Constantin Daicoviciu 15, RO-400020 Cluj-Napoca, Romania
Doina.Pisla@mep.utcluj.ro, Calin.Vaida@mep.utcluj.ro, Nicolae.Plitea@mep.utcluj.ro

Jürgen Hesselbach, Annika Raatz, Marc Simnofske, Arne Burisch
Technical University of Braunschweig, Institute of Machine Tools and Production Technology
19B Langer Kamp, D-38106 Braunschweig, Germany
J.Hesselbach@tu-bs.de, A.Raatz@tu-bs.de, M.Simnofske@tu-bs.de, A.Burisch@tu-bs.de

Liviu Vlad
University of Medicine and Pharmacy "Iuliu Hatieganu", Surgery Clinic III
Emil Isac 13, 400023 Cluj-Napoca, Romania
lvlad@cluj.astral.ro

Keywords: Parallel robots, Minimally invasive surgery, Modeling, Simulation, Kinematics, Workspace.

Abstract: Surgery is one of the fields where robots have been introduced due to their positioning accuracy which exceed the human capabilities. Parallel robots offer higher stiffness and smaller mobile mass than serial ones, thus allowing faster and more precise manipulations that fit medical applications. In the paper is presented the modeling and simulation of a new parallel robot, based on an innovative structure, used in minimally invasive surgery. The parallel architecture has been chosen for its superiority in precision, repeatability, stiffness, higher speeds and occupied volume. The robot kinematics, singular position identification and workspace generation are illustrated. Using the developed virtual model of the parallel robot, some simulation tests are presented.

1 INTRODUCTION

Simulation is one of the essential aspects in the robots research field. The industrial trend is to develop simulation systems, which could be helpful for the robot off-line control and for the robot operational workspace generation. Due to the high graphical possibilities and memory capacities, the personal computers offer already the necessary instruments to achieve efficient simulation systems. Within the simulation system, an important role plays the achievement of a virtual robot functional model, which allows its interactive visualization and functional simulation.

The first important application of the simulation systems is during the design process of a new robot structure or a new flexible cell. In both cases it is possible to test the functionality and the restrictions influence without the actual building of the prototype. In this way the costs, the implementation and testing time are considerable reduced.

The second kind of important application of the simulation systems refers to the existent robots that could be better programmed, analyzed and optimized if the research activity is considered.

Within the simulation systems, the visualization of the robot trajectory with collision detection is achieved, offering a safer system especially for the case in which several robots are coupled, situation that requires a more rigorous planning.

The robotized system performance can be tested without any danger for the man or for the robot. The programs generated within the simulation system could be optimized before their practical implementation leading to any damage avoidance due to programming facilities, cost reduction and quality improvement.

Another advantage of the simulation systems consist in their use in the research field. New control algorithms can be created and tested before investments in the necessary hardware devices for the robot control to be made.

To the parallel robots simulation systems are attributed different meanings: numerical simulation for a specific robot function or element in transition; parallel computing for the robot state and transitions definition; the graphical simulation of the robot structure and motion. In some cases the animation is considered as simulation.

Geometric modeling is one of the keystones of CAD systems. It uses mathematical descriptions of geometric elements to facilitate the representation and manipulation of graphical images on a computer display screen. In three-dimensional representations, there are four types of modeling approaches, (Foley et al., 1990) and (Pisla, 2001): Wire frame modelling; Surface modeling; Solid modeling; Hybrid solid modeling.

Simplifying the virtual model enables researchers to apply for the use of different methods (usually starting with the most recommended one) in correspondence with the available time and existing endowment in order to compare the obtained results and refine only the most adequate solution. In practice, the application of simulation techniques is used in two situations: for already made structures to improve their characteristics; for new designed structures to forecast their behaviour before construction.

Lately is considered more the second case, due to the intense development in creating new structures with better dynamics and larger cinematic facilities, or considering the parallel kinematic, a multibody system can be designed to be used like a machine tool or like a robot, therefore the same kinematical structure must respond to different tasks that can be demonstrated with the simulation techniques.

One of the first achieved models in order to achieve the robot simulation is the geometric model. There are many possibilities to create this model also in this case. In respect with the chosen model, different aspects regarding the mechanism properties or actuating methods are emphasized.

The modeling process is oriented to the correlation between the method and the structure in order to point out the influence of the modeling instruments on the structure design.

Considering that the robot performances depend on the adequate combination between the designed structure, the actuating and the control system, the applying of suitable modeling and simulation techniques could improve the robot design and control through: identification of standardized element behaviour with modular dimensions; identification and visualization of the operational workspace, which is harder to be achieved in the parallel robot case; identification of different parallel robot singularities and their avoiding; trajectory visualization which must be programmed for certain parallel robot tasks; speeds and acceleration visualization in different trajectory points.

Regarding the application of robots for medical applications there are some investigators focused on exploring the capabilities of parallel robots in this field (Ben-Porat, 2000), (Glozman, 2001), (Grace, 1993). AESOP robotic arm, used to guide a tiny camera inside the body, was the first robotic system used in surgery dating from 1993 (Brown University, 2005). It was produced by Computer Motion, which developed several such versions of AESOP until they created Zeus^{TM} Robotic Surgical System with three robotic arms attached on the side of the operating table. A competitor of Computer Motion, Intuitive Surgical, designed another revolutionary equipment, da Vinci^{TM} Surgical System, which became the market competitor of Zeus until 2003 when the two companies merged (Brown University, 2005).

Most of the robots, which assist the surgeons, are serial robots (Brown University, 2005). The serial module generates a large workspace while the parallel module is steadier and offers a high accuracy during the surgical operation. In this case, there are used force control algorithms in order to ensure the safety behavior and the accepted accuracy.

Parallel robots offer a higher stiffness and smaller mobile mass than serial robots, thus they allow faster and more precise manipulations. The problems concerning parallel structures kinematics are usually more complicated than for the serial structures. The drawbacks of serial robots, determined by their structural construction, motivate the research in the field of robotic assisted surgery for a continuous search of task oriented robot architectures that best fit a specific group of medical applications.

The paper is organized as follows: Section 2 is dedicated to modeling techniques; Section 3 deals with kinematic modeling of the new parallel structure designated to minimally invasive surgery; Section 4 presents its workspace modeling, Section 5 presents the developed simulation program for parallel robots with some simulation tests on a studied parallel robot. The conclusions of this work are presented in Section 6.

2 MODELING TECHNIQUES

Bottom-up Modeling Assembly Approach is the modeling technique where component parts are designed and edited in isolation of their usage within some higher level assembly (Pisla, 2004). All assemblies using the component are automatically updated when opened to reflect the geometric edits made at the piece part level. Using the bottom-up approach, characteristic of traditional engineering design practices, designs are built from known components (embodiments) in anticipation of satisfying functional requirements. The bottom-up approach is known to be highly iterative. In the case of large or complex designs, the problem quickly leads to combinatorial explosion, inefficiency in the design process, and difficulties assessing the effectiveness and merits of design alternatives.

Bottom-up design is the traditional method. In bottom-up design, you create parts, insert them into an assembly, and mate them as required by your design. Bottom-up design is the preferred technique when you are using previously constructed, off-the-shelf parts. An advantage of bottom-up design is that because components are designed independently, their relationships and regeneration behavior are simpler than in top-down design. Working bottom-up allows you to focus on the individual parts.

Top-Down Modeling Assembly approach is the modeling technique where component parts can be created and edited while working at the assembly level (Pisla, 2004). Geometric changes made at the assembly level are automatically reflected in the individual component part immediately. In the top-down approach, characteristic of a systems engineering process, design is driven from functional requirements toward solution alternatives. While design solutions using this approach are likely to meet functional requirements, there is no guarantee that solutions are realizable in terms of physical manifestations.

Top-down design is different because you start your work in the assembly. You can use the geometry of one part to help define other parts, or to create machined features that are added only after the parts are assembled. You can start with a layout sketch, define fixed part locations, planes, and so on, then design the parts referencing these definitions. For example, you can insert a part in an assembly, then build a fixture based on this part. Working top-down allows you to reference model geometry, so you can control the dimensions of the fixture by creating geometric relations to the original part. That way, if you change a dimension of the part, the fixture is updated.

Analyzing both techniques characteristics, the bottom-up assembly approach has been chosen for parallel robot modeling due to their advantages.

3 KINEMATIC MODELING OF THE PARAMIS ROBOT

In Figure 1 is presented the kinematic scheme of the new parallel structure – PARAMIS, which can be used for surgical instruments positioning (Plitea, 2007). The structure has three degrees of freedom in space and it consists of three active joints (two of them are prismatic and one is rotational). The passive joints are two cylindrical joints, one prismatic joint and one Cardan joint. The robot must position the end of the laparoscope, namely the video camera in any point of the operational field.

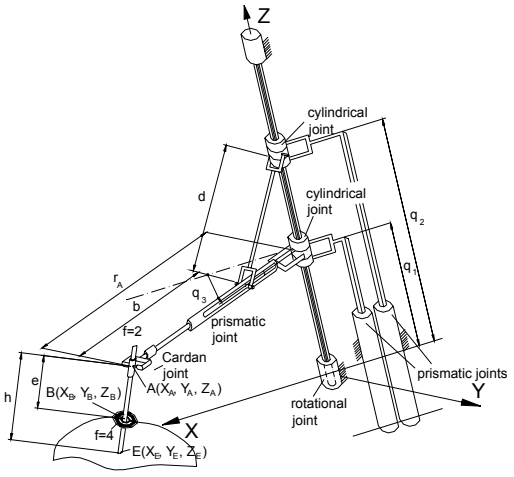


Figure 1: Kinematic scheme of the new parallel robot.

The particularity of this motion is the fact that the endoscope will move around a fixed point in space, which is the entrance point of the trocar in the abdominal wall of the patient (namely the point B). The presence of the passive Cardan joint before the laparoscope will allow the motion around the fixed point of the abdominal wall. The trocar, the abdominal entrance point and the endoscope can be seen as a virtual joint with four degrees of freedom, which allow the camera the rotation around the three axis and a translation parallel with the axis of the trocar. Thus, the endoscope can be positioned in any point of the abdominal area using the three degrees of freedom of the robot. The geometrical parameters of the PARAMIS robot are given by b, d, h; X_B, Y_B, Z_B (figure 1). In (Plitea, 2007) the inverse and direct geometric and kinematic

models for speeds and accelerations of the structure have been already presented.

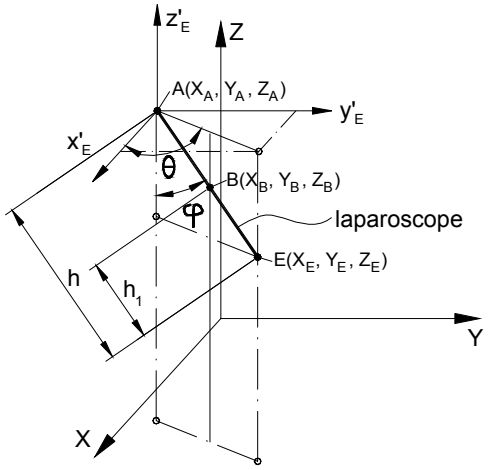


Figure 2: The angles of the passive Cardan joint: φ, θ.

The implicit functions equation system, which has been obtained from the kinematic model, are:

$$\begin{cases} f_1(X_E, q_1, q_2, q_3) \equiv X_A - \left[b + \sqrt{d^2 - (q_2 - q_1)^2} \right] \cdot \cos q_3 = 0 \\ f_2(Y_E, q_1, q_2, q_3) \equiv Y_A - \left[b + \sqrt{d^2 - (q_2 - q_1)^2} \right] \cdot \sin q_3 = 0 \\ f_3(Z_E, q_1) \equiv Z_A - q_1 = 0 \end{cases} \quad (1)$$

With the notations:

$$\dot{X} = \left[X_E, Y_E, Z_E \right]^T, \quad \dot{q} = \left[q_1, q_2, q_3 \right]^T \quad (2)$$

Through derivation of the relations (1), it yields:

$$A \cdot \dot{X} + B \cdot \dot{q} = 0 \quad (3)$$

where A is the Jacobian matrix of the direct kinematics and B is the Jacobian matrix of the inverse kinematics, x the vector of end-effecter coordinates and q the vector of generalized coordinates of the actuators.

The form of the A and B matrices are:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial X_E} & \frac{\partial f_1}{\partial Y_E} & \frac{\partial f_1}{\partial Z_E} \\ \frac{\partial f_2}{\partial X_E} & \frac{\partial f_2}{\partial Y_E} & \frac{\partial f_2}{\partial Z_E} \\ \frac{\partial f_3}{\partial X_E} & \frac{\partial f_3}{\partial Y_E} & \frac{\partial f_3}{\partial Z_E} \end{bmatrix} \quad (4)$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial q_3} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_2}{\partial q_3} \\ \frac{\partial f_3}{\partial q_1} & \frac{\partial f_3}{\partial q_2} & \frac{\partial f_3}{\partial q_3} \end{bmatrix} \quad (5)$$

3.1 Singularities Analysis

An important problem for the parallel robots is to identify and avoid singularities, ensuring the system stability and kinematic accuracy. Physically, these singular positions lead to an instantaneous change in the system number of degree of freedom and hence loss of controllability and degradation of the natural stiffness. Therefore it is important to identify singular configurations at the design stage in order to improve the system performance. For an optimal robot control these positions should be identified and avoided from the robot workspace.

A number of papers have studied the singularity problem for close kinematic chains (Merlet, 2005), (Sefrioui, 1992), (Park, 1999), (Romdahne, 2002).

The used algorithm for the PARAMIS structure singularity analysis is based on the deriving the determinants for the two Jacobian matrices derived from the inverse and the direct geometric model.

Singularities of Type I. First type of singularities occurs when the determinant of the Jacobian matrix B is zero.

Using (1), it yields:

$$\det(B) = \frac{q_2 - q_1}{\sqrt{d^2 - (q_2 - q_1)^2}} \cdot \left[b + \sqrt{d^2 - (q_2 - q_1)^2} \right] \quad (6)$$

If det(B)=0, physically, the parallel robot looses one or more degrees of freedom.

Case 1. $q_1 = q_2$ - The mechanisms is locked. The situation corresponds to the positioning on the horizontal plane of the rod of length d (the first extreme position).

Solution. Constructively, the mechanical structure of the robot will impose $q_1 < q_2$

Case 2. $d^2 = (q_2 - q_1)^2 \equiv d = q_2 - q_1$ - The mechanisms is locked. This situation corresponds to the positioning on the vertical axis of the rod of length d (the second extreme position).

Solution. Constructively, the structure of the robot will not allow displacements so long as the rod of length d to reach a vertical position.

Case 3.

$b + \sqrt{d^2 - (q_2 - q_1)^2} = 0 \quad b > 0 \quad \sqrt{d^2 - (q_2 - q_1)^2} > 0$

is impossible (as they represent lengths of mechanical components).

Singularities of Type II. Second type of singularities appears when the determinant of the Jacobian matrix A is zero.

Using (1), it yields:

$$\det(A) = \left(1 - \frac{h}{h_1}\right)^2$$

If det(A)=0, the system gains one or more degrees of freedom.

Case 1. $h = h_1$ it yields a point where the robot cannot be controlled. In case of the point A will superpose over point B.

Solution. The attachment mechanism of surgical instruments on the robot and the shape of the trocar will not allow point A to superpose over point B. (In this situation the instrument or laparoscope will be situated entirely in the patient).

Case 2. $h_1 = 0$, correponds to the situation when the laparoscope is positioned completely out of the surgical field, with $E \equiv B$ situation which will be impossible to attain with the robot, as the insertion and removel of the laparoscope is done manually by the surgeon.

The **Third Type of Singularities** so called "architectural singularities" takes place when both Jacobian determinants are zero.

Superposing the cases when the two determinants are equal with zero, it can be easily seen that in none of the above situations the determinants cannot be null in the same time. In conclusion, from the constructive point of view, for the functioning conditions imposed, there are no singularity points of type III.

These types of singularities can be easily avoided in the design stage.

Based on the fact that the only singularity cases appear in areas where the positions that the robot cannot attain it can be concluded that, in the working space of the robot, there are no points where the robot cannot be controlled.

4 WORKSPACE MODELING

For the geometrical determination of the working space of point E (the laparoscope extremity where the camera is placed) the first step is the determination of the working space of point A and its projection, keeping in mind that for on the segment ABE the point B is fixed in space and AE has a constant length.

The working space of point A can be easily determined, by determining the cross-sections within the working space in a parallel plane with OXY which for the limits imposed for q_1 and q_2 will have the same shape for any position of the Z axis.

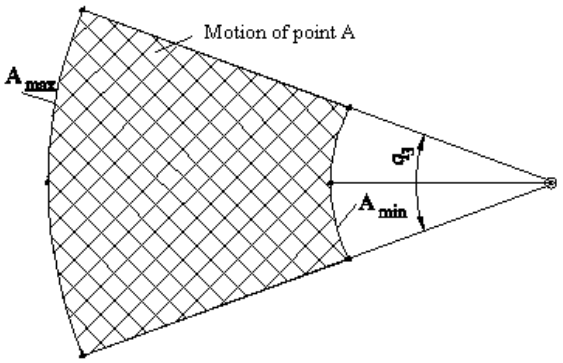


Figure 3: Motion of point A in a parallel plane with OXY (horizontal).

The hatched area in Figure 3, represents all the possible coordinates that point A can attain in a horizontal plane. Generating the three-dimensional working space it results the shape presented in Figure 4.

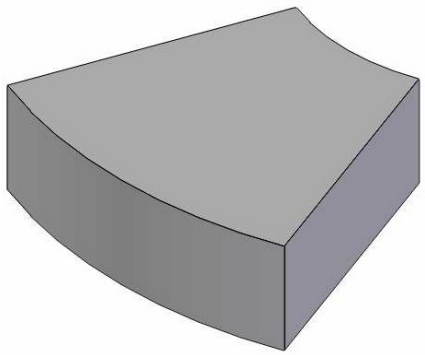


Figure 4: The workspace of point A.

For the workspace generation for point E has to be considered that the laparoscope (or any other surgical instrument) has a constant length and will always pass through a fixed point in space. The working space of point E by generating two intermediate working spaces whose intersection will represent the effective working space. The first intermediate working space is generated by projecting the point A from sections parallel with OXY plane. The second one is generated from sections of the working space of point A perpendicular on the plane OXY.

The intersection of the two intermediary volumes will result in the working space of point E presented in Figure 5.

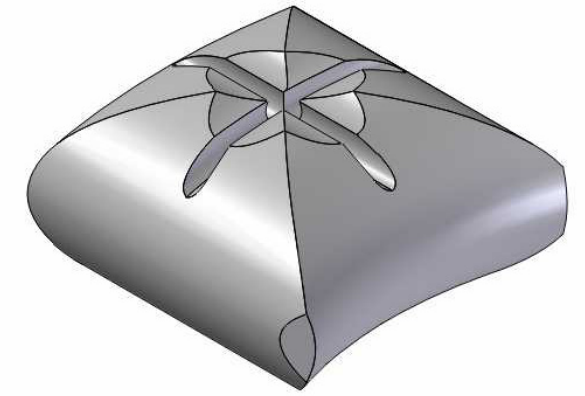


Figure 5: The workspace of point E.

5 SIMULATION PROGRAM FOR PARAMIS PARALLEL ROBOT

A complex simulation program was developed in order to study the geometric, kinematic and dynamic characteristics of parallel robots with different degrees of freedom (Pisla, 2001), (Pisla, 2005).

This simulation system consists of the following modules:

-the geometrical module (GM) defines the robot geometric structure;

-the kinematic module (KM) defines the parallel kinematic structure. The KM consists of: the algorithms for the kinematics, the algorithms for workspace generation and for singularities identification;

-the modeler (MM) represents the combination of the geometrical representations, kinematic descriptions and the adequate disposition of different objects;

-the simulator (SM) has the main task to compute the current state of the parallel robot with respect to certain instructions such as: the robot source program or signals which are generated by sensors;

-the graphical module (GPM) is responsible with the simulated robot transformations and screen graphical visualization.

The simulator is interactively achieved such as the user could influence the simulation parameters and could be informed about the possible errors during the simulation process.

The simulator determinates the next robot state, independent of the input mode. If an error appears, the user is informed through an error message with its type and elimination way. The simulator is interactively achieved such as the user could influence the simulation parameters and could be informed about the possible errors during the simulation process.

The data exchange between simulation system modules is achieved through data files, which allow future module integration in the existing system and makes easy the data exchange with another off-line programming systems.

For the graphical modeling of the parallel structures it was used Solid Edge©, one of the most advanced software for computer aided design, available on the market (www.solidedge.com). This program was selected especially for its outstanding performances in terms of stability and user-friendly interface, and even more for its total compatibility with Visual Basic.Within the simulation system the virtual graphical model of parallel structures has been created.

The geometric parameters of the parallel structure can be modified within the 3D modeling software (Solid Edge Assembly) influencing the simulation environment. The assembly relations between parts, between subassemblies or between parts and subassemblies can be also modified (Pisla, 2001), (Pisla, 2004).

The program has the possibility to select the parallel structure from a list in accordance with the class of the mechanism and its degree of freedom.

The parallel structure parameterization enables the possibility to develop a study regarding the geometric optimization and the robot workspace shape.

The results obtained are useful for the designers not only to understand the distribution of characteristics of the workspaces for various geometrical parameters of parallel structures, but also to optimize the parallel robots.

The presented simulation system enables the motion visualization, it is valid for any structure of parallel robot enabling to introduce the kinematic model over the virtual robot the single foreseen limit being the hardware computing capacity. The introduction of extra conditions related to any element, joint or overall behaviour of the robot is possible with a small number of actions.

These facilities of the simulation software enable the possibility to develop a complex study about the kinematics and dynamics in order to optimize the parallel structure.

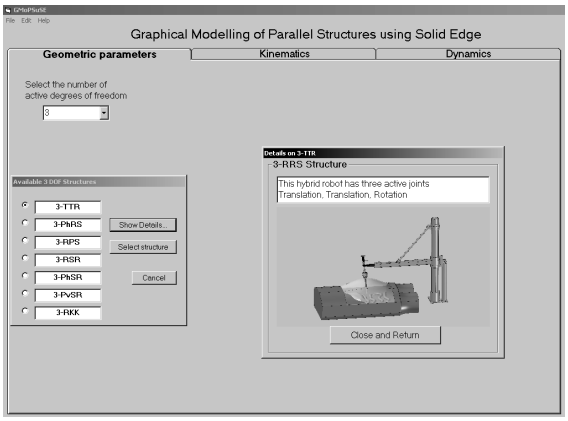


Figure 6: Simulation of the PARAMIS parallel structure.

In Figure 6 is presented the graphical interface for graphical modeling for different parallel structures. The kinematic algorithms have been implemented in the kinematical module (Kinematics). In the simulation is included the parallel robot with the instrument and the virtual human body.

Within the simulation system (Figure 6) the virtual graphical model was created. The 3D functional model of the parallel structure allows the designer to understand better its form and its functionality. Using the Solid Edge Assembly module it is possible to impose the assembling conditions between the components including the mounting possibilities and the tolerance for and between the parts.

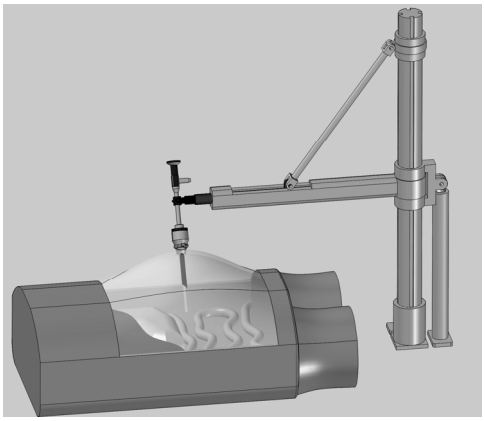


Figure 7: The PARAMIS robotic system.

The simulations were performed using the parallel robot presented in figure 7 which manipulates the laparoscope in the virtual human model to ensure also the fixed point in the working space of the robot (point B from Figure 1).

The virtual model was used for the validation of the geometrical model. For that, the robot was placed in an arbitrary position and all its parameters measured, including the point B (Figure 1). The second step was the computation of the coordinates of the point E in parallel with their measurement on the virtual model (Figure 8).

The calculated results based on the equations matched 100% with the ones given by the virtual model.

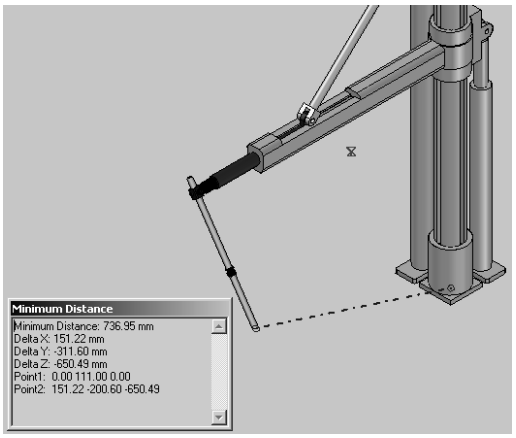


Figure 8: Validation of the geometric model.

The second goal concerning the virtual model was its use for the simulation of the robotic structure in the medical environment created. In this stage it was important to achieve proper motions for different commands in the active joints. Solid Edge allows a the definition of motion trajectories in terms of displacements, speeds and accelerations as points, lines or equations implemented in the active joints, as shown in figure 9.

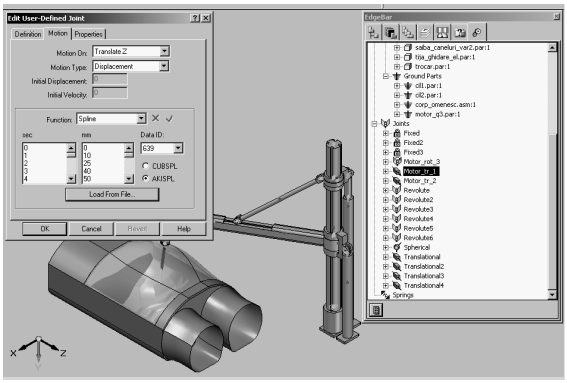


Figure 9: Input of motion trajectory in a module.

The simulation enabled the validation of the structure from the points of view of both engineers and surgeons and opened the way for the next step in the PARAMIS robotic design evolution, the construction of an experimental model.

6 CONCLUSIONS

Modeling and simulation are important and essential stages in the engineering design and problem solving process because it allows to prevent the risks and to lower the costs that appear with the design, construction and testing stage of a new robot. Since parallel mechanisms have a complex structure, using the proposed approached of modeling and simulation will lead to an increased efficiency in developing new structures.

In the paper the modelling and simulation of the PARAMIS new parallel robot is presented. The simulations results obtained with the developed simulation system for parallel robots allow a structure verification before a parallel robot is built.

The developed simulation system for the parallel robots offer multiple advantages: modularity of the simulation system for parallel robots; friendly graphical user interface of the simulation system; possibility for parallel structure parameterization; possibility to generalize some parallel structures; Identification of singularities; possibility to identify the optimal working zone; workspace generation: the structures verification before a parallel robot is built.

In this case, the contribution is the use a virtual reality definition system to create the virtual objects (bodies) that includes shape, dimensions, mass and center of mass, surface characteristics and the deviations from these characteristics may be also considered. Between the virtual objects are defined the appropriate structure joints. The type of joint, their constructive dimensions, together with the actuating define the motion restrictions that are encapsulated within the subassemblies.

Over the predefined virtual structure is applied the kinematic model using a programming interface in Visual Basic ensuring the PTP or the path control motion.

The presented simulation system enables the motion visualization, enabling to introduce the kinematic model over the virtual robot the single foreseen limit being the hardware computing capacity. The introduction of extra conditions related to any element, joint or overall behavior of the robot is possible with a small number of actions. The up-to-date results validated the solution creating the premises for the next step, the construction of the experimental model for the PARAMIS robot.

ACKNOWLEDGEMENTS

This research was financed from the research grants awarded by the Romanian Ministry of Education and Research and the "Institutional Academic Cooperation Research Grant" between the Technical University of Braunschweig, Germany and the Technical University of Cluj-Napoca, Romania, awarded by Alexander von Humboldt foundation.

REFERENCES

Foley, van, D., Dam, A., Feiner, K.S., Hughes, F.J., 1990. *Computer Graphics Principles and Practice*. Addison-Wesley Publishing Company, 2nd edition.

Pisla, D., 2001. *Graphical Simulation of the Industrial Robots.* Todesco Publishing House.

Ben-Porat, O., Shoham, M., Meyer, J., 2000. *Control Design and Task Performance in Endoscopic Teleoperation, Presence.* Massachusetts Institute of Technology, 9, 3, 256-267.

Glozman, D., Shoham, M., Ficher, A., 2001. *A Surface-Matching Technique for Robot-Assisted Registration,* Computer Aided Surgery 6, 259-269.

Grace, K. W., Colgate, J. E., Glucksberg, M. R., Chun, J.H., 1993. A Six Degree of Freedom Micromanipulator for Ophthalmic Surgery. In *Proc. of IEEE International Conference on Robotics and Automation,* 630-635.

Brown University, Division of Biology and Medicine, 2005.

Pisla, D., Stan, S., New Approaches Regarding the Modelling and Simulation of Parallel Robots. In *Proc. of the 2nd International Conference on Robotics,* Robotica-2004, Intergraf Reşiţa Publishing House, 151-152.

Plitea, N., Hesselbach, J., Vaida, C., Raatz, A., Pisla, D., Budde, C., Vlad, L., Burisch, A., Senner, R., 2007. Innovative development of surgical parallel robots, Acta Electronica, Mediamira Science, Cluj-Napoca, 201-206.

Merlet, J-P., 2005. *Parallel Robots,* 2nd Edition, Kluwer, Dordrecht.

Sefrioui, J., C.M. Gosselin,, 1992. *Singularity Analysis and Representation of Planar Parallel Manipulators*, Robotics and Autonomous Systems 10, 209-224.

Park, F.C., Kim, J.W., 1999. *Singularity Analysis of Closed Kinematic Chains,* ASME Journal of Mechanical Design 121, 32-38.

Romdahne, L., Affi, Z., Fayet, M. 2002. *Design and Singularity Analysis of a 3–Translational-DOF In-Parallel Manipulator,* Journal of Mechanical Design, ASME 124, 419-426.

Pisla, D., Pisla, A., 2001. *Effiziente dynamische Rechnersimulation für Parallelroboter,* ZAMM, vol. 81, Suppl 5, 277-278.

Pisla, D., 2005. *Kinematic and dynamic modeling of parallel robots,* Dacia House, (published in Romanian), Cluj-Napoca.

REPRESENTATION OF ODOMETRY ERRORS ON OCCUPANCY GRIDS

Anderson A. S. Souza, Andre M. Santana, Ricardo S. Britto, Luiz M. G. Gonçalves and Aderlardo A. D. Medeiros

Depatment of Computating Engineering and Automation, UFRN, Natal, Brazil

{*abner, andremacedo, rbritto, lmarcos, adelardo*}*@dca.ufrn.br*

Keywords: Mapping, Occupancy grid, Odometry.

Abstract: In this work we propose an enhanced model for mapping from sonar sensors and odometry that allows a robot to represent an environment map in a more suitable way to both the sonar sensory data and odometry system of the robot. We use a stochastic modelling of the errors that brings up reliable information. As a contribution, we obtain a final map that is more coherent with the reality of the original data provided by the robotic system. Practical experiments show the results obtained with the proposed modification to be trustable in such a way that this map can be used to provide previous knowledge to the mobile robot in order to perform its tasks in an easier and accurate way. Moreover, the map can help the robot to support unexpected situations inside of the environment.

1 INTRODUCTION

In order for a robotic system to be efficient in the accomplishment of tasks, an important requirement is a correct spatial description of its underlying space that can be constructed from its own sensorial data. This description, a trustable map, makes possible a coherent interaction of the robot with its environment, so that it can perform its tasks efficiently and also can deal with unexpected situations like dynamical obstacles appropriately. The process of construction of this representation is generally named as mapping and the result is a map of environment.

Two main approaches are used to represent an environment in a map: the topological and the metric. In the topological approach, the environment map is represented by using graphs, in which the nodes are free spaces and the edges have information about connection among the nodes, for example, distance. Topological mapping makes easier the accomplishment of high level tasks, as navigation, with smaller computational cost.

In the metric approach, the geometry of the environment is defined in a more detailed way, presenting, accurately, the position of the objects inside of the environment, for instance walls, chairs and desks.

A common method used in the construction of metric maps is the occupancy grid. In this method, the environment is represented by a matrix in which each element represents a place in the environment that can be empty or occupied, or can be a unknown area.

In this work, we propose a mapping methodology with representation using an occupancy grid with a modification proposed in the model of the sonar sensor that embodies the uncertainty of the odometry system.

2 RELATED WORKS

There are many works in the literature dealing with occupancy grid. In 1987, Elfes (Elfes, 1987) proposed the occupancy grid method that is better formalized in his Ph.D thesis (Elfes, 1989). His work is implemented in two robots, Neptune and Terregator and is part of a more complete system, that integrates navigation and mapping based on sonars. This system is named Dolphin. Our current work also deals with sonars in the construction of the map in occupancy grid, but we have improved the sonar model by treating the noise of the system in a better way.

Moravec (Moravec, 1988) proposes a system, in wich the occupancy grid map is based on sonar data and stereo data. Information from a sonar sensor array and from a stereo vision system is combined to build

the metric map of environment.

In a subsequent work, Moravec (Moravec, 1996) introduces the idea of a map represented in a three dimensional occupancy grid. During the mapping, a sequence of stereo images is processed and the result of the mapping stored in a three dimensional array named evidence grid. The cells are initializes with zero value, indicating that there is no occupation evidence. After several sensor readings, the cells are filled out so that blocks of negative cells indicate free space, while positive blocks indicate obstacle. In his work, Moravec has used the theory of evidence of Dempster-Shafer. We use here the Bayesian approach also used by Elfes (Elfes, 1987).

Konolige (Konolige, 1997) has presented a method to treat the problems relative to the sonars in a more efficient way. Example of problems are the specular reflection and redundancy of reading. The proposed method is a mathematical refinement of the method introduced by Elfes (Elfes, 1987). The method is named as MURIEL (MUltiple Representation Independent Evidence Log).

In the Elfes' works, mapping is performed without considering dependence among the cell and its neighbours. This implicates in inconsistent maps when the mapping is performed in cluttered environments. In a more recent work, Thrun (Thrun, 2003) has introduced an advanced sensor model to deal with the problem of dependence among cells presented by the standard algorithm of Elfes (Elfes, 1987). The model introduced by Thrun (Thrun, 2003) verifies the values of occupation of the neighboring cells and then attributes the value of occupation to the current cell. Thrun has based his work on the Bayesian theory.

Thrun et. al. (Thrun et al., 2005) affirm that the main usefulness of the mapping with occupancy grid is in the post-processing, that is, in the map constructed. The map can be useful in several applications like, navigation, path planning, recognition of landmarks, obstacle avoidance and localization. Borenstein and Koren (Borenstein and Koren, 1991) for example, have implemented a method of obstacle avoidance in real time named VFF (Virtual Force Field). This method uses an occupancy grid map, obtained from sonar data, to define the localization of the obstacles inside of the environment.

In the Dutra's work (Dutra et al., 2003), a robot provided with an array of 24 sonars builds an occupancy grid map of its surroundings and store it in its internal memory. Later, this map is used for navigation, but the results obtained by this work in both navigation and mapping were quite influenced by accumulated errors of the odometry system. In our work, we model the sonars taking into account the intrinsic errors also to the odometry system.

Other works in the literature exemplify the use of occupancy grid maps. Kong et. al. (Kong et al., 2006) has implemented a localization system based on EKF (Extended Kalman Filter) in which features inside the environment, as corners and flat surfaces, are detected. The information obtained about the features is integrated with an occupancy grid map known a priori to yield an accurate localization of the robot in the environment.

3 THE STANDARD ALGORITHM OF OCCUPANCY GRID MAPPING

The standard algorithm formalized by Elfes (Elfes, 1989), builds the map from both the sensorial data and the robot position (localization and orientation). The mathematical formulation of the occupancy grid mapping is derived from Equation (1), which gives the value of occupation of the whole map (Elfes, 1987; Thrun et al., 2005; Thrun, 2003).

$$p(m|z_{1:t}) \tag{1}$$

In this equation m represents the obtained map, $z_{1:t}$ represents the set of measurements until the instant t. The continuos space of the environment is converted in a discrete space of cells, which form together an approximation of the environment. Thus, the map is defined as a finite set of cells $m_{x,y}$. Each cell posses a value, among 0 and 1, associated which corresponds to the state of the cell, occupied and empty, or can represent an unknown state. The value 0 means empty cell and 1 occupied cell. The notation $p(m_{x,y})$ refers the probability of a cell of index $< x,y >$ to be occupied.

The standard algorithm divides the problem of construction of the map in a set of smaller problems of estimate of the values of each cell $m_{x,y}$ separately.

$$p(m_{x,y}|z_{1:t}) \tag{2}$$

Due to reasons of numerical instabilities for probabilities near 0 or 1, it is common to calculate the log-odds of $p(m_{x,y}|z_{1:t})$ instead of $p(m_{x,y}|z_{1:t})$. The log-odds is define for:

$$l^t_{x,y} = log\frac{p(m_{x,y}|z_{1:t})}{1-p(m_{x,y}|z_{1:t})} \tag{3}$$

The probabilities are easily recovered from the log-odds ratio:

$$p(m_{x,y}|z_{1:t}) = 1 - \frac{1}{1+e^{l^t_{x,y}}} \tag{4}$$

The value of the log-odds can be estimated recursively in any instant t by the rule of Bayes applied on $p(m_{x,y}|z_{1:t})$:

$$p(m_{x,y}|z_{1:t}) = \frac{p(z_t|z_{1:t-1}, m_{x,y})p(m_{x,y}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \quad (5)$$

Supposing that we are mapping static environment, we can affirm that the current measurements of the sensors ones are independent of the last ones:

$$p(z_t|z_{1:t-1}, m) = p(z_t|m) \quad (6)$$

Because the map is decomposed in cells, this supposition is also extended all. We assume, then, the independence of each cell. Without take into account the occupation of the neighboring cells, we have:

$$p(z_t|z_{1:t-1}, m_{x,y}) = p(z_t|m_{x,y}) \quad (7)$$

This allows us to simplify the Equation (5).

$$p(m_{x,y}|z_{1:t}) = \frac{p(z_t|m_{x,y})p(m_{x,y}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \quad (8)$$

Applying the Total Probability Theorem to $p(z_t|m_{x,y})$ we obtain:

$$p(m_{x,y}|z_{1:t}) = \frac{p(z_t|m_{x,y})p(m_{x,y}|z_{1:t-1})}{\sum_{m_{x,y}} p(z_t|m_{x,y})p(m_{x,y}|z_{1:t-1})} \quad (9)$$

The Equation (9) provides the probability of a cell $m_{x,y}$ to be occupied.

4 THE PROPOSED MODEL OF THE SONAR

In our work, we consider that noise is intrinsic to both data sources, sonar sensor and odometry system, in the construction of the map. The sonar sensors have internal and external features which arouse errors in their measurements. Usually, these errors are published by the manufacturers and made available to the public. In Figure 1, for example, we can verify some typical features of a sonar Polaroid series 6500, that are used in our work. This sonar presents higher sensibility in the area near to its main axis. Besides, it presents an error of absolute measurement of +/- 1%.

Besides the typical errors of the sonar, we attempt to the errors accumulated by the odometry during the motion performed by the robot. The odometry calculates the robot's current pose in relation to the previous pose. With this, errors accumulate in an incremental way.

If we neglect these two sources of error, the final map will not be kept in accordance with the data originated from the robot, committing others applications like, navigation, path planning and others.

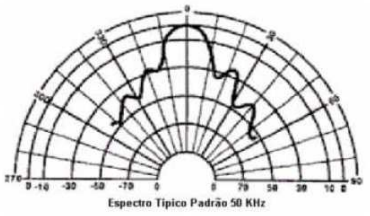


Figure 1: Features of the Polaroid sensor.

To deal whit those errors, we looked for a way of include them in the occupancy grid map representation. Thus, we modified the sonar sensor model of the standard algorithm, so that, in a probabilistic way, the typical errors of the sonars and odometry are incorporated to the value of occupation of the cell in the grid (Equation 10).

$$p(z_t|m_{x,y}) = \frac{1}{2\pi\sigma_{z_t}\sigma_{\theta_t}} \times e^{\left[-\frac{1}{2}\left(\frac{(D_{x,y}-z_t)^2}{\sigma_{z_t}^2}+\frac{(\theta_{x,y}-\theta)^2}{\sigma_{\theta_t}^2}\right)\right]} \quad (10)$$

Where:
z_t represents the sensor reading in the instant t;
θ_t represents the orientation of the sensor;
σ_{z_t} represents the standard deviation regarding the error in the measured distance by the sensor;
σ_{θ_t} represents the standard deviation regarding the error in the orientation of the sensor;
$D_{x,y}$ represents the Euclidean distance between the sensor and the cell $m_{x,y}$;
$\theta_{x,y}$ represents the angle of orientation of the cell $m_{x,y}$.

The standard deviation regarding the error in the measured distance depends the value of the odometry error of a translational movement, that is,

$$\sigma_{z_t} = z_t \times k + f \quad (11)$$

Where:
k is the factor of errors intrinsic of the sonar (in our case, +/- 1%);
f is a function which describes the odometric systematic error, when the robot moves linearly.

The standard deviation regarding the error in the orientation angle of the sensor depends on the value of the odometry error of a rotational movement, that is,

$$\sigma_{\theta_t} = \frac{\beta}{2} + g; \quad (12)$$

Where:
β is the solid angle subtending the main lobe of the area of sensitivity;
g is a function which describes the odometric systematic error, when the robot moves in a rotational way.

Function f in Equation 11 and function g in Equation 12 were experimentally deduced from several data samples. Figure 2 shows a graph of the sensor model deduced.

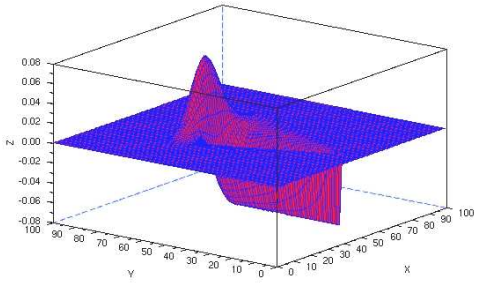


Figure 2: Graphics of the proposed model.

With this modification, the value of the probability of occupation of a cell passes to be weighed by the uncertainties in the real robot position, due to the odometry errors. Thus, the map is corrupted gradually, representing the real error.

5 EXPERIMENTS

To validate de proposed model, we have made some experiments with the mobile robot named Galatia, model Pioneer 3-AT of the ActivMedia Robotics, provided with two array sonar sensor (front and back) and odometry system (Figure 3).

Figure 3: The robot Pioneer 3-AT.

The experiments are performed inside of the Department of Computing Engineering and Automation - UFRN, trying to map the corridors. The first experiment is performed using the standard algorithm of occupancy grid mapping without relying on the odometry errors. The result of this experiment can be seen in Figure 4.

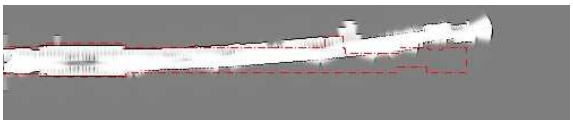


Figure 4: Map without representation of the errors.

Later, we perform the same experiment again using the probabilistic sonar model proposed here, to represent the systematic errors of the sonars and odometry (Figure 5).

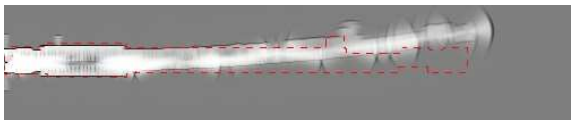


Figure 5: Corrupted map.

In the maps, the white cells represents free areas or no obstacles, the black cells represents occupied areas or obstacles and the grey cells represents the unknown areas. The sketched lines show the real contour of the corridors.

6 CONCLUSIONS

In this work, we propose a modification in the sensor model used in the standard algorithms based on occupancy grid mapping, including in the probabilistic model the intrinsic uncertainties of the robotic system. In order to consider these uncertainties in the mapping, we have found a way to include them in the representation of the map of the environment. Thus, we basically modify the sensor model so that, in a probabilistic way, the typical errors from sonars and odometry can be incorporated in the occupancy value of a cell. The model was tested in practice with the robot Pioneer 3-AT and demonstrated to be correct, giving the right expectation for the error

Based on the presented results, we can see that the proposed model supplies a more realistic way to represent a mapped environment using occupancy grid, being known that the originated information from the robot have errors, and what are really these errors. That is, the errors corrupt the quality of the map, showing in this way coherence whit the sensorial data.

As extension of this work, we will improve the treatment of the incoherent readings from the sonars, like speculate reflections and so. Furthermore, a way for relocation of the robot, when the errors grow a lot, will be studied and implemented.

ACKNOWLEDGEMENTS

We thanks CAPES, RNP and CNPq by the financial support.

REFERENCES

Borenstein, J. and Koren, Y. (1991). The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7(3):278–288.

Dutra, P. R. C., de Souza, M. M., Andriolli, G. F., lvares, A. J., and Ferreira, J. C. E. (2003). Navmap: Um sistema para navegao por mapeamento do rob mvel nomad xr4000. In *VI Simpsio Brasilieiro de Automao Inteligente (SBAI)*, Bauru, SP.

Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265.

Elfes, A. (1989). *Occupancy Grid: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Carnegie Mellon University, Pittsburg, Pennsylvania.

Kong, F., Chen, Y., Xie, J., Zhang, G., and Zhou, Z. (2006). Mobile robot localization based on extended kalman filter. In *6th World Congress on Inteligent Control and Automation*, Dalian, China.

Konolige, K. (1997). Improved occupancy grids for map building. *Autonomous Robots*, (4):351–367.

Moore, R. and Lopes, J. (1999). Paper templates. In *TEMPLATE'06, 1st International Conference on Template Production*. INSTICC Press.

Moravec, H. P. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74.

Moravec, H. P. (1996). Robot spatial perception by stereoscopic vision and 3d evidence grid. Technical Report CMU-RI-TR-96-34, CMU Robotis Intitute, Pittsburgh, Pennsylvania.

Smith, J. (1998). *The Book*. The publishing company, London, 2nd edition.

Thrun, S. (2003). Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, (15):111–127.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, Cambrige, Massachusetts, USA.

VISUAL BASIC APPLICATIONS FOR SHAPE MEMORY ELEMENTS DESIGN USED IN INTELLIGENT SYSTEMS

Sonia Degeratu, Petre Rotaru, Horia Octavian Manolea
Faculty of Electromechanical Engineering and Faculty of Physics, University of Craiova, Romania
Faculty of Dentistry, University of Medicine and Pharmacy, Craiova, Romania
sdegeratu@em.ucv.ro, protaru@central.ucv.ro, manoleahoria@gmail.com

Gheorghe Manolea, Anca Petrisor, Bizdoaca Nicu George
Faculty of Electromechanical Engineering and Faculty of Automation, University of Craiova, Romania
ghmanolea@gmail.com, apetrisor@em.ucv.ro nicu@robotics.ucv.ro

Keywords: Shape memory alloy, austenite phase, martensite phase, cantilever strip, helical spring.

Abstract: The paper presents the design strategies for two typical configurations of intelligent systems, using as active elements the Shape Memory Alloy (SMA) cantilever strip and the SMA helical spring. Based on the advantages and unique properties of shape memory alloys the authors defined the operating mode, the mechanical considerations and the design assumptions for these two examples. It also includes the experimental results of thermal analysis in order to determine the transformation temperatures for studied SMA elements. A comprehensive graphical interface, which runs under Visual Basic environment, has been developed for each design strategy. Each one provides a user friendly environment that allows intelligent system parameters configuration as well as the choice of the most adapted analysis methods and data displaying. At this moment, these two Visual Basic applications are used for engineering purposes as well as didactical ones.

1 INTRODUCTION

The shape memory alloys (SMA's) possess the ability to undergo shape change at low temperature and retain this deformation until they are heated, at which point they return to their original shape (Van Humbeeck, 2001), (Waram, 1993).

The unique behavior of SMA's (e.g. shape memory effect, pseudo-elasticity) is based on the temperature-dependent austenite-to-martensite phase transformation on an atomic scale, which is also called thermoelastic martensitic transformation.

The SMA's can exist in two different temperature-dependent crystal structures (phases) called martensite (lower temperature) and austenite or parent phase (higher temperature).

When martensite is heated, it begins to change into austenite and the temperatures at which this phenomenon starts and finishes are called austenite start temperature (As) and respectively austenite finish temperature (Af). When austenite is cooled, it begins to change into martensite and the temperatures at which this phenomenon starts and finishes are called martensite start temperature (Ms) and respectively martensite finish temperature (Mf).

SMA's have been extensively studied as functional materials for a wide range of applications: robotics, actuators, automobiles, aerospace, military field, medicine, toys, electronics, optical industry, constructions, agroalimentary sector (Degeratu, 2003), (Dolce, 2001), (Van Humbeeck, 1999).

The use of SMA's can sometimes simplify a mechanism or device, reducing the overall number of parts, increasing reliability and therefore reducing associated quality costs. Moreover, their functioning can be scheduled and controlled by establishing an adequate strategy in respect to the operating mode (Bizdoaca, 2006), (Petrisor, 2007).

2 CHARACTERISTICS OF SMA'S

The thermoelastic martensitic transformation causes the folowing main properties of SMA's: one-way and two-way shape memory effect, hysteresis behavior, superelasticity, vibration damping capacity (Dolce,

2006), (Waram, 1993), (Van Humbeeck, 2001). The two applications presented in this paper are mainly using the two-way shape memory effect representing the ability of SMA's to recover a preset shape upon heating above the transformation temperatures and to return to a certain alternate shape upon cooling.

3 DESIGN STRATEGIES FOR SMA ELEMENTS

The first step an engineer should take when undertaking a design involving shape memory material is to clearly define the design requirements.

This article includes the design strategies for the SMA cantilever strip and for a SMA spring working against a conventional steel spring (referred to in this case as the "biasing" spring). In both design models the friction effect is neglected and a linear stress-strain behavior is assumed (Nasser, 2005).

3.1 Operating Modes of SMA's

The most used operating modes of SMA's are: free recovery, constrained recovery and work production.

The two Visual Basic applications presented in this paper use a work production operating mode. In this kind of operating mode a shape memory element (SME), such as a strip or a helical springs, works against a constant or varying force to perform work. The element therefore generates force and motion upon heating.

3.2 Transformation Temperatures

SMA's exhibit a large temperature dependence on the material shear modulus, which increases from low to high temperature. Therefore, as the temperature is increased the force exerted by a SME increases dramatically (Dolce, 2001), (Nasser, 2005).

This section presents the transformation temperatures obtained for the studied SMA elements (strip and helical spring) using Thermal Analysis Methods. Ni-Ti-Cu (Raychem proprietary alloy) is the material used for the two SMA elements.

Thermogravimetric Analysis (TGA), Differential Thermal Analysis (DTA) and Differential Scanning Calorimetry (DSC) methods were used to determine the required parameters. The measurements were carried out on a Perkin Elmer Thermobalance in dynamic air atmosphere, showing that the sample's mass does not undergo any changes at heating and cooling. In consequence, the TGA curves are ignored.

SMA strip transformation temperature

The temperature control program used for SMA strip measurements contains the following sequences:

- heating from 30°C to 160°C at 5°C/min;
- holding for 10min at 160°C;
- cooling from 160°C to 20°C at 5 °C/min.

The DTA and DSC curves are presented in Figure 1. By analyzing this figure we can observe two phase transitions. The first occurs during the heating while the second one appears during the cooling process. The details of these thermal effects are presented in figures 2 and 3 (reported from the DSC curve).

Figure 2 shows that the determined transformation temperatures at heating are A_s=80°C and A_f=111°C. The enthalpy of the endothermal transition process is ΔH_h=36.8858J/g. The temperature corresponding to maximum transformation speed is 98.79°C.

The transformation temperatures at cooling result from Figure 3: M_s=69°C and M_f=48.25°C. The enthalpy of the exothermal transition process is ΔH_c=-28.7792J/g and the temperature corresponding to maximum transformation speed is 59.75°C.

SMA helical spring transformation temperature

The transformation temperatures of SMA helical spring are obtained by similar measurements as in the case of SMA strip, using the following temperature-control sequences:

- heating from 30°C to 100°C at 5°C/min;
- holding for 10 min at 100°C;
- cooling from 100°C to 20°C at 5 °C/min.

The form of DTA and DSC curves is similar to the ones represented in Figure 1, for 6.849mg SMA spring sample. The experimental transformation temperatures at heating are A_s=58.89°C and A_f=67.93°C. The enthalpy of the endothermal transition process is ΔH_h=9.2J/g and the temperature corresponding to maximum transformation speed is 60.42°C. The transformation temperatures at cooling are M_s=45°C and M_f=33°C, the enthalpy of the exothermal transition process is ΔH_c = -5.03J/g and the temperature corresponding to maximum transformation speed is 39.07°C.

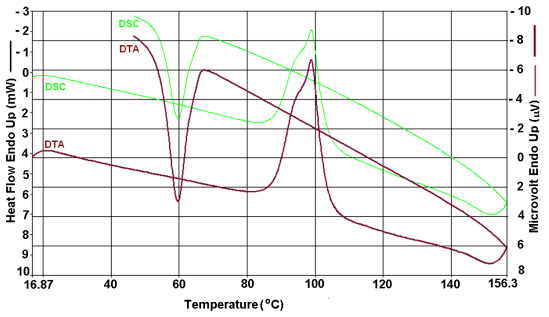


Figure 1: DTA and DSC curves for 18.275mg SMA strip.

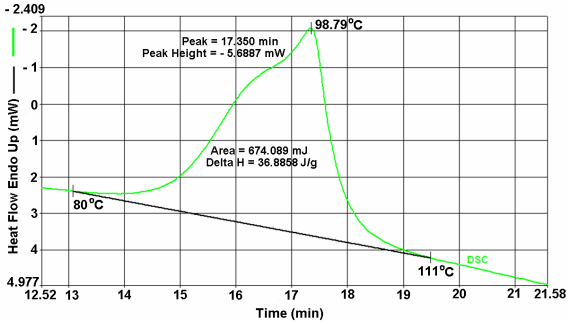


Figure 2: Detail of DSC curve for computation transition at heating of SMA strip.

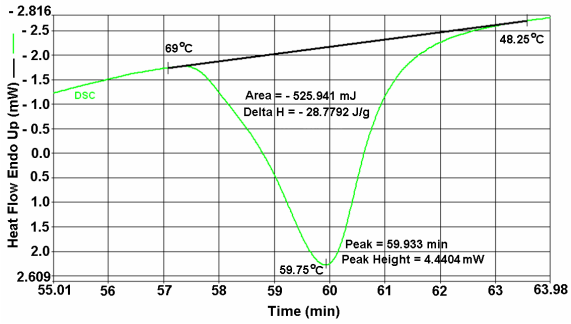


Figure 3: Detail of DSC curve for computation transition at cooling of SMA strip.

4 VISUAL BASIC APPLICATIONS

Two Visual Basic applications were implemented for SMA cantilever and SMA helical spring with biasing spring. This section details these applications.

Visual Basic application for SMA cantilever strip

Cantilevers made from SMA strip can be used to provide a lifting force and a nominal amount of motion by heating.

For the design example, assume that a cantilever is required to lift a force F=2N (at electrically energized) for a distance of 5mm (required motion) and that the maximum allowable width is 3.8mm. The high temperature stress is σ=140MPa. The operational temperatures, at heating and cooling, are those determined at section 3.2, that are 111°C and respectively 48.25°C. For these temperatures the experimental determined values of Young's modulus are E_h=59000MPa and respectively E_l=6900MPa.

A Visual Basic project SMA cantilever strip was implemented. The application background computations are entirely presented in (Degeratu, 2003).

After providing the initial parameters in the dialogue boxes of the user interface (Figure 4), by pressing the compute button the designed parameters are being displayed in the upper part of the window: cantilever length, thickness and width, reset force, high and low temperature deflections. The middle of the window displays the typical SMA cantilever configuration as well as all design parameters.

SMA cantilever strip can be used to provide thermal control of a microswitch or automatic control of a cooling fan (Waram, 1993, Bizdoaca, 2006).

Visual Basic application for SMA spring with biasing spring. In the work production operation mode the SMA helical spring can work against varying forces such as a steel spring, fluid presure or a magnetic force. In this application the varying force is produced by a steel spring.

For the design example, assume the following requirements: a Ni-Ti-Cu spring/biasing spring combination is required which provides a net force F_n=3N with a 8mm stroke; the maximum cavity length and diameter are 38mm and respectively 5.5mm.

Assume that the force exerted by the biasing spring F_h=2N, the maximum corrected shear stress T_c=175MPa, the SMA spring index c=6 and the low temperature shear strain γ_l=0.015 (in order to ensure a good cyclic life of 50000 cycles). The operational temperatures, at heating and cooling, are those determined at section 3.2, that are 67.93°C and respectively 33°C. For these temperatures the experimental determined values of shear modulus are G_h=16890MPa and respectively G_l=3759MPa. Also assume that the two springs are separated by a plug of thickness 2.5mm.

A Visual Basic project SMA spring with biasing spring was implemented. The application background computations are entirely presented in (Degeratu, 2003) and (Degeratu, 2007).

Using standard steel spring design procedure, assume that the maximum shear stress for the wire T = 675MPa and the shear modulus G = 79300MPa.

When the VISUAL BASIC project for SMA spring with biasing spring design is run, a user interface is displayed, Figure 5.

First the user has to provide the initial parameters in the dialogue boxes in the lower part of the interface. By clicking on the Compute button, the designed parameters are being displayed for both SMA spring and biasing spring in the upper part of the interface. The total actuator system comprised of SMA spring and biasing spring is shown in the middle part of the interface.

This configuration is frequently used for SMA Controlled Valves (Bizdoaca, 2006) and SMA Latching Mechanisms and SMA Bell Crank Mechanisms (Nesser, 2005, Waram, 1993).

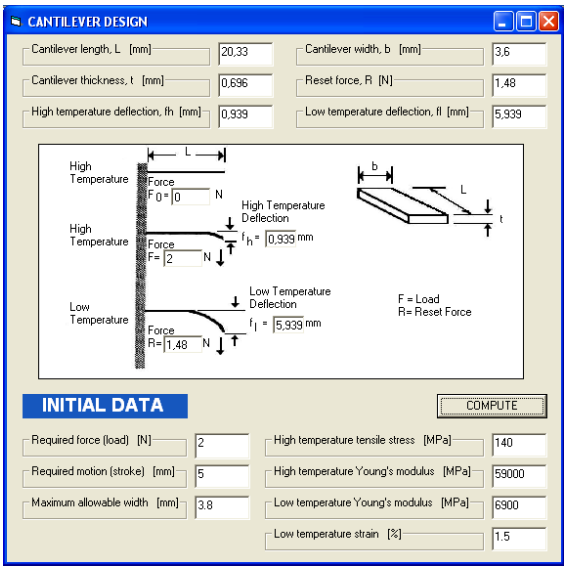


Figure 4: Dialog interface for SMA cantilever design.

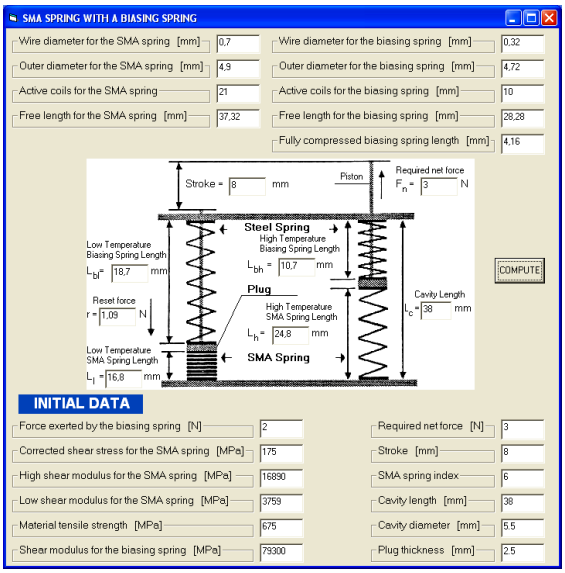


Figure 5: Dialog interface for SMA spring with biasing spring

5 CONCLUSIONS

The paper presents two design strategies for SMA cantilever and SMA spring with biasing spring. For these two design strategies the authors defined: the operating mode, the mechanical considerations and the design assumptions.

Using Thermal Analysis Methods the authors determined the experimental transformation temperatures for the studied SMA elements. These temperatures were necessary to precisely establish the shear modulus values for a high-quality design. In addition, for each design strategy, a Visual Basic application was developed, providing:

- adequate dialogue boxes for fast and easy initial parameters configuration;
- fast computation and display of all required information for a complete SMA element design;
- remarkable facilities to analyze results and choose an optimal solution.

These two Visual Basic applications are already used by ICMET Craiova for engineering purposes and by the Faculty of Electromechanical Engineering of Craiova for didactical ones.

REFERENCES

Bizdoaca, N., G., Degeratu, S., Pana, C., Vasile, C., 2006. *Fuzzy Logic Controller for Hyperredundant SMA Tendons Actuated Robot*. 7th International Carpathian. Control Conference. Ostrava Beskydy, Czech Republic.

Bizdoaca, N., G., Degeratu, S., Diaconu, I., Petrisor, A., Patrascu, D., 2006. *Demining Robots: Behavior Based Control Strategy*. Annals of University of Craiova, Craiova, Romania.

Degeratu, S., Bizdoaca, N., G., 2003. *SMA's. Fundamental Notions, Design and Applications*. Universitaria Press, Craiova, Romania.

Degeratu, S., Manolea, H., O., Bizdoaca, N., G., 2007. *Shape Memory Elements Design for Intelligent Systems*, University Annuars from Oradea, Romania

Dolce, M., Cardone, D., 2001. *Mechanical Behavior of Shape Memory Alloys for Seismic Applications 2: Austenitic NiTi Wires Subjected to Tension*. Int. Met. Schi, 43.

Nasser, S., N., Choi, J., Y., Guo, W., Issacs, J., B., Taya, M., 2005. *High Strain-Rate, Small Strain Response of NiTi Shape Memory Alloy*. Journal of Engineering Materials and Technology, vol.127.

Petrisor A., Rosca D., Bizdoaca N., G., Degeratu S., 2007, *Mathematical Model for Walking Robot with SMA Ankle*, ISI Proceedings ICINCO, Angers, France.

Van Humbeeck, J., 1999. *Non-Medical Applications of Shape Memory Alloys*. Matter. Sci. Eng. 237A-275A

Van Humbeeck, J., 2001. *Shape Memory Alloys: A Material and a Technology*. Adv. Eng. Mater., 3.

Waram, T., C., 1993. *Actuator Design Using Shape Memory Alloys*. Ontario, Canada.

FLEXIBLE TRAJECTORY GENERATION TO EXTEND HUMAN-ROBOT INTERACTION WITH DYNAMIC ENVIRONMENT ADAPTATION

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

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

Keywords: Human-Robot Interaction.

Abstract: In our daily life, we use many elements that help us by means of a higher protection level (thimble, door stop) or by improving our dexterity (funnel, compasses). Both kinds of elements allow us to execute well known tasks with less concentration, faster, and, above all, improving performance. Like the real tools mentioned above, in the robotics field, virtual constraints enhance human-machine interaction. This work presents a multi-parametric behaviour model for an agent that increases task safety, and enables higher integration possibilities. The model presented here allows the perturbation of a programmed task, by introducing virtual elastic and viscous forces. This work presents the behaviour model, a description of it's implementation and experimental results in human-robot interaction.

1 INTRODUCTION

Some robotic applications need to benefit from the accuracy and precision of a robotic system, while preserving a degree of human control. Some of such application fields are assistive or surgical robotics. The goal of a robotic assistant is to provide motion commands that enhance precision, stability, safety and skilfulness. Significant research of assistant robotics systems is illustrated in Dario et al (1999), as an assistant for colonoscopy. In assistive robotics, due to the difficulties in modelling the environment with enough definition or under changing scenarios, it is necessary to aid the robotic arm to adapt its movements to the real environment or to the needs of the user.

These requirements have motivated the study and development of behaviour models. The model must allow software-generated force, velocity and position signals applied to human operators through the robotic system. A behaviour model can improve human performance in robot-assisted manipulation tasks, restricting movements into a region, constraining velocities in a specific direction and/or introducing virtual correction forces. The presented multi-parametric behaviour model allows perturbing on-line a predefined path, applying forbidden region restrictions, and tuning model parameters (like masses, viscosity, and stiffness).

There's many procedures performed nowadays by surgical robots, most of them are in the orthopaedic field, using CAD/CAM surgical systems, or teleoperated surgical robots for laparoscopic interventions. Examples of successful procedures performed with the Zeus system are reported in Zhou et al (2006). This success has been achieved as a result of the human enhancement that robotic-assisted surgery systems offer.

Despite this success, there are several key challenges that require to be solved in order to achieve a complete development of surgical robotic systems. Kanade (2004) carries out an analysis of technological barriers. Introduced in Rosenberg (1993), virtual fixtures are playing an important role in the development of human-machine cooperative interaction enhancement. Several groups have integrated different implementations of virtual fixtures in surgical robotic systems, as described in

Bettini et al (2002). The work presented increases human capabilities, integrating three categories of restrictions: geometric, kinematic and dynamic. On the other hand, by being a parametric representation, different responses can be achieved with the same model. Changing the virtual mass, length, elasticity and viscosity... the behaviour can be tuned according to specifications of a certain task.

2 FLEXIBLE TRAJECTORY GENERATION

2.1 Task Oriented Trajectories

Many robots execute tasks by repeating a programmed sequence of movements. These sequences can be stored either with teaching by demonstration techniques or by using the corresponding robot programming language.

The work presented here aims to provide some means of changing this trajectory during the execution of the programmed task, through the action of a human that perturbs the robot movement by steering the end effector in the desired direction, with a given force, F.

The developed method is based on the definition of the robot path by means of two functions: $A(t) = (x, y, z)$ and $B(t) = (x, y, z)$. As the Figure 1 shows, the segment defined by P_1 and P_2 is considered the non-perturbed position of the end-effector.

The programmed path can be modified if an external force is applied on the robot, or the presence of an obstacle is detected on its way. In both cases, the trajectory is modified producing an elastic movement away from the trajectory of the end-effector. Considering the two trajectory functions $A(t)$ and $B(t)$, the resulting behaviour can be compared with a cable car, where the cabin is modelled by a linear segment held from two extreme points. The two links are springs with non linear behaviour endowed with damping. In this way the end-effector trajectory (the cabin) can be moved away with respect to the theoretical trajectory (the cable) by applying a perturbation force which is perceived by the user itself.

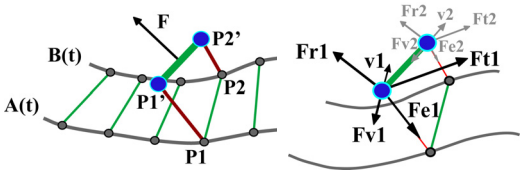


Figure 1: End-effector trajectory and the resulting forces.

The perturbation of the position caused by an external interaction produces a movement away from the position of the segment $P_1(t) - P_2(t)$. The representation for one such segment can be seen in Figure 1.

The perception of the increasing effort enables us to get better results to produce smooth movements with a reasonable effort of the user. The distance $P_1 - P_1'$ and $P_2 - P_2'$ that can be produced in each movement is the result of the following four set of forces: Two vectors Fr_1 and Fr_2 equivalent to the forces and torques measured on the force sensor. A second component is an elasticity force Fe_1 and Fe_2. A third element is an attraction force Ft_1 and Ft_2, towards the non perturbed trajectory. And a forth factor corresponds to viscosity, Fv, that contributes to smoothing the trajectory when the robot returns to its programmed trajectory after a perturbation.

The forces that appear at each instant are shown in Figure 1. The resultant of these two systems, of four forces each, is what determines the segment dynamics.

2.2 Behaviour Model for the Human-Robot Interaction

The implementation of this model uses the motion equation for a rigid body. Below, the structure of this implementation is described, as well as several equations and concepts needed. For a system of n particles $X(t)$ extends to

$$X(t) = \begin{pmatrix} x(t) & v(t) \end{pmatrix}^T \quad (1)$$

We define the sum of forces acting at this particle at time t as $F(t)$. Then, if the particle's mass is m, the change of $X(t)$ is defined by

$$\frac{d}{dt} X(t) = \begin{pmatrix} v(t) & F(t)/m \end{pmatrix}^T \quad (2)$$

Given these equations, a simulation starts with initial conditions for $X(0)$ and uses a numerical

solver to trail the change of X over time. When simulating a rigid body,

$$X(t) = \begin{pmatrix} x(t) & R(t) & P(t) & L(t) \end{pmatrix}^T \quad (3)$$

With $R(t)$ representing the orientation of the body, $P(t) = M \cdot v(t)$ it's linear momentum and $L(t) = I(t) \cdot \omega(t)$ it's angular momentum. Where the mass, M, of the solid is constant; and the inertia tensor is computed as $I(t) = R(t) I_{body} R(t)^T$, with I_{body} also constant. With this, (2) is now

$$\frac{d}{dt} X(t) = \begin{pmatrix} v(t) & \omega(t) * R(t) & F(t) & \tau(t) \end{pmatrix}^T \quad (4)$$

Where $F(t)$ is the sum of forces applied to the body, and $\tau(t)$ is the sum of torques applied to it.

Arrived at this point, different behaviours can be designed and implemented changing parameters like the mass, and the inertia tensor. But what brings further capabilities of this model is the insertion of virtual forces and torques, so that $F(t)$ and $\tau(t)$ become the sum of the external actions and the virtual ones. This virtual forces and torques are designed to add different types of constraints like impedance walls, viscosity of the medium, forbidden regions or elastic correction forces.

The basis of the proposed model is defined as a rigid body with two masses and a rigid link between them. The virtual environment is a spring connection element between each of the masses and a reference point, as well as a viscosity of the medium.

Using this model, the reference point location, damping, elasticity, mass and rigid link length are parameters. Also, a constant virtual force can be added to the system in either solid or world reference. The equations of the virtual forces in the solid reference frame are:

$$A1_{solid} = R(t)^{-1} * (A1_{world} - x(t)_{world}) \quad (5)$$

$$F1_{viscous}(t) = (v(t)_{solid} + \omega_{solid} \times P1_{solid}) \cdot (-c) \quad (6)$$

$$F1_{elastic}(t) = (A1_{solid}(t) - P1_{solid}) \cdot (k) \quad (7)$$

$$F1_{solid}(t) = F1_{elastic}(t) + F1_{viscous}(t) + F1_{constant}(t) \quad (8)$$

$$\Gamma 1_{solid}(t) = P1_{solid} \times F1_{solid}(t) + \Gamma 1_{constant}(t) \quad (9)$$

The analogous equations can be written for the second mass. At this point, the external forces can be included in the model, and the resulting forces and torques in the world reference are

$$F1_{world}(t) = R(t) * (F1_{solid}(t) + F2_{solid}(t) + F_{extern}(t)) \quad (10)$$

$$\Gamma 1_{world}(t) = R(t) * (\Gamma 1_{solid}(t) + \Gamma 2_{solid}(t) + \Gamma_{extern}(t)) \quad (11)$$

The added value that this model of behaviour provides is based on the fact that both virtual and real interactions are defined with a natural, intuitive and transparent approach.

From the dynamic behaviour point of view, the parameters that have been tuned for a desired response are the mass, distance between the spheres, viscosity of the medium and elasticity of the virtual links.

These parameters can be fixed for a desired performance during the execution of a task, but their values can also be tight to a parameter that evolves during the execution of the task. This dynamic adaptation can be model based and environment based. In the first case, both viscosity and elasticity parameters are function of the minimum distance to an object.

3 EXPERIMENTAL SYSTEM

An experimental setup has been designed in order to evaluate the different behaviours, shown in Figure 2. The system developed includes a 6 degrees of freedom robotic arm manufactured by Stäubli, a motion controller by Adept, an ATI force and torque sensor and a Dell personal computer.

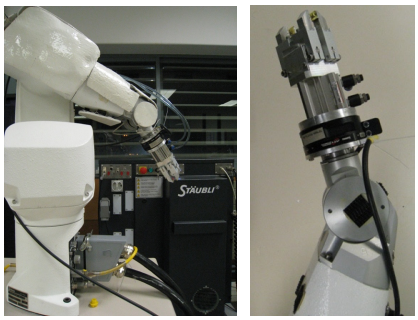


Figure 2: Experimental system used for the evaluation of the proposed behaviour.

The robotic arm is initially programmed to perform task. The trajectory is specified by either position and orientation, transformation matrices or two trajectory paths as mentioned before. Geometric, cinematic and dynamic restrictions of the robotic arm are also considered when the trajectory is programmed.

The motion controller is programmed with a low level firmware that computes motion commands. A hybrid position/force control loop is running with a 16 millisecond period time. Motion orders can be sent in either Cartesian or Joint type, as well as in either incremental or absolute modes. The motion

controller runs a communication server under TCP/IP protocol.

The force sensor measures forces and torques applied at the tool sustained by the robot. An analog to digital converter hardware and a calibration routine are called by the personal computer. As a result, two vectors are ready to be introduced to the behaviour model: external forces and torques.

The personal computer runs the hybrid force/position control loop linked to the motion controller and the force/torque sensor. As described in next section, a set of algorithms have been developed to accomplish all computing requirements.

The control loop accomplishes the main functionality of the software developed for the system. The sequence of routines called at each cycle is:

- Capture position and orientation of the robot
- Capture voltages at the force sensor
- Compute forces and torques from voltages
- Subtract weight of the tool using its orientation
- Calculate new state according to the behaviour
- Send new state to the motion controller

4 RESULTS AND CONCLUSIONS

The model proposed, based on a double virtual mass body, and elastic and viscous links has been tested with some results shown in Figure 3. Different experiments have been designed and tested. Simple tasks like object pick and place, path following or surface polishing are accomplished. During the execution of the overall task, a perturbation is introduced by means of external forces and torques. The system reacts to the perturbations measured by the force sensor, computing new positions according to the described model, and sending the perturbed positions to the motion controller. After the real perturbation, the virtual forces described earlier act as guidance of the endpoint, smoothly driving the end effector back to the pre-programmed path.

In order to increase human capabilities, some cooperative tasks include virtual constraints. The proposed model integrates three categories of restrictions: geometric, kinematic and dynamic. In order to accomplish the geometric constraints, a proximity library (Giralt and Hernansanz 2006) and a surface navigation method (Hernansanz et al 2007) have been developed and incorporated.

As it's a parametric model, different responses can be achieved with the same model. By changing the values of the virtual mass, elasticity and viscosity, the behaviour can be tuned according to specifications of a certain task.

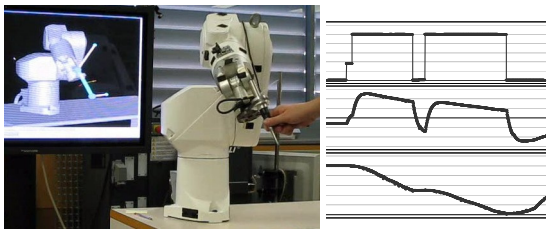


Figure 3: Experimental results. The left picture shows the perturbation of a trajectory by means of external forces. The two below the picture describe numerical values of the reaction. First graph is the evolution of the external force applied to the endpoint of the manipulator. The graph in the middle shows the evolution of velocity of the end effector. Last graph describes position response to these force steps.

REFERENCES

Dario, P., Carrozza, M.C., Pietrabissa, A., 1999. Development and in vitro testing of a miniature robotic system for computer-assisted colonoscopy. In Computer Aided Surgery; 4(1):1-14.

Zhou, H.X., Guo, Y.H., Yu, X.F., Bao, S.Y., Liu, J.L., Zhang, Y., Ren, Y.G., Zheng, Q., 2006. Clinical characteristics of remote Zeus robot-assisted laparoscopic cholecystectomy: A report of 40 cases. In World J Gastroenterol, 12(16):2606-2609.

Kanade, T., 2004. Workshop on Medical Robotics. International Advanced Robotics Program. Hidden Valley, Pennsylvania, USA., pp. 19-22.

Rosenberg, L. B., 1993. Virtual fixtures: Perceptual tools for telerobotic manipulation. In Proceedings of the IEEE Annual Int. Symposium on Virtual Reality, pp. 76–82.

Bettini, A., Lang, S., Okamura, A., Hager, G., 2002. Vision Assisted Control for Manipulation Using Virtual Fixtures: Experiments at Macro and Micro Scales. In Proceedings International Conference on Robotics and Automation, pp. 3354-3361.

Giralt X., Hernansanz A., 2006. Optimització de consultes de proximitat en entorns robotics. In Proceedings 2ª Jornadas de Recerca en Automàtica, Visió i Robòtica ESAII IOC IRI, pp. 257-261.

Hernansanz A., Giralt X., Rodriguez A., Amat J., 2007. RPQ: Robotic Proximity Queries. Developement and Applications. In Proceedings of The 4th International Conference on Informatics in Control, Automation and Robotics. INSTICC Press.

MULTI-AGENT ARCHITECTURE FOR SIMULATION OF TRAFFIC WITH COMMUNICATIONS

Pedro Fernandes and Urbano Nunes
Institute of Systems and Robotics, University of Coimbra, Polo II, Coimbra, Portugal
pedro@isr.uc.pt, urbano@isr.uc.pt

Keywords: Multi-agent systems, Traffic simulation, Network simulation, Intelligent Transportation Systems.

Abstract: Inter-vehicle communications, in the context of Intelligent Transportation Systems, will probably bring a significant improvement in both traffic safety and efficiency. In order to evaluate in what measure this is true, traffic simulations that take into account the communications between vehicles are needed.
In this paper, we propose an agent-based architecture, in which the simulation and management of the inter-vehicle communications are integrated in the simulation of vehicles, in a hierarchical multi-agent environment. An overview of multi-agent methodologies, platforms, among other, is also presented.

1 INTRODUCTION

Human transport in urban spaces relies mostly on individual vehicles, congesting the transportation networks. Studies and simulations of traffic have been made for decades, through macroscopic, mesoscopic and microscopic traffic simulators.

Recently, in the context of Intelligent Transportation Systems (ITS), vehicle to-vehicle (V2V) and vehicle to infrastructure communications (V2I) are being developed, namely the DSRC (Dedicated Short Range Communications), operating in 5.9 GHz band. The standardization process is almost finished under IEEE 802.11p/IEEE 1609.x (also designated by WAVE: Wireless Access in Vehicular Environments) and IEEE 1556 standards. In EU, the International Organization for Standardization (ISO), under the Technical Committee TC204, is working in similar standards – Communication Air Interface Long and Medium Range (CALM) – to ensure European-wide inter-vehicle communications interoperability.

To study the impact that such systems may have in the near future, efforts to integrate traffic and network simulators have been pursued. However, a useful solution has not been reached yet.

The integration of both traffic and network simulations in a system may be considered a complex task, due to a vast set of reasons, such as the intrinsic complexity of traffic theory, the wireless network transmissions involved, the real-time constraints and the distributed nature of the system, among others. At the present, traffic theory does not account to driver behavior changes due to the existence of communications. Therefore, equation-based modeling is not the most appropriate method to use in simulation. Agent-based modeling allows the development of a more adaptive system, and although system validation may be more difficult, it can be done at both system and individual levels.

2 RELATED WORK

The use of intelligent agents in traffic simulation is an emergent area of research. Table 1 presents some of the works in this area and simulators integration.

Table 1: Related work.

Vogel and Nagel (2005)	Multi-agent simulation model with application to Berlin traffic.
Hallé et al., (2004)	Agent-based architecture to develop centralized and decentralized platoons.
Li et al., (2006)	Urban traffic control system using multi-agent technology.
Dresner and Stone (2005)	Agent-based simulation of a traffic intersection.
Eichler et al., 2005	Coupling traffic and network simulators and a V2V messaging application.
Piorkowski et al., 2006	Network- and application-centric evaluation oriented architecture.
Avila et al., 2005	Intersection warning system, coupling traffic and network simulators.

3 DEVELOPMENT ISSUES

According to Wooldridge (2002), "an agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives". Autonomy, situatedness, reactivity and proactivity are some important characteristics of agents. In a multi-agent architecture, issues like organization, coordination and security are also relevant.

To develop a MAS system, a disciplined approach should be followed, and an appropriate platform should be chosen, along with communication standards between agents – preferably based on open standards – and appropriate ontologies. The simulation platform must also be selected or developed.

3.1 Methodologies

Several proposed methodologies to develop a MAS may be considered. Prometheus, Gaia and Tropos are some of the examples in the literature. However, not all existing methodologies are appropriate for every problem. Some of them aim at generality. Others focus more on specific platforms and languages, gaining in detail and adaptability.

Prometheus methodology was proposed by Padgham and Winikoff (2002). According to the authors, the reason why they proposed a new methodology was the methodology claimed detail, support of BDI (Beliefs, Desires and Intentions) agents, scaling ability and tool support. To support design and development of multi-agent systems using Prometheus, Padgham and Winikoff developed the Prometheus Design Tool, that implements the three phases of Prometheus and process some consistency checking.

Gaia methodology presents a general approach, to allow its use for a broad type of agent-based systems. However, this characteristic, which is one of its strengths, is also its most pointed weakness, since the detailed design phase and implementation have intentionally been left out.

Table 2: Methodology phases.

Methodology	Phases
Prometheus	1-specifications; 2-architectural design; 3-detailed design; 4-implementation.
Gaia	1-requirements; 2-analysis; 3-design.
Roadmap	1-requirements; 2-analysis; 3-design.
OPM/MAS	1-requirements; 2-analysis; 3-design; 4-deployment.

Other methodologies appear in literature, namely ROADMAP, Tropos, SODA, MESSAGE, MaSE, MAS-CommonKADS, AOR, OPM/MAS, MASSIVE, Ingenias, DESIRE, PASSI and AgilePASSI.

In Table 2, the phases of some methodologies are presented.

Prometheus seems an appropriate methodology for initial system development. All the relevant phases are covered conveniently, and PDT tool allows consistency and completeness checking through the steps of each of the phases.

3.2 Platforms

Choosing the right platform for the problem domain at hand is not a trivial task. The choice is closely connected with the methodology adopted.

Follows a short description of some platforms:

Jade framework is probably the most used agent-oriented middleware. Is an open source distributed middleware system, compliant with FIPA specifications, that implements both white and yellow pages, agent mobility, ontologies and content languages, among other features. JADE does not provide, however, direct support to the development of BDI agent architectures.

Jadex is a software framework for the development of goal-oriented agents following the BDI model. Since JADE platform does not allow direct implementation of this model, Jadex, using JADE, allows the creation of rational agents. Jadex agents have two main components: an agent definition file (ADF), coded in XML, and Java code. Jadex BDI metamodel is specified in XML Schema.

Jason is an interpreter of the an extended version of AgentSpeak(L), allowing agents to be distributed over the net using Simple Agent Communication Infrastructure (SACI). Jason is available as open source and uses jEdit (http://www.jedit.org) as IDE.

JACK™ is a commercial agent platform, which uses syntactic and semantic extensions of Java that allows the implementation of BDI agents.

The use of an open source platform is preferable. Moreover, the compliance with FIPA specifications is important to allow interoperability of the systems. JADE platform provides those and other features.

Table 3 presents some platform characteristics.

Table 3: Platform classification.

Platform	Open source	BDI	Compliance	White & yellow pages
JADE	Yes	No	FIPA	Yes
Jadex	Yes	Yes	FIPA*	Yes*
Jason	Yes	Yes	KQML	No
JACK™	No	Yes	FIPA	Yes

* with JADE

3.3 Ontologies and Languages

Communication is a valuable tool for agents to interact, exchange information and request services. At the present, Ontology Web Language (OWL) is the language of the Semantic Web that is being standardized by the World Wide Web Consortium.

An agent platform must allow the use of content language (e.g. FIPA-SL Content Language Specification), and communication languages (e.g. FIPA-ACL Agent Communication Language).

3.4 Simulation

Multi-Agent Based Simulation is considered the support of choice for the simulation of complex systems, replacing or integrating with other micro-simulation techniques, most of them object-oriented.

4 THE MODEL

The model proposed consists of a novel multi-agent system that manages the communications inside each vehicle and simulates the communications between each of them and the infrastructure. Inter-vehicle communications are managed by an agent-based module that simulates real wireless communications between vehicles, using the appropriate standards. To allow interoperability, the platform supporting the development of the proposed multi-agent system complies with FIPA specifications.

The architecture will be tested in the context of an intersection, where the management of communications and localization of the vehicles will have both a distributed and a centralized component. This option aims to provide simulation functionalities at the communication level that, in the reality, would be provided by the transmission media. Moreover, localization of hazardous situations (vehicles without communications, pedestrians) is better provided by centralized facilities.

The architecture proposed to the multi-agent system is depicted in Figure 1.

4.1 Multi-agent Architecture

A brief description of the main agents involved in the proposed architecture follows:

Network Simulator: The main function of the Network Simulator (NS) is to receive all communications between Communication Manager agents, and simulate the network transmission between them, considering the environment and the location of each one. Appropriate communication standards must be used by this agent, namely DSRC and CALM.

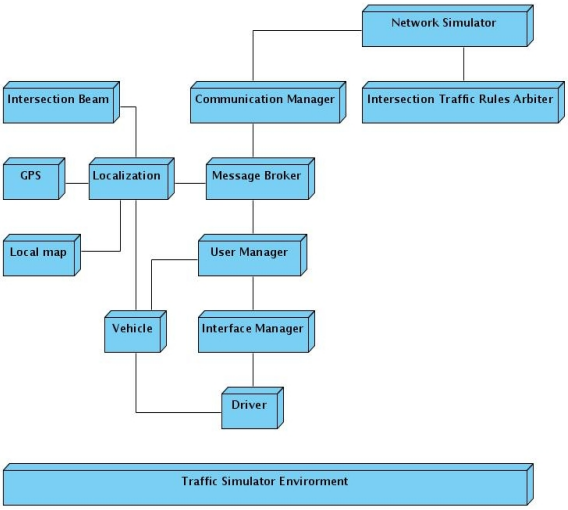


Figure 1: Multi-agent architecture.

Intersection Traffic Rules Arbiter (ITRA) must deal with intersection control of traffic, recording all traffic events and dealing with resolution of conflicts between User Managers (UM). With low traffic throughput, we may have a distributed control of traffic, where UM may agree with the priority of each other, always under ITRA supervision. As traffic flow grows, ITRA will have to validate all UM decisions, eventually overcoming some of them. In a high traffic flow scenario, all traffic rules decisions must be taken by ITRA, and vehicles become "data probes" of the centralized traffic rule management. Although this might seems contradictory with the choice of an agent-based system, real-time constraints impose the option presented above.

Communication Manager (CM) manages communications between the vehicle and external systems, such as the infrastructure and other vehicles. In both cases, NS is used as an intermediary, to simulate wireless network transmissions. Each vehicle communicates through its own CM.

Message Broker (MB) must manage all internal messages, and has the incumbency of filtering and its prioritization, ensuring that critical messages are dealt first by the appropriate agents. In this scheme, MB may delay low priority messages or, in some cases, even discard such messages.

User Manager (UM) main function has to do with decisions about the priority of the vehicle, with the agreement of all vehicles in its direct neighborhood, always under ITRA supervision. As stated before, as traffic flow grows, the decisions are taken by ITRA,

in a centralized manner. To avoid deadlocks, all the decisions must be taken with anticipation, allowing the forecast of possible deadlocks and its resolution before they actually occur.

Interface Manager (IM) agent deals with the selection of the most appropriate message interface to the driver, taken in account the type of message.

Localization agent determines the localization of the vehicle in the intersection map, using GPS data and an intersection beam signal, and compares its position with neighbor vehicles positions, periodically transmitted through wireless communications. This agent must decide whether the situation is critical, based on position and vehicle data, and warn UM in case of imminent danger.

Vehicle agent gathers vehicle data (e.g. speed, acceleration, brakes, steering) and feeds Localization agent with that information. UM receives also similar feedback. Moreover, this agent gets commands issued by Driver agent.

Driver agent deals with the control of the whole vehicle. It receives information, whether critical or not, via IM agent and responds accordingly to that information and the type of driver modeled. For that purpose, Driver agent maintains a driver type database. This agent issues commands to Vehicle agent directly and indirectly through IM.

Traffic Simulation Environment represents the environment where the agents evolve. One of its main functions is to provide communications between agents, in the platform level, allowing appropriate management of agents' percepts and actions. Graphical presentation of simulation results will also be directly connected with this component.

5 CONCLUSIONS AND FUTURE WORK

In this paper we propose an architecture in which the simulation and management of the inter-vehicle communications are integrated in the simulation of vehicles, in a hierarchical multi-agent environment. We also present a short survey of existing methodologies, platforms, ontologies and languages, and suggest some possible choices to allow appropriate system implementation.

MAS development using the appropriate methodology, the implementation of the solution in the selected platform, the validation of the process and final deployment will follow.

ACKNOWLEDGEMENTS

This work was supported by Institute of Systems and Robotics and Fundação para a Ciência e Tecnologia under contract NCT04:POSC/EEA/SRI/ 58016/2004.

REFERENCES

Avila, A. et al., (2005). A complete Simulator Architecture for Inter-vehicle Communication Based Intersection Warning Systems. In *Proc. 8th Int. IEEE Conf. Intelligent Transportation Systems*, Vienna, Austria.

Braubach, L. Pokahr, A., & Lamersdorf W. (2006). Tools and Standards, Chapter of Multi-agent Engineering - Theory and Applications in Enterprises. Springer Series: *Int. Handbooks on Information Systems*, Editors: S. Kirn, O. Herzog, P. Lockemann, O. Spaniol.

Dresner, K. and Stone, P., 2005. Multiagent Traffic Management: An Improved Intersection Control Mechanism. In *The Forth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 05)*, pp. 471-477, Utrecht, The Netherlands.

Eichler, S et al. (2005). Simulation of Car-to-Car Messaging: Analyzing the Impact on Road Traffic. In *Proc. of the 13th IEEE Int. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, vol. 00, pp. 507 – 510.

Hallé, S., Laumonier, J., & Chaib-draa, B. (2004). A Decentralized Approach to Collaborative Driving Coordination. In *IEEE Intelligent Transportation Systems Conference*, Washington, D.C., USA.

Li, Z., Wang, F., Miao, Q., & He, F. (2006). An Urban Traffic Control System Based on Mobile Multi-Agents. In *Proc. of IEEE Int. Conference on Vehicular Electronics and Safety*, Shanghai, China.

Padgham, L., & Winikoff, M. (2002). Prometheus: A Methodology for Developing Intelligent Agents. *In proceedings of the Third International Workshop on Agent-Oriented Software Engineering, at AAMAS'02*.

Piorkowski, M. et. al. (2006). Joint Traffic and Network Simulator for VANETs. *Poster Session, MICS 2006*, Zurich, Switzerland.

Sturm, A., & Shehory, Onn. (2003). A Framework for Evaluating Agent-Oriented Methodologies. In M. Winikoff P. Giorgini, B. Henderson-Sellers, editor, *Proc. of the Int. Bi-Conference Workshop on Agent-Oriented Information Systems, AOIS*.

Vogel, A., & Nagel, K. (2005). Multi-agent based simulation on individual traffic in Berlin, In *Proc. of the Int. Conference on Computers in Urban Planning and Urban Management (CUPUM'05)*, London.

Wooldridge, M. (2002). *An Introduction to multi-agent Systems*. John Wiley & Sons, Chichster, U.K.

DYNAMIC MODELING OF A 6-DOF PARALLEL STRUCTURE DESTINATED TO HELICOPTER FLIGHT SIMULATION

Nicolae Plitea, Adrian Pisla, Doina Pisla and Bogdan Prodan
Technical University of Cluj-Napoca, Constantin Daicoviciu 15, RO-400020 Cluj-Napoca, Romania
plitea@rdslink.ro, Adrian.Pisla@muri.utcluj.ro, Doina.Pisla@mep.utcluj.ro, Bogdan.Prodan@mep.utcluj.ro

Keywords: 6-DOF Parallel robot, Stewart platform, Dynamics, Modeling, helicopter flight simulation.

Abstract: The dynamic analysis is the basic element of the mechanical design and control of parallel mechanisms. The parallel robots dynamics requires a great deal of computing as regards the formulation of the generally nonlinear equations of motion and their solution. In this paper a solution for solving the dynamical model of a 6-DOF parallel structure destined to helicopter flight simulation is presented. The obtained dynamical algorithms, based on the kinematical ones, offer the possibility of a complex study for this type of parallel structure in order to evaluate the dynamic capabilities and to generate the control algorithms.

1 INTRODUCTION

Parallel robots have some advantages over serial ones such as higher stiffness, very good precision, high speeds and accelerations, a better weight over payload rate. However, kinematic and dynamic analysis of the parallel structures is much more complicated due to the constraints and singularities presence. Dynamic effects and their analysis are the basis of design specifications and advanced control of the parallel mechanical systems.

Many of the mechanics classical methods cannot be successfully applied for parallel robots.

There are essentially four methods:

1. Newton-Euler equations with impulse and momentum formulation or the D'Alembert equations;
2. Lagrange equations of first kinds with so-called Lagrange multipliers;
3. Lagrange equations of second kind with a minimum number of system coordinates;
4. Virtual work formulation including inertia forces.

In (Pierrot, 1990), a simplified method of determining the dynamic model of the HEXA robot in two steps is proposed.

(Codourey, 1991) proposes the first dynamic model that can be used to control the parallel DELTA robot in real time.

(Guglielmetti, 1994) presents the inverse dynamic model for the DELTA robot in the analytical form using the Newton's laws.

(Honneger, 1997) suggested the use of the dynamic equations in an adaptive control scheme for the Hexaglide robot, in which the pursuance errors are used on-line to correct the parameters used in dynamic equations.

(Stamper, 1998) present a dynamical model for a parallel structure with three degrees of freedom. This model was also generated with the simplications of the Codourey model.

(Tsai, 1999) present a dynamical model for a parallel structure with three degrees of freedom, using the virtual principle.

(Miller, 1992) presents the complete dynamic model of the DELTA robot based on Lagrange equations. In this case one considers that the robot bars possess inertia moments themselves.

To solve the dynamic model, (Merlet, 2000) uses Lagrange formulas. He has applied the direct and the inverse dynamic model for the *"left hand"*, to a prototype accomplished at INRIA based on a KPS kinematic chain structure.

(Pisla, 2000) propose a generalized dynamic model for parallel robots using first order Lagrange equations on the basis of equivalent masses.

(Guégan, 2002) presents a new solution for the dynamic model for the Orthoglide with Newton-Euler equations.

(Itul, 2003 and 2006) present a comparative study among various dynamical methods and

different solutions for solving the dynamical model for the guided in three points parallel robots.

Generally, in the above mentioned contributions, the experimental identification of dynamics for the parallel robots is restricted to simple models in combination with adaptive control algorithms.

Flight simulators are extensively used by the aviation industry and the military for pilot training, disaster simulation and aircraft development. The different types of flight simulators range from video games up to full-size cockpit replicas mounted on hydraulic, electric or electromechanical actuators (Nahon, 2000), (Andreev, 2000).

Contrary to popular belief, flight simulators are not used to train pilots how to fly aircraft. Today's modern simulators are used by commercial airlines and the military alike, to familiarize flight crews in normal and emergency operating procedures. Using simulators, pilots are able to train for situations that they are unable to safely do in actual aircraft. These situations include loss of flight surfaces and complete power loss etc. In all cases **dynamics** plays a very important role for the behaviour of parallel structures used as flight simulators.

It is widely acknowledged that the cues provided by a good visual system offer the bulk of realism in a flight simulator. It has also been shown that pilots consider the provision of consistent motion cues to add substantially to the realism of the simulation and to be helpful in the piloting task (Reid, 1988).

Thus, motion platforms are used on modern high-end flight simulators in order to provide motion cues consistent with the visual, auditory and control-feel cues to which the pilot is also subjected.

Within the motion-related subsystems, the most consistent research effort is over the washout subsystem which takes the motions generated by the aircraft equations including large displacements and filters to provide simulator motion-base commands. These commands must provide the pilot with realistic motion cues, while remaining within the simulator's motion limits (Nahon, 2000).

The paper is organized as follows:

Section 2 is dedicated to the description of the studied 6-DOF parallel structure;

Section 3 deals with the dynamic modeling using the virtual work principle;

Section 4 presents some simulations tests on a parallel robot;

The conclusions of this work are detailed in the section 5.

2 DESCRIPTION OF THE 6-DOF PARALLEL STRUCTURE

Taking into consideration the imposed requirements for a flight simulator, which should have 6-DOF, it was chosen the family of type Stewart-Gough parallel structures (Figure 1).

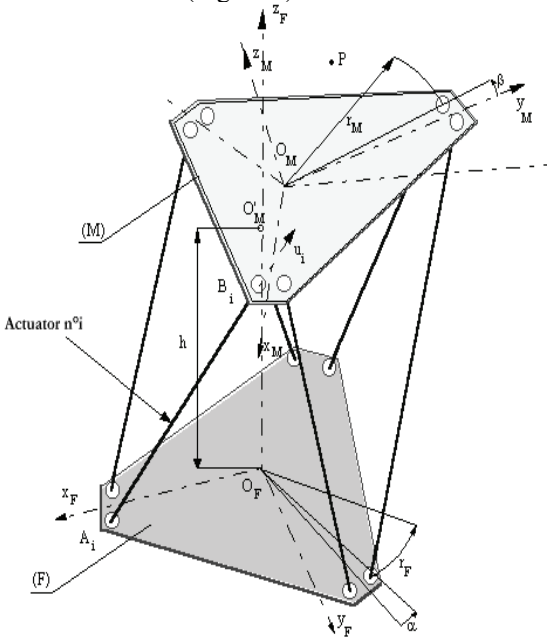


Figure 1: The 6-DOF parallel structure.

Generally, these parallel structures consist of six mobile arms, connected to the base and mobile platform through universal joints located at each end of the arm.

The mobile platform materializes the end element (end-effector). These kind of parallel structures are characterized by a robust mechanical structure and a high dynamic performance, a good ratio between the manipulated mass and the own mass.

The main difficulty results from the complexity in the motion control. Thus, the dynamics and its simulation is an important stage in order to test the capabilities of the robot and to develop the adequate control system.

2.1 Structural Considerations

For parallel mechanisms of F family the number of degrees of mobility is calculated with formula (Plitea, 2005):

$$M = (6-F)N - (5-F)C_5 - (4-F)C_4 - \\ -(3-F)C_3 - (2-F)C_2 - (1-F)C_1 \quad (1)$$

where:
M - mobility degree of the mechanism; F - mechanism family - the number of common constraints for all mechanism elements; N - number of mobile elements; C_i - number of "i" class joints; k = number of kinematic chains which connect the mobile platform to the base; n - number of elements of a kinematic chain for platform guidance for symmetric structures; c_i - number of "i" class joints of a kinematic chain for platform guidance.

The parallel robot mechanism family is:

$$F = 0 \quad (2)$$

In our case:

$$N = 13;\ C_5 = 6;\ C_3 = 12; \quad (3)$$

The mobility degree of the parallel mechanism will be:

$$M = 6N - 5C_5 - 3C_3 \\ M = 6 \quad (4)$$

2.2 Kinematic Modeling

In the case of inverse geometric problem, the actuation displacements are obtained with respect to the position and orientation of the mobile platform. An analytical solution could be obtained and applied in the control algorithms. For solving the inverse geometric problem, the transformation matrices method was used, using the Euler angles. The model has been already presented in (Pisla, 2007).

In the case of direct geometric problem the position and orientation of the mobile plate is calculated with respect to the actuation displacements. For solving the inverse geometric problem the transformation matrices method was used, using the Euler angles. The solution is a numerical one and the obtained nonlinear system could be computed by means of Newton-Raphson method (Pisla, 2007). The singularities of this paralle structure have been extensively discussed in (Pernkopf, 2002).

3 DYNAMIC MODELING OF THE 6 DOF PARALLEL ROBOT

The inverse dynamics consists in finding the relationships between the actuating joint forces τ_i, (i=1,2,…,6) and the motion laws for the manipulated object.

To study the dynamics, several simplifying hypotheses were adopted in the model:

-all joints are frictionless;

-the masses of guiding arms A_iC_i are neglected;

In Figure 2 the geometric parameters, the corresponding system coordinates and the forces are represented.

The used notations in the model are:

R_B - radius of fixed base; e_B, d_B - geometric parameters on the base; $\lambda_1 = \lambda_2 = 0$; $\lambda_3 = \lambda_4 = 120^0$; $\lambda_5 = \lambda_6 = -120^0$; r_p - radius of the working platform (WP); e_p, d_p - geometric parameters on the working platform; $\delta_1 = \delta_2 = 0^0$; $\delta_3 = \delta_4 = 120^0$; $\delta_5 = \delta_6 = -120^0$; m_p - mass of the working platform + the helicopter;

C = mass centre for the working platform + the helicopter; $C_{x'}$,$C_{y'}$,$C_{z'}$ - main central inertia axes; $I_{x'}$,$I_{y'}$,$I_{z'}$ - main inertia moments; oxyz - coordinate system of the mobile platform; OXYZ - fixed reference coordinate system; $OX_i'Y_i'Z_i'$ - coordinate system rotated with the angle λ_i with respect to the OXYZ system around the Z axis; $ox_iy_iz_i$ - coordinate system rotated with the angle δ_i with respect to the oxyz system around the z axis;

CXY'Z' - mobile reference system; its axes are parallel with the fixed coordinate system OXYZ axes; x_c, y_c, z_c - the coordinates of the mass centre C with respect to the oxyz system; X_c, Y_c, Z_c = the coordinates of the mass centre C with respect to the OXYZ fixed system of the robot.

In the inverse dynamic model, the input data are:

$$x_C = x_C(t),\ y_C = y_C(t),\ z_C = x_C(t), \\ \psi = \psi(t),\ \theta = \theta(t),\ \varphi = \varphi(t)$$

The actuation forces should be computed.

$$\tau_1 = \tau_1(t),\ \tau_2 = \tau_2(t),\ \tau_3 = \tau_3(t), \\ \tau_4 = \tau_4(t),\ \tau_5 = \tau_5(t),\ \tau_6 = \tau_6(t)$$

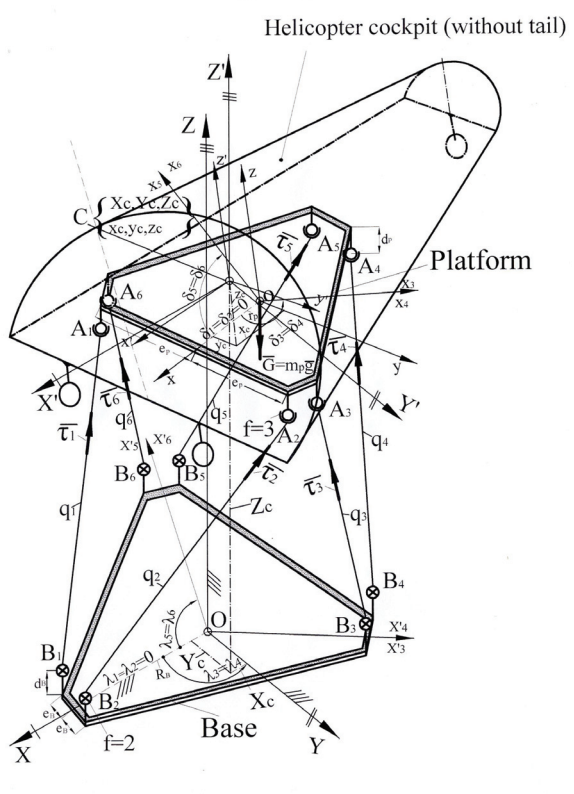


Figure 2: Dynamic modeling of the parallel robot.

The algorithm for solving the inverse dynamic model is presented as follows.

$$e_{Bi} = (-1)^i e_B, \qquad i = 1,2,....,6 \tag{5}$$

The coordinates of B_i points with respect to the $OX'_iY'_iZ'_i$ are:

$$X'_{Bi} = R_B,\ Y'_{Bi} = e_{Bi},\ Z'_{Bi} = d_B,\ i = 1,2,...,6 \tag{6}$$

The coordinates of B_i points with respect to the OXYZ are:

$$\begin{bmatrix} X_{Bi} \\ Y_{Bi} \\ Z_{Bi} \end{bmatrix} = \begin{bmatrix} C\lambda_i & -S\lambda_i & 0 \\ S\lambda_i & C\lambda_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X'_{Bi} \\ Y'_{Bi} \\ Z'_{Bi} \end{bmatrix},\ i = 1,2,...,6 \tag{7}$$

$$e_{pi} = (-1)^{i-1} e_p, \quad i = 1,2,...,6 \tag{8}$$

$$x'_{Ai} = r_p, \quad y'_{Ai} = e_{pi}, \quad z'_{Ai} = -d_p, \qquad i = 1,2,...,6 \tag{9}$$

The coordinates of A_i points with respect to the $ox_iy_iz_i$ are:

$$\begin{bmatrix} x_{Ai} \\ y_{Ai} \\ z_{Ai} \end{bmatrix} = \begin{bmatrix} c\delta_i & -s\delta_i & 0 \\ s\delta_i & c\delta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_{Ai} \\ y'_{Ai} \\ z'_{Ai} \end{bmatrix},\ i = 1,2,...,6 \tag{10}$$

The coordinates of A_i points with respect to the OXYZ are:

$$\begin{bmatrix} X_{Ai} \\ Y_{Ai} \\ Z_{Ai} \end{bmatrix} = \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} + \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \begin{bmatrix} x_{Ai} - x_C \\ y_{Ai} - y_C \\ z_{Ai} - z_C \end{bmatrix},\ i = 1,2,...,6 \tag{11}$$

Using the relations (7) and (11) the joint coordinates are:

$$q_i = \sqrt{(X_{Ai} - X_{Bi})^2 + (Y_{Ai} - Y_{Bi})^2 + (Z_{Ai} - Z_{Bi})^2},\ i = 1,2,...,6 \tag{12}$$

We consider the rotations around the CX', CY', CZ' axes (axes parallel with the fix ones OX, OY, OZ):

A_1=Cθ Cφ	A_2=-Sφ	A_3= 0
B_1= Cθ Sφ	B_2= Cφ	B_3= 0
C_1=-Sθ	C_2=0	C_3= 1

(13)

Then, the angular speed $\bar{\omega}$ and the angular acceleration $\bar{\varepsilon}$ of the mobile platform may be determined with the following relations:

$$\begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \\ C_1 & C_2 & C_3 \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} \varepsilon_X \\ \varepsilon_Y \\ \varepsilon_Z \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \\ C_1 & C_2 & C_3 \end{bmatrix} \begin{bmatrix} \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\varphi} \end{bmatrix} + \begin{bmatrix} \dot{A}_1 & \dot{A}_2 & \dot{A}_3 \\ \dot{B}_1 & \dot{B}_2 & \dot{B}_3 \\ \dot{C}_1 & \dot{C}_2 & \dot{C}_3 \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} \tag{15}$$

For the rotations around the fix axes CX', CY', CZ' the corresponding cosines may be determined using the following equations:

$$\begin{array}{lll} \alpha_1 = c\theta c\varphi & \alpha_2 = -c\psi s\varphi + s\psi s\theta c\varphi & \alpha_3 = s\psi s\varphi + c\psi s\theta c\varphi \\ \beta_1 = c\theta s\varphi & \beta_2 = -c\psi c\varphi + s\psi s\theta s\varphi & \beta_3 = -s\psi c\varphi + c\psi s\theta s\varphi \\ \gamma_1 = -s\theta & \gamma_2 = s\Psi c\theta & \gamma_3 = c\Psi c\theta \end{array} \quad (16)$$

The inertia moments are:

$$\begin{cases} I_{X'} = I_X\alpha_1^2 + I_Y\alpha_2^2 + I_Z\alpha_3^2 \\ I_{Y'} = I_X\beta_1^2 + I_Y\beta_2^2 + I_Z\beta_3^2 \\ I_{Z'} = I_X\gamma_1^2 + I_Y\gamma_2^2 + I_Z\gamma_3^2 \\ I_{X'Y'} = I_{Y'X'} = -I_x\alpha_1\beta_1 - I_y\alpha_2\beta_2 - I_z\alpha_3\beta_3 \\ I_{Y'Z'} = I_{Z'Y'} = -I_x\beta_1\gamma_1 - I_y\beta_2\gamma_2 - I_z\beta_3\gamma_3 \\ I_{Z'X'} = I_{X'Z'} = -I_x\gamma_1\alpha_1 - I_y\gamma_2\alpha_2 - I_z\gamma_3\alpha_3 \end{cases} \quad (17)$$

Then, the actuation forces τ_i are obtained:

$$\left(I_P^T\right)^{-1}\left\{M\ddot{X}_P + C\dot{X}_P + T_C^g\right\} = \tau \quad (18)$$

where

$$I_P^T = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \quad (19)$$

$$M = \begin{bmatrix} m_p & 0 & 0 & 0 & 0 & 0 \\ 0 & m_p & 0 & 0 & 0 & 0 \\ 0 & 0 & m_p & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{11} & a_{12} & a_{13} \\ 0 & 0 & 0 & a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 & a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (20)$$

$$\ddot{X}_p = \begin{bmatrix} \ddot{X}_C \\ \ddot{Y}_C \\ \ddot{Z}_C \\ \ddot{\Psi} \\ \ddot{\theta} \\ \ddot{\varphi} \end{bmatrix} \quad C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{11} & b_{12} & b_{13} \\ 0 & 0 & 0 & b_{21} & b_{22} & b_{23} \\ 0 & 0 & 0 & b_{31} & b_{32} & b_{33} \end{bmatrix} \quad (21)$$

$$\dot{X}_P = \begin{bmatrix} \dot{X}_C \\ \dot{Y}_C \\ \dot{Z}_C \\ \dot{\Psi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} \quad T_g^C = \begin{bmatrix} 0 \\ 0 \\ -m_p g \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \end{bmatrix} \quad (22)$$

In (19)-(21): $a_{11}, \cdots, a_{33}$ depend on the platform inertia moments $I_{X'}, I_{Y'} I_{Z'} I_{X'Y'} I_{Y'Z'} I_{Z'X'}$ and $A_i, B_i, C_i \; i = 1..3$; $b_{11}, \cdots, b_{33}$ depend on the platform inertia moments $I_{X'}, I_{Y'} I_{Z'} I_{X'Y'} I_{Y'Z'} I_{Z'X'}$ and $\dot{A}_i, \dot{B}_i, \dot{C}_i \; i = 1..3$; $C_{11}, \cdots, C_{66}$ depend on the direction cosines for the platform, the coordinates of points $A_i \; i = 1..6$ and the coordinates of the platform mass center $C_i \; i = 1..6$.

4 SIMULATION TESTS

The achieved kinematic and dynamic algorithms have been implemented in the developed simulation system (Pisla, 2005), (Pisla, 2007). It consists of five main modules: Kinematics; Singularities; Workspace; Trajectory, Dynamics. Within the simulation system the virtual graphical model was created, the 3D functional model allows the designer to understand its functionality (Figure 3).

The geometric parameters can be modified within the 3D modeling software influencing the simulation environment. The assembly relations between the parts, subassemblies and between parts and subassemblies can be also modified. These facilities enable the possibility to develop complex relations between the shape of the workspace, links and geometrical dimensions in order to optimize the parallel structure.

The parallel structure parameterization enables the development of the geometric optimization and the robot workspace shape. The obtained results are useful for the designers in understanding the workspaces characteristics distribution and parallel robots optimization.

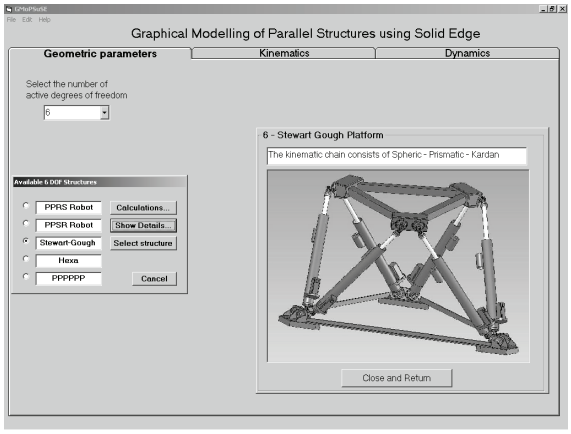


Figure 3: Simulation program for a 6-DOF parallel structure.

The presented simulation system enables the motion visualization in a modular manner valid for virtually any structure of parallel robot, introducing the kinematic and dynamic models over the virtual robot. The introduction of extra conditions related to any component is possible with a relative small number of actions. By using the graphical interface presented in Figure 3, the facilities of the simulation software enable to develop a complex study about the robot kinematics and dynamics in order to optimize the parallel structure.

5 CONCLUSIONS

In this paper a solution for solving of the inverse dynamics for a 6-DOF parallel robot conceived for a helicopter simulator has been presented. The dynamic model derived through virtual work principle has a compact form and offer the possibility of a more complex dynamic study in order to evaluate their dynamic capabilities and to generate innovative control algorithms.

ACKNOWLEDGEMENTS

This research was financed from the research grants awarded by the Romanian Ministry of Education Research and Youth.

REFERENCES

Pierrot, F., M. Uchijama, , P. Dauchez, Fournier, A., 1990, *A New Design of a 6-DOF Parallel Robot*, Journal of Robotics and Mechatronics, 2: 92-99.

Codourey. A., 1991. *Contribution a la Commande des Robots Rapides et Precis. Application au robot DELTA a Entrainement Direct.* These a l'Ecole Polytechnique Federale de Lausanne.

Guglielmetti, P. and Longchamp, R., 1994. A Closed Form Inverse Dynamics Model of the DELTA Parallel Robot. In *the Symposium on Robot Control*, pages 51-56, Capri, Italia.

Stamper, R. and Tsai, L.W., 1998. Dynamic Modeling of a Parallel Manipulator with Three Translational Degrees of Freedom. DETC98/MECH-5956, in *ASME Design Engineering Technical Conference,* Atlanta, GA.

Honneger, M. Codourey, A. and Burdet, 1997. E. Adaptive control of the Hexaglide, a six d.o.f. parallel manipulator. In *IEEE Int. Conf. On Robotics and Automation,* pages 543-548, Albuquerque.

Tsai, L.-W., 1999. *Robot Analysis, the Mechanics of Serial and Parallel Manipulators*, Wiley.

Guégan, S. and Khalil. W., 2002. *Dynamic Modeling of the Orthoglide.* Advances in Robot Kinematics (J.Lenarcic and F. Thomas, Ed.). Kluver Academic, Publication, Netherlands, 387-396.

Geng Z..and Haynes, L.S., 1992. *On the dynamic model and kinematic analysis of a class of Stewart platforms.* Robotics and Autonomous Systems, 9:237-254.

Liu, K., 1993. *The singularities and dynamics of a Stewart platform manipulator.* Journal of Intelligent and Robotic Systems, 287-308.

Miller, K. and Clavel. R., 1992. *The Lagrange-Based Model of DELTA-4 Robot Dynamics.* Robotersysteme**,** 8:49-54.

Merlet. J.-P., 2000. *Parallel robots*. Kluver Academic Publisher.

Pisla, D., Kerle, H., 2000. Development of Dynamic Models for Parallel Robots with Equivalent Lumped Masses. *In 6th International Conference on Methods and Models in Automation and Robotics,* , pages 637-642, Międzydroje, Poland.

Itul, T. and Pisla, D., 2003. Comparative Study between D'Alembert Principle and Lagrange Formulation for Guiding in three Points Parallel Robot Dynamic Analysis. In *14th International Conference on Control Systems and Computer Science*, Politehnica Press, Bucharest, 1:100-105.

Itul, T., Pisla, D. and Pisla, A., 2006. On the Solution of Inverse Dynamics for 6-DOF Robot with Triangular Platform, in *1st European Conference on Mechanism Science, EUCOMES,* ISBN 3-901249-85-0 (on CD), Obergurgl, Austria, February 21–26.

Kovecses, J., Piedboeuf and J.C., Lange, C., 2002. Methods for Dynamic Models of Parallel Robots and Mechanisms, in *the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators,* pages 339-347, Quebec.

Reid L.D., Nahon M.A., 1988. *Response of Airline Pilots to Variations in Flight Simulator Motion Algorithms*, Vol. 25, No. 7, 639-646.

Nahon M.A., Gosseli, 2000. *A comparison of flight simulator motion – base architectures*, Journal of Mechanical Design, Volume 122.

Andreev A. N., Danilov A. M.. *Information models for designing, conceptual broad-profile flight simulators,* Measurement Techniques, Vol 43, No. 8.

Plitea, N., Hesselbach, J., Pisla, D., Raatz, A., Vaida, C., Wrege, J., Burisch, A., 2006. *Innovative Development Of Parallel Robots And Microrobots*, Acta Tehnica Napocensis, Series of Applied Mathematics and Mecanics, no. 49, vol. 5, pp. 15-26.

Pisla, A., Plitea, N., Prodan, B., 2007. Modeling and simulation of parallel structures used as flight simulators, *in Proc of TMT2007, Tunisia*.

Pernkopf, F., Husty, M.L., 2002. Singularity analysis of spatial Stewart-Gough platforms with planar base and platform, *In Proc. ASME Design Eng. Tech. Conf.* Montreal, Canada, September 30 October 2.

Pisla, D. L, 2005. *Modelarea cinematica si dinamica a robotilor paraleli,* Editura Dacia Cluj-Napoca.

RACBOT-RT: ROBUST DIGITAL CONTROL FOR DIFFERENTIAL SOCCER-PLAYER ROBOTS

João Monteiro and Rui Rocha
ISR – Institute of Systems and Robotics, Department of Electrical and Computer Engineering
University of Coimbra, 3030-290 Coimbra, Portugal
jmonteiro@alunos.deec.uc.pt, rprocha@deec.uc.pt

Keywords: Digital control, mobile robots, non-holonomic, Lyapunov stability convergence, robot soccer.

Abstract: In the field of robot soccer, mobile robots must exhibit high responsiveness to motion commands and possess precise pose control. This article presents a digital controller for pose stability convergence, developed to small-sized soccer robots. Special emphasis has been put on the design of a generic controller, which is suitable for any mobile robot with differential kinematics. The proposed approach incorporates adaptive control to deal with modeling errors and a Kalman filter which fuses odometry and vision to obtain an accurate pose estimation. Experimental results are shown to validate the quality of the proposed controller.

1 INTRODUCTION

This paper describes the work that is being done in the field of digital control and real time systems for mobile robots within the RACbot-RT M.Sc. project. Many approaches for differential control of mobile robots have been presented. For instance, (A. Gholipour, 2000) presents a generic controller where pose estimation is extracted from the robot's kinematics, and an adaptive control block is introduced to deal with modelling errors. In (Y. Kanayama, 1990), a Lyapunov based nonlinear kinematic controller is presented where the influence of the control parameters is studied, without giving emphasis to modeling errors. The present approach brings together the simplicity of the Lyapunov mathematical laws, the adaptive control concept to deal with modeling errors and proper fusion of two sensorial data – vision and odometry – for robust pose estimation (T. Larsen, 2000).

2 ROBOT DYNAMIC MODEL

Based on the Lagrange's mathematical modelation of mechanical systems, on (A. Gholipour, 2000), and considering $G(q) = C(q,\ddot{q}) = 0$, the dynamic equations of the mobile robot can be written as

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{h} \end{bmatrix} = \quad (1)$$

$$\frac{1}{R}\begin{bmatrix} \cos(h) & \cos(h) \\ \sin(h) & \sin(h) \\ L & -L \end{bmatrix} . \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} sin(h) \\ cos(h) \\ 0 \end{bmatrix} \lambda, \quad (2)$$

where τ_1 and τ_2 are the left and right motor torques respectively, m and I are the robot's mass and inertia, R is the wheel's radius and L is the line distance between the two wheels. The non-holonomic restriction is deduced from (1) and given by the equation

$$\dot{x}\sin(h) - \dot{x}\cos(h) = 0, \quad (3)$$

from where it is imposed that a non-holonomic mobile robot can only move in the direction normal to the axis of the driving wheels.

3 TRAJECTORY DEFINITION AND ROBOT KINEMATICS

Our 2D path planner defines a trajectory as a time variant pose vector represented in the playing field, which has its own global cartesian system defined. The robot in the world possesses three degrees of freedom, which are represented by the actual pose vector

$$q = \begin{bmatrix} x \\ y \\ h \end{bmatrix}, \quad (4)$$

where x, y are the robot's coordinates and h is its heading. The latter is defined positively in the counter-clockwise direction, beginning at the positive xx axis. The state q_0 is denoted as the zero pose state $(0,0,2n\pi)$, where n is an integer value. Since the robot is capable of moving in the world, the pose q is a function of time t. The movement of the robot is controlled by its linear and angular velocities, v and ω respectively, which are also functions dependent of t. The robot's kinematics is defined by the following Jacobian matrix

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \end{bmatrix} = \dot{q} = Jp = \begin{bmatrix} cos(h) & 0 \\ sin(h) & 0 \\ 0 & 1 \end{bmatrix} q, \qquad (5)$$

where the velocity matrix is defined by

$$p = \begin{bmatrix} v \\ w \end{bmatrix} \qquad (6)$$

This kinematics is common for all non-holonomic robots.

4 POSE ERROR

To implement the controller, two pose vectors need to be defined: the actual pose of the robot already represented in (4), and the desired pose vector represented by

$$q_d = \begin{bmatrix} x_d \\ y_d \\ h_d \end{bmatrix}, \qquad (7)$$

which, by definition, is the target pose for the robot to achieve at the end of its movement. We will define the pose error q_e as the transformation of the reference pose q_d to the local coordinate system of the robot with origin (x_c, y_c), where the actual robot's abscissa is given by h_c's amplitude. Such transformation is the difference between q_d and q_c,

$$q_e = \begin{bmatrix} \dot{x_e} \\ \dot{y_e} \\ \dot{h_e} \end{bmatrix} = \begin{bmatrix} cos(h_c) & sin(h_c) & 0 \\ -sin(h_c) & cos(h_c) & 1 \\ 0 & 0 & 1 \end{bmatrix} .(q_d - q_c) \qquad (8)$$

One can easily see that if $q_d = q_c$, the pose error is null, being this the ideal final state.

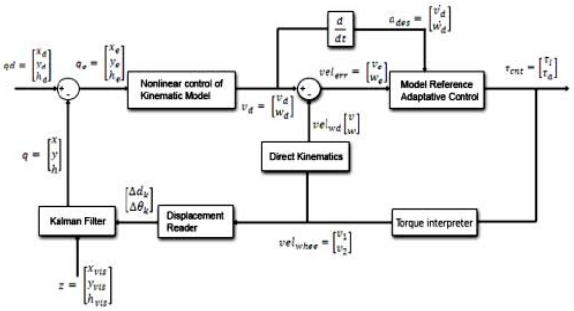


Figure 1: Control scheme.

5 DIGITAL CONTROLLER DESIGN

The controller is designed in three parts. In the first, kinematic stabilization is achieved using nonlinear control laws. For the second, the acceleration is used for exponential stabilization of linear and angular velocities. The uncertainties related with the robot's physical structure modeled parameters are compensated using adaptive control. For the final part, pose estimation is made fusing odometry and vision by means of a Kalman filter. The latter was designed in a way that independence of the mathematical system's model is achieved.

The developed approach is depicted in Fig. 1. It is a feedback controller, in which the input state is the desired robot's pose $[x_d\ y_d\ h_d]'$. At its output, proper update of the torques for each wheel is done to fulfill the controller's objective. Next, we will explain the relevant blocks.

5.1 Pose Error Generator

The error dynamics is written independently of the inertial (fixed) coordinate frame by Kanayama transformation. Expanding (8), we have

$$q_e = \begin{bmatrix} x_e \\ y_e \\ h_e \end{bmatrix} = \begin{bmatrix} \cos(h) & \sin(h) & 0 \\ -\sin(h) & \cos(h) & 1 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x_d - x \\ y_d - y \\ h_e - h \end{bmatrix} \qquad (9)$$

which will compose the pose error vector.

5.2 Nonlinear Kinematic Controller

Lyapunov based nonlinear controllers are very simple and yet, at the same time, very successful in kinematic stabilization. So, bringing together the concepts simplicity and functionality, the Lyapunov stability theorem proved to be of great utility for this project. Based

on such theorem, the deduced equations for the desired linear and angular velocities of the platform are,

$$v_d = v_r \cos(h_e) - K_x x_e \quad (10)$$

$$w_d = w_r + K_y v_r y_e + K_h \sin(h_e), \quad (11)$$

where K_x, K_y and K_h are positive constants. By La Salle's principle of convergence and proposition 1 of (Y. Kanayama, 1990), the null pose state q_0 is always an equilibrium state if the reference velocity is higher than zero ($v_r > 0$). This way, we can have three weighting constants for the pose error, without interfering in the overall pose stability of the robot.

5.3 Model Reference Adaptive Control

The motivation to include this block comes from the need to alter the control laws used by the controller for it to cooperate with parameter uncertainties. Based on (A. Gholipour, 2000), one can extract the adaptation rules for the linear velocity,

$$\frac{d\theta_1}{dt} = -\varepsilon_1 e\dot{v}_d \Leftrightarrow \theta_1 = \int -\varepsilon_1 e\dot{v}_d dt \quad (12)$$

$$\frac{d\theta_2}{dt} = -\varepsilon_2 e\dot{v}_d \Leftrightarrow \theta_2 = \int -\varepsilon_2 e\dot{v}_d dt. \quad (13)$$

where v_d is the desired linear velocity, and e the velocity error. The parameters ε_1 and ε_2 are manually tuned for best performance achievement.

Identically, similar rules for the angular velocity can be found.

5.4 Kalman Filter

A differential robot with odometry system as in our case, is equipped with an encoder in each motor. An angular displacement of α radians on the rotor corresponds to a performed distance d on the periphery of the wheel, and subsequently to an encoder count. The distance is given by $d = k\alpha$ with $k = \frac{1}{r}$, where r is the wheel radius. If the robot's movement is assumed to be linear, the distances d_1 and d_2 performed by the left and right wheels respectively, can be transformed in linear and angular displacements. For a particular sample instant, we have:

$$\Delta d_k = \frac{d_{1,k} + d_{2,k}}{2} \;;\; \Delta h_k = \frac{d_{1,k} - d_{2,k}}{b}. \quad (14)$$

The robot's coordinates referenced on the world's coordinates can be determined by the following equations:

$$X_{k+1} = X_k + \Delta d_k \cos(h_k + \frac{\Delta h}{2}) \quad (15)$$

$$Y_{k+1} = Y_k + \Delta d_k \sin(h_k + \frac{\Delta h}{2}) \quad (16)$$

$$h_{k+1} = h_k + \Delta_k. \quad (17)$$

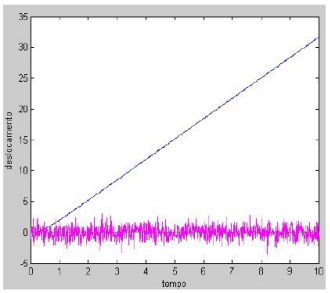


Figure 2: Filter test case results.

These coordinates constitute the state vector, and are observed by the vision coordinate vector z. These measurements can be described as a nonlinear function c of the robot's coordinates, which possesses an independent noise vector v. Defining the above equations as vector α and placing Δd_k and Δh_k in an input vector u_k, the robot can be modeled by the following equations

$$x_{k+1} = a(x_k, u_k, w_k, k) \quad (18)$$

$$z_k = c(x_k, v_k, k), \quad (19)$$

where $w_k \tilde{} N(0, Q_k)$ and $v_k \tilde{} N(0, r_k)$, being both not correlated, i.e., $E[w_l v_l^T] = 0$.
We can now design the extended Kalman filter, using the odometry-based system model:

$$\hat{x}_{k+1} = a(x_k, u_k, w_k, k) \quad (20)$$

$$P_{k+1} = A_k P_k A_k^T + Q_k \quad (21)$$

$$K_k = P_k C_k^T [C_k P_k C_k^T + R_k]^{-1} \quad (22)$$

$$\hat{x}_k = \hat{x}_k + K_k[z_k - C_k \hat{x}_k] P_k = [I - K_k C_k] P_k \quad (23)$$

The process noise is modeled by two Gaussian white noises applied on the two odometry displacement measurements Δd_k and Δh_k.
– *Filter simulation test case: Robot in* $x = 0$, $y = 0$, *heading* $= 0$, $\sigma^2_{vis} = 1$, $\sigma^2_{odo} = 1$
In this simulation, the displacement made by the robot in open loop will be indefinitely linear along the xx axis. Fig. 2 shows this situation, being the blue slope the displacement over xx, the red slope the displacement over yy and the green one, the robot's heading. The magenta slope represents the vision error. As we can see, the filter possesses little but visible sensitivity to vision noise.

6 ON-THE-FIELD RESULTS

– *Setpoint (final target position) command to (0,0)*
For this test, we send a *setPosition* command to $(0,0)$. In Fig. 3(A), we see a screenshot of the visualizer tool developed for the controller module. The robot

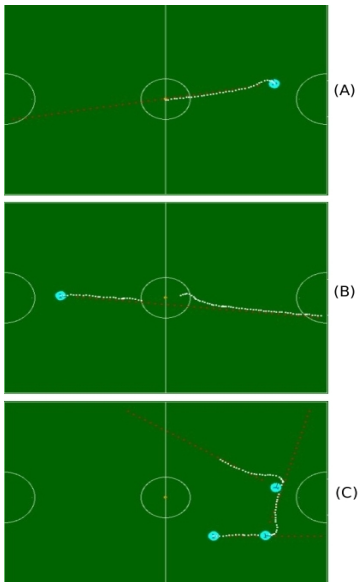


Figure 3: (A) – *SetPoint* command to $x = 0, y = 0$. (B) – *setVelocity* with $v = 0.3mps$ and desired $velocity\ angle = 0$. (C) – Sequence of *setVelocity* commands with $v = 0.3mps$ and $velocity\ angle = 1.3rad$.

accurately goes to the defined *setPoint*, but possessing a *yy* axis precision error of 1 to 2 centimeters maximum. This precision error exists because of two main causes. The first is the backward force exerted by the energy cable that feeds the robot in test environment[1]. The second is because of the defined tuned parameter for the influence of the robot's error over the *yy* coordinate – K_y. Tuning for near-zero error is possible but leads to a very *hard* control scheme in the presence of a physical disturbance, making the controller produce high overshoot for compensation. Since our robot is to walk on a field where collisions with other robots may be present, the revealed accuracy perfectly suits for our needs. For collision-free applications, where minimum physical errors exist and depending on the world's space, we can make the control *harder*, raising K_y.

– setVelocity (velocity vector) with v = 0.3mps and desired velocity angle = 0
For this test, the robot was subjected to extreme noise conditions.

Referring to Fig. 3(B), the robot is subjected to two disturbances done by blocking its left wheel, evident by the multiple white dots in the same place. Also, we blocked the color code of the robot used to be identified by the vision module for a while, so that no vision data was being received by the controller for it to estimate the actual pose. Controller's robustness is proved.

– Sequence of setVelocity commands with $v = 0.3mps$ and $velocity\ angle = 1.3rad$
In this final test (Fig. 3(C)), we sent a sequence of *setVelocity* commands to evaluate the control module's response in the presence of new velocity instructions. This test approaches from our real application target, the soccer game, where a high movement dynamic is required for the robot. Referring to Fig. 3(C), a velocity vector with zero desired angle is first sent, followed by two *setVelocitie*'s with the same module and 1.3 *rad* for the desired $velocity\ angle$. Finally, the robot is halted with a *halt* instruction. As we can conclude, the robot accurately executes the performed commands, evidencing the software module's robustness.

7 CONCLUSIONS

A controller for pose error elimination of a soccer-player robot was projected and its practical results have been shown. For the theoretical basis of the controller, particular interest was given to create a general scheme independent of the mathematical extraction of the test structure. On-the-field tests reveal that the projected approach is not only valid, but also robust. Despite the fact that the Real Time System has not already been implemented, the idea of placing it in the system is planned and will be present in the final version of this project.

REFERENCES

A. Gholipour, M. J. Y. (2000). Dynamic tracking control of nonholonomic mobile robot with model reference adaptation for uncertain parameters. University of Tehran.

T. Larsen, M. B. (2000). Location estimation for an autonomously guided vehicle using an augmented kalman filter to autocalibrate the odometry. Technical University of Denmark.

Y. Kanayama, Y. Kimura, F. M. T. N. (1990). A stable tracking control method for an autonomous mobile robot. IEEE.

[1]During tests, we prefer to have the robot constantly fed with energy instead of placing the batteries that discharge with time.

REMOTE ROBOT CONTROL AND HIGH AVAILABILITY

Silvia Anton, Florin Daniel Anton and Theodor Borangiu
University Politehnica of Bucharest, Dept. of Automation and Applied Informatics
313 Spl. Independentei, sector 6, RO-060032, Bucharest, Romania
anton@cimr.pub.ro

Keywords: Networked robotics, high availability, remote control, flexible manufacturing systems, robot vision.

Abstract: Nowadays production flows are modular, each module in the enterprise being specialized and used to achieve a particular task. In many cases the modules are connected and materials are sequentially processed in each module resulting a final, unique product or assembly. One typical such production module is a flexible cell/system using multiple robots. In this structure, providing continuous service for applications is a key component of a successful robotized implementing of manufacturing. High availability (HA) is one of the components contributing to continuous service provision for applications, by masking or eliminating both planned and unplanned systems and application downtime. A high availability solution in robotized manufacturing provides automated failure detection, diagnosis, application recovery, and node (robot controller) re integration. The paper describes a platform which is a software product designed to control and supervise multiple robot-vision controllers using remote connections with a number of Adept Technology V+ controllers configured to use a high availability implementation, either located in a local network or via Internet.

1 INTRODUCTION

In a robotized flexible manufacturing cell, robot (-vision) controllers are masters over local workstations or cells, because robot manipulators connect two important material flows: the *processing* flow and the *transportation* flow. One solution to integrate these two flows with on-line *quality control* in the manufacturing module, further networked with the design and planning modules, is to adopt a unified feature-based description of parts and assemblies, technological operations, geometric & surface quality control, grasping and manipulating (Tomas Balibrea, *et al*., 1997).

The system is configured for *high availability* . HA systems are a combination of hardware and software components configured to work together to ensure automated recovery in case of failure with a minimal acceptable downtime (Harris et. al., 2004).

2 THE STRUCTURE OF THE SYSTEM

The system is composed by the following applications (Figure 1):

The **Server Application (SA)**: Remote visual control and monitoring of multiple robot controllers from mobile and stationary matrix cameras.

- *Visual control*: the Server Application supports almost all V+ and AdeptVision program instructions and monitor commands.
- *Monitoring*: a Monitoring/Treatment scheme can be defined for each Client/Station. For each client a list of events and controller variables to be monitored according to a user-definable timing and precedence, and reacted at by user-definable actions/sequences can be specified in an Automatic Treatment Window.
- *Communication management*: the Server Application manages the communication with the robot controllers and cameras, transfers real-time images from the cameras observing the robot workplace and production environment, reports status information, stores in a database and displays images taken by the robot camera.

The **eClients Applications (eCA)**: Java applications running in web browsers. They provide portal services and the connection of networked production agents: image data and RV program / report management; real-time robot control and cell / workplace observation. The eCA are composed by two applications:

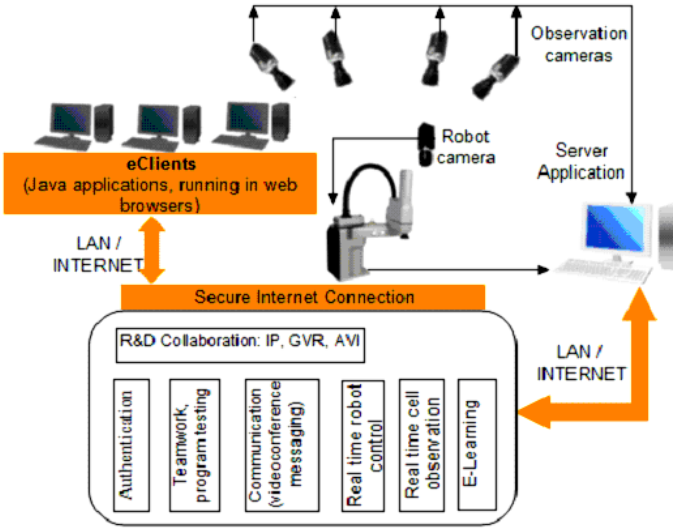


Figure 1: The System Structure.

- one application which has the function of retrieving the images from the observation cameras and display them in real-time and also gives the user the possibility to change the orientation and zoom factor of the cameras.
- the second application is a VNC client.

The VNC viewer is a web teleoperation application which can be executed into a web browser. The application connects to the Domino web server which makes a secure connection using a TCP/IP tunnel with a server having a private IP address, which cannot be accessed from internet but only using the Domino server.

The private IP machine has a VNC server that exports the display, and also the teleoperation application. Using the exported display the user can view and use the application as when the application runs on his own computer. The access is made using a username and a password, process managed by the Domino server.

3 ACCESSING THE SYSTEM

To have access to the system, a user must have a username and a valid password to enter in the system. First the user must access the portal site using a java aware browser (like Internet Explorer, Opera, Firefox, with the JRE installed).

The portal is structured in two zones:

- one zone is a public zone which contains all the documentation, tutorials courses and so on..., needed by users to learn how to use the system this part of the portal can be accessed by anyone.
- and a private zone where the access is based on username and password. The private zone gives access to the eClients for teleoperation purposes.

After entering the correct username and password, the user is allowed in the system and has access to a the teleoperation application which is a menu driven interface which allows him to interact with the system (see Figure 2).

The teleoperation application is composed by two windows:

A command window where the user can select the robot system which he want to control and issue commands from the command line or activate the vision window.

The robot stations are commanded using the command line and the menus. When a client is connected, the IP address is checked and if the client is accepted, the name attached to the IP address is added to a drop down list from which the user can select what client he wishes to command. When a client who has a video camera attached the VISION button is enabled and if it is pressed the VISION Window will open.

From the VISION window, vision commands can be issued by selecting the wanted actions from the menus. The most important functions are:

- selecting the physical and virtual cameras, and the virtual image buffers;
- selecting the display mode and the resolution;
- image acquisition;
- issuing primary operations;
- displaying the vision system status; training models;
- switches and parameters configuration for virtual camera set-up.

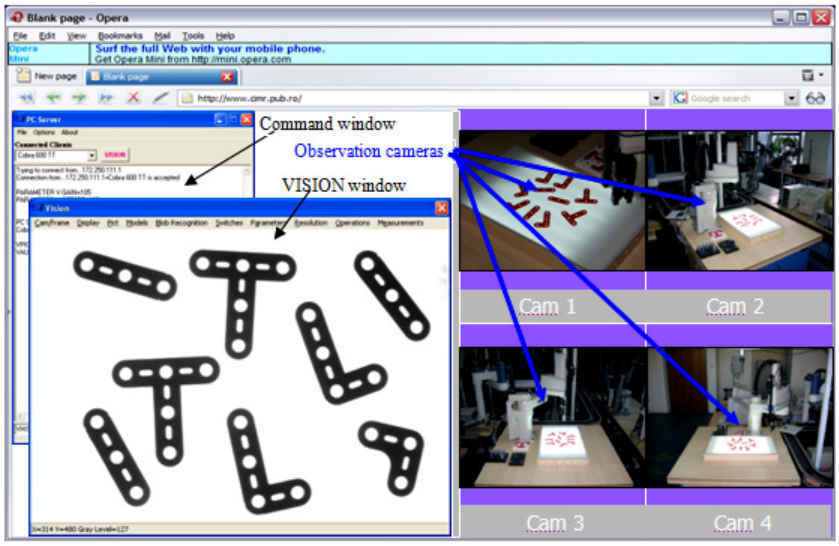


Figure 2: Accessing the system.

4 SOLUTION IMPLEMENTING FOR NETWORKED ROBOTS

In order to implement the solution on a network of robots, first a shared storage is needed, which must be reached by any robot controller from the cluster.

The file system from the storage is limited to NFS by the operating system of the robots. Five Adept robot manipulators were considered, each one having its own multitasking controller.

For the proposed architecture, there is no option to use a directly connected shared storage, because Adept robot controllers do not support a Fiber Channel Host Bus Adapter (HBA). Also the storage must be high available, because it is a single point of failure for the Fabrication Cluster (FC).

Due to these constraints, the solution was to use a High Availability cluster to provide the shared storage option (NFS Cluster), and another cluster composed by Adept Controllers which will use the NFS service provided by the NFS Cluster (Figure 3). The NFS cluster is composed by two identical IBM xSeries 345 servers, and a DS4100 storage.

The storage contains a volume named Quorum which is used by the NFS cluster for communication between nodes, and a NFS volume which is exported by the NFS service which runs in the NFS cluster. The servers have each interface (network, serial, and HBA) duplicated to assure redundancy (Anton *et al*., 2006; Borangiu *et al*., 2006).

There are three communication routes: the first route is the Ethernet network, the second is the Quorum volume and the last communication route is the serial line. If the NFS cluster detects a malfunction of one of the nodes and if this node was the node which served the NFS service the cluster is reconfiguring as follows:

1. The server which is still running writes in the Quorum volume which is taking the functions of the NFS server, then
2. Mounts the NFS volume, then
3. Takes the IP of the other server and
4. Starts the NFS service.

The Fabrication Cluster can be composed by at least two robot controllers (nodes) – *group leader* (GL) and a common node. The nodes have resources like: robot manipulators (with attributes like: collision detection, current robot position, etc...), serial lines, Ethernet adapter, variables, programs, NFS file system. The NFS file system is used to store programs, log files and status files. The programs are stored on NFS to make them available to all controllers, the log files are used to discover the causes of failure and the status files are used to know the last state of a controller.

In the event of a node failure, the production flow is interrupted. In this case, if there is a connection between the affected node and the group leader, the leader will be informed and the GL takes the necessary actions to remove the node from the cluster. The GL also reconfigures the cluster so the fabrication process will continue.

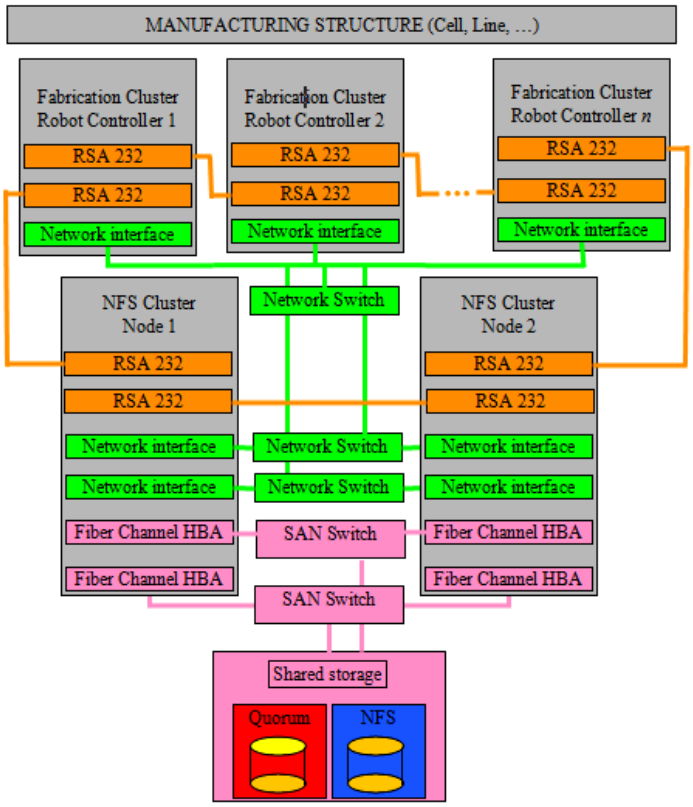


Figure 3: Implementing the high availability solution for the networked robotic system.

5 CONCLUSIONS

The project was started at the end of 2005 as part of the P.R.I.C. research program and is in the final stage of development.

The research project will provide a communication and collaboration portal solution for linking the existing pilot platform with multiple V+ industrial robot-vision controllers from Adept Technology located in four University Labs from Romania. This will allow teachers to train their student using robots and expensive devices which they do not dispose, and allow students to practice their skills using specialised labs without geographical barriers, and even from home. Also the portal will allow team training and research due to the messaging feature introduced by Domino.

The high availability solution presented in this paper is worth to be considered in environments where the production structure has the possibility to reconfigure, and where the manufacturing must assure a continuous production flow at batch level.

The advantages of the proposed solution are that the structure provides a high availability robotized work structure with a insignificant downtime.

The project is under development and can be accessed at: http://pric.cimr.pub.ro.

REFERENCES

Anton F., D., Borangiu, Th., Tunaru, S., Dogar, A., and S. Gheorghiu, 2006. Remote Monitoring and Control of a Robotized Fault Tolerant Workcell, *Proc. of the 12th IFAC Sympos. on Information Control Problems in Manufacturing INCOM'06*, Elsevier.

Borangiu, Th., Anton F., D., Tunaru, S., and A. Dogar, 2006. A Holonic Fault Tolerant Manufacturing Platform with Multiple Robots, Proc. of *15th Int. Workshop on Robotics in Alpe-Adria-Danube Region RAAD 2006*.

Harris, N., Armingaud, F., Belardi, M., Hunt, C., Lima, M., Malchisky Jr., W., Ruibal, J., R. and J. Taylor, 2004. *Linux Handbook: A guide to IBM Linux Solutions and Resources,* IBM Int. Technical Support Organization, 2nd Edition.

Tomas Balibrea, L.M., L.A. Gonzales Contreras and M. Manu (1997). *Object Oriented Model of an Open Communication Architecture for Flexible Manufacturing Control*, Computer Science 1333 - Computer Aided System Theory, pp.292-300, EUROCAST '97, Berlin.

SPECIAL SESSION ON SERVICE ORIENTED ARCHITECTURES FOR SMEROBOTS AND PLUG-AND-PRODUCE

CHAIRS:
J. NORBERTO PIRES
KLAS NILSSON
LUCA LACHELLO

TOWARDS A STANDARDIZED AND EXTENSIBLE MECHANISM FOR ROBOT DEVICE INTEGRATION

A XIRP-based Approach and Test Bed Implementation

Fan Dai and Joachim Unger
ABB Corporate Research, Wallstadter Str. 59, D-68526 Ladenburg, Germany
{fan.dai, joachim.unger}@de.abb.com

Keywords: Robot device integration, plug and produce, XML, XIRP.

Abstract: In industrial robot automation, the integration of intelligent peripheral devices becomes more and more important. But there is no standardized mechanism to setup the communication interface between them and to configure the usage of the device information for the robot applications. This makes the integration task often very tedious and time consuming. XIRP – the XML-based Interface for Robots and Peripherals is a recommendation that was published by the German standardization institute DIN in 2006. It specifies a standardized mechanism and the corresponding communication protocol for robot device integration. Our experiences with XIRP-based implementations have shown its big potential to support robotics PnP on the communication and configuration level. As one of the topics within the SMErobot™ project, we are working on further developments or the concept. This paper introduces our approach for a revised XIRP concept, and discusses our experiences made on test-bed implementations.

1 INTRODUCTION

Integration of peripheral devices like sensors to an industrial robot becomes more and more important because of the demands on more intelligent solutions. However, there is no standardized mechanism to achieve the integration. In most cases, proprietary interfaces and protocols are used that require a profound knowledge about the communication level and often requires an implementation of the interface on one of the devices. This often causes significant costs and time efforts and is one of the burdens especially for small and medium-sized enterprises for using industrial robots in their environments. There is a high demand to support "Plug and Produce" (PnP) integration. It should be as easy as plug-in a USB device to a personal computer, so that the devices are recognized, configured and ready to use.

Within the SMErobot™ project, robotics PnP is one of the major research topics (Nilsson et.al., 2005, Naumann et.al., 2006). In this project, several levels and aspects for Plug'n'Produce were identified as depicted in Figure 1.

On the application level, we can consider each device as a provider of certain functionalities, which we also could call as services. These services have to be combined to compose an application level service that can be used to carry out the desired task. In order to obtain the appropriate services, the devices, including any control units, must be configured accordingly. The single services must be assigned with the right parameters. Finally, on the communication level, the parties have to agree on which protocols are to be used. To achieve PnP, there must be a standardized mechanism for the configuration and for the establishment of the communication. The communication protocol must be also standardized. In this way, there will be no need to implement the communication interface for different devices each time when you get a new device, or connect an existing device to another control unit.

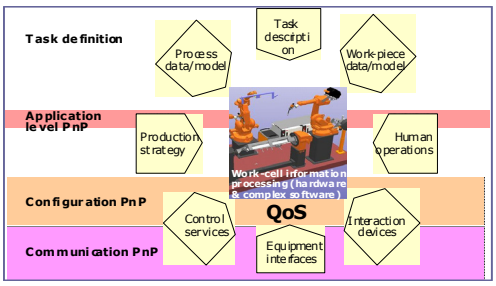


Figure 1: Aspects and Levels of Robotics PnP according to SMErobot™.

In this paper, we will focus on the lower levels, thus the communication and configuration levels.

Our approach is based on the general concept of XIRP – the XML-based Interface for Robots and Peripherals (VDMA 2006, Gauss et. al. 2006). XIRP is a recommendation that was published by the German standardization institute DIN in 2006. It specifies a standardized mechanism and the corresponding communication protocol for robot device integration. Our experiences with XIRP-based implementations have shown its big potential to support robotics PnP on the communication and configuration levels.

Within the European project SMErobot™ (Plug and Produce work package), the project partners are working corporately on a revised XIRP version, also defined as XIRP version 1.1 or XIRP+. This paper introduces our approach that was partly our contribution to XIRP+, and discusses our experiences made on test-bed implementations. Finally, we will describe our view on the integration of different communication concepts and standards to support robotics PnP.

2 THE APPROACH

2.1 Application Model

In an automation environment, there are basically two roles: control units and peripheral devices. This can be compared with the control points and the services in the context of UPnP. Depending on the application scenario and how the user configures the whole system, each device can theoretically take both of these roles. However, in a robot work cell, the control unit is usually the robot or the cell controller.

For the control unit, a device providing certain services can be seen as a logical unit with three types of functions:

- connection
- execution of commands
- exchange of data

This is independently from the complexity of a device. It can also consist of several physical devices that in their turn can be connected in any form to provide a more powerful service.

This view is common both for service-oriented or client-server communication models. There is no clear separation between both. However, we don't emphasize the concept of loosely-coupled client-server relationship. We use the client-server model and combine it with the discovery methods of service-oriented concepts.

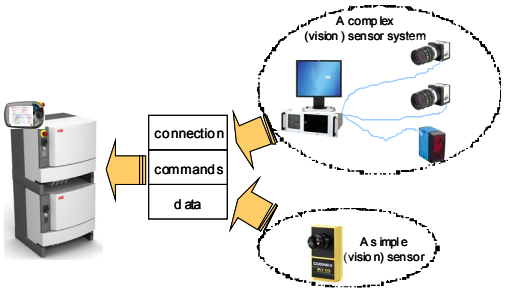


Figure 2: Relation between control units and devices.

In our client-server model, there is a defined relationship between two parties during runtime. One party, typically the robot controller, is a master of the application during the whole session. It provides a better overview of the coherences between services, and allows easier management of the logical structure in an automation world, especially in an industrial robot application. Service-oriented approaches have the potential for flexibility and upwards scalability, which can be beneficial for large scaled, distributed environments. But dealing with robotic cells for small and medium-sized enterprises (SME), it is important to have controllable structure, other than scalability. Furthermore, the real-time efficiency is also essential for robot-device communication.

2.2 Protocol Stack

Figure 3 shows the communication stack of XIRP devices according to the ISO/OSI reference model from ISO/IEC 7498-10.

Although XIRP does not specify Ethernet for the physical layer, it is recommended to use IP over Ethernet. The standardized protocols like TCP, UDP or HTTP can be used on the session layer to handle the reliable or unreliable transport of communication packages.

The message presentation uses XML, whereas XIRP specifies the basic structure of the messages, like SOAP does for Web services. But XIRP messages have less overhead so that it is much more efficient. Metadata as required in the SOAP messages are not required for XIRP, as XIRP presumes, that metadata are exchanged during the device configuration phase, if they are not available before that.

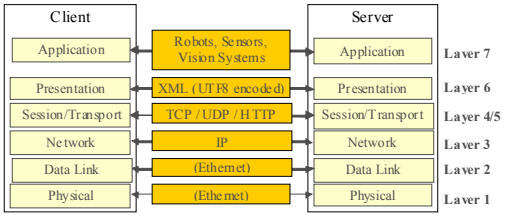


Figure 3: Protocol stack.

There are basically three types of messages that a client device can send to a server device: commands or requests for getting and setting property values. XIRP specifies the structure of these messages and the corresponding reply messages. On top of the basic specification, device type or application domain specific agreements can be defined.

2.3 Communication Channels

When connecting two devices, one device takes the role of the client and the other one the role of the server. The main communication channel (command channel) allows the client to "control" the server by sending command messages, which triggers the execution of an action (command) on the server device, or getting and setting the value of properties on the server.

To enable the PnP integration of devices, automatic discovery via UDP multicast is supported like in UPnP, but XML is used for the message representation without the SOAP overhead. Similarly, device and service descriptions and other files can be retrieved via HTTP. It is furthermore possible to transfer files from a client device to a server device via HTTP.

A client device can subscribe individual events by sending a subscription message to the server over the command channel. If the server device confirms the subscription, the client device will receive these events on the event channel. In addition a XIRP client can also send events to a server device. Before an event can be sent, the client announces it to the server via the command channel, so that the server can be prepared to receive the event on the event channel.

In robotics applications there is also demand for the periodical transmission of data on a defined rate and under some time constraints. This can be done via a periodic channel. Similar to event subscription, periodical data are also subscribed (for receiving) or registered (for sending) using appropriate messages on the command channel.

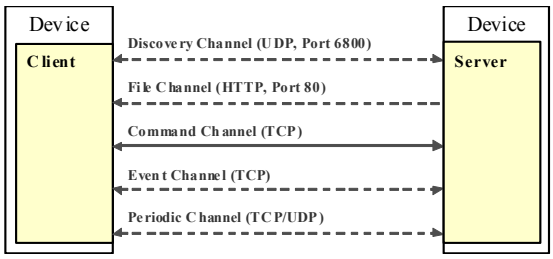


Figure 4: Communication channels.

Most robotics applications nowadays can be supported using these communication channels. For applications that have special requirements, we also allow the creation of additional channels via the command channel. These channels would use other standardized protocols. One example is the time synchronization channel that can use the PTP (Precision Time Protocol) protocol of IEEE 1588, to adjust the clocks in a distributed system and create a common time basis. A binary channel using a standardized binary transfer protocols could be also created, even when binary data can be embedded into XML messages using Base64 coding and transferred via the above mentioned standard channels too. These additional channels and the corresponding protocols have to be specified in the description of the devices, and can only be used, if both sides of a communication session support them.

2.4 Device Profiles

The definitions described in the XIRP specification are specified as mandatory or optional agreement. Mandatory definitions have to be implemented as specified. Optional agreements don't have to be supported, but if such a definition is implemented, it has to be done according to the XIRP specification. Furthermore, the definitions are grouped into so-called device profiles and organized into Basic Profile, Class Profiles and Custom Profiles as depicted in Figure 5.

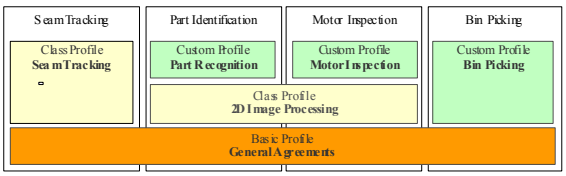


Figure 5: Device profiles.

The Basic Profile contains general agreements that are mandatory for all devices that claim conformity to the XIRP specification. It specifies the basic commands for the connection establishment and parameters for the control of common functions of XIRP-confirm devices. Also specified as general agreements are aspects like

- Schema and rules for description of devices
- Supported communication channels
- Basic structure of communication messages
- Message synchronization
- Termination of communication messages
- Language selection for comments
- Parameters for connection management

- Connection establishment
- Primitive data types
- Rules for extensions

To run applications, additional agreements on commands and parameters are needed. Those are specified in the class or custom profiles. Definitions in a class profile are only mandatory for devices that claim conformity to this device class, which could be smart cameras, laser trackers, complex vision systems, etc. Devices that conform to a certain class profile should be exchangeable without affecting the execution of an existing application, if no custom profile is used in addition. Otherwise, a smooth replacement of devices is not ensured.

Custom profiles can extend the general agreements and class profiles with additional commands and parameters that are needed to run a specific application. They do not represent a standardized specification, but have to conform to the specifications in the General Agreements. If an application can be implemented with existing device class profiles, no custom profile should be defined and used in addition.

Preliminary, a communication profile for a new device, or a set of device functions, which has not been defined in a class profile yet, can be defined as a custom profile and submitted to the XIRP working group as a proposal for standardization. For a certain time slot the preliminary profile will be put up for discussion and then released as a standardized class profile.

2.5 Protocol Schemas

Commands and data type definitions, as well as definitions for message structure are specified as XML schemas, which can be stored in one or several files. A device can refer to multiple schema files as long as they do not contain conflicting definitions. The benefit of using the standardized XML Schema format for these definitions is that the files can be interpreted with validating XML parsers and created with common XML tools.

2.6 Device Description

Each device must be supplied with a device description. It contains the general information about a device and is stored in a XML file. The contents are:

- The XIRP version
- the unique device identifier (or name)
- the type classification for the device
- the category classification for the device

Optionally, it can also contain:

- a textual description of the device
- the user interface URL of the device
- custom or device class schemes associated with the device
- the selection of languages that are supported by the device for the comments on commands and parameters in the configuration files
- The collection of communication channels that are supported by the device.

Following is an example for a device description:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Description Version="1.1">
   <Name>TestDevice</Name>
   <Vendor>MyVendor</Vendor>
   <Model>MyModel</Model>
   <Class>SmartCamera</Class>
   <Url>127.0.0.1</Url>
   <Schemes>
     <Schema Path="Camera.xsd" />
     <Schema Path="Calibration.xsd" />
     <Schema Path="Contour.xsd" />
   </Schemes>
   <Languages>
     <Language Code="en" />
     <Language Code="de" />
   </Languages>
   <Channels Url="127.0.0.1">
     <Discovery />
     <File Port="80" />
     <Command Port="3002" />
     <Event Port="3003" />
     <Cyclic Protocol="UDP" Port="3004"/>
   </Channels>
</Description>
```

The device description is an important element for PnP, and can be used to automatically establish connection and configure the communication and programming setting.

2.7 Plug'n'Produce

When talking about PnP, we mean the ability to use a device after plugged it onto the network, without the need of manually setting communication parameters or configure the interfaces. As indicated in previous sections, the key elements for PnP are: the device description, the corresponding schema definitions and the mechanism of connection establishment.

The following figure shows how PnP works in our approach.

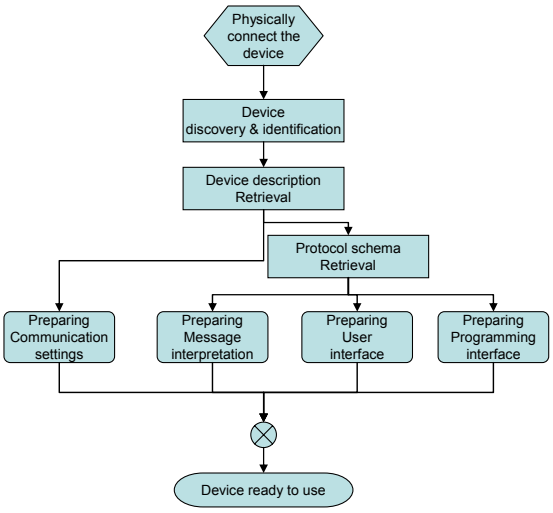


Figure 6: The PnP mechanism.

The discovery channel is the first mandatory functionality of a device that supports PnP. After physically connected the device to a control unit or to the network of the work cell, the device will announce it's presence with a broadcast message via the discovery channel. A control unit has to listen to this broadcast channel, if it wants to be aware of devices that are plugged-in. The other option is to actively detect new devices by sending a discovery message. Every PnP device would respond with a unicast message to this control unit. In this way, the amount of broadcast messages can be reduced to a minimum during runtime of the robot cell, because they are only sent as needed.

After device discovery, a control unit would be able to identify the server devices and check, which additional information are needed to establish the connection to these devices. First of all, the device descriptions must be made available to the control unit. A PnP enabled device should have the device description accessible via the file channel. Alternatively, there could be also a repository of device descriptions elsewhere. But the safest way to provide compatible and up-to-date device description is to provide it on the device itself.

With the device description, the control unit can be prepared to establish the connection to the server device. This means, the communication level PnP would be done. The device can be connected whenever it is needed for the application.

Configuration level PnP needs more information than directly provided in the device description file. But with the references specified in the device description, the control unit can further retrieve the schema files. Information provided with the schema allows the control unit to configure the interpretation of the communication messages. Also possible is to create or update the user interface for application programming based on the commands and data type definitions in a schema.

3 TEST-BED IMPLEMENTATIONS

Three test-bed implementations are described briefly in this section. They represent three scenarios with different requirements on robot-device communication, so that the functionalities of the different communication channels could be demonstrated and evaluated.

3.1 Smart Camera

As an example for relatively simple, but intelligent device, we used a smart camera from DVT (now owned by Cognex) of the type Legend 510. This camera has embedded image processing functions and Ethernet link on-board. It can be configured and programmed using a kind of scripting language via a PC. Afterwards, the camera can be used stand-alone.

The device description file and corresponding schema files are located on the camera, and are automatically downloaded to the robot system via the file channel (using HTTP). The robot system is then automatically configured so that the commands and variables defined in the schema file are accessible for robot programming.

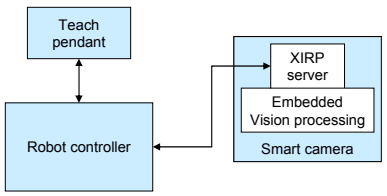


Figure 7: Smart camera test-bed.

In the sample application, a vision method for blob finding and analysis is implemented on the camera. It can be triggered from the robot system. Results like number of objects found, and size and position of each object can be read from the camera using the "GetOnce" command. This data is used to generate robot motion targets to pick up the objects.

The device and its properties as well as the vision object data are automatically mapped to variables in the robot programming language RAPID. They can be used like other RAPID variables on the teach pendant.

3.2 The Chess Robot

The ChessRobot scenario, which has been presented on the Hannover Fair 2007, was used for the evaluation of XIRP-based PnP as well. In this scenario, multiple XIRP server instances were connected to the robot controller, using command and event channels. The test-bed architecture is illustrated in Figure 8. Figure 9 shows the setup on the Hannover exhibit 2007.

The robot controller is the control unit in this scenario. It controls two robot arms, each of them playing two games – one against a visitor and the other one against each other. The boards for the games with visitors are observed by cameras. The moves of the visitors are recognized using a vision system, whereas a game engine checks the validity of the visitor moves and generates counter-moved for the robot. To test the concurrent communication with several server devices, we used three instances of XIRP server device running on three separate PCs.

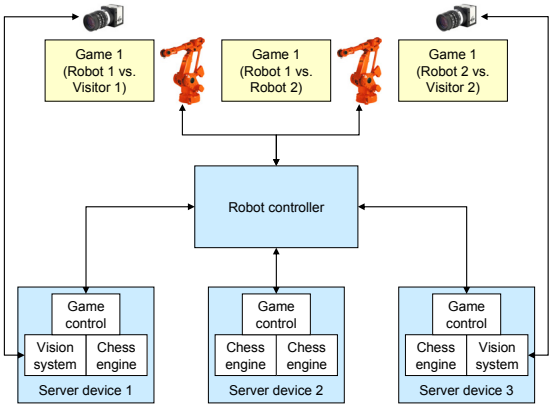


Figure 8: ChessRobot testbed.

Figure 9: ChessRobot on exhibition.

3.3 Safety Zone Supervision

Another test-bed application used cameras to supervise the safety zone in the robot workspace. A modified version of SIMERO (Gecks, Th., Henrich, D., 2004) runs on a Linux PC and analyses the spatial relation between human worker or moving obstacles and the robot arm. Depending on the spatial situation, it sends then adapted velocity data to the robot so that the robot moves always safely.

For this purpose, a periodical channel is used to transmit robot data to the vision system and the updated velocity data to the robot controller. The data exchange could be done in real-time with very small jitters. In addition, the robot data are assigned with timestamps so that correct mapping of spatial data is possible.

4 DISCUSSIONS AND OUTLOOK

We implemented our XIRP-based approach for three scenarios with different requirements. With the smart camera test-bed, it has been shown that even using scripting language, it is possible to implement XIRP server on a low-cost smart device. PnP worked with downloading the device description from the camera. With the ChessRobot test-bed, multiple server devices were connected to the robot controller. Also the functionality of event subscription could be demonstrated. The safety zone supervision test-bed further evaluated the subscription of periodical data for real-time communication.

These examples have shown that the implementation of XIRP protocol doesn't require much computation resources. The approach works both for simple intelligent devices and PC-based systems.

This approach is control unit centric, with clear controller-device relationship. This is also the typical case for SME scenarios. Even though, the XIRP-based architecture is also scalable to support scenarios with multiple clients and servers, we haven't tested our approach with large-scaled distributed environments. Because of the nature of this kind of client-server model, it has limitations with flexibility and scalability in terms of distributed, large-scaled environments.

When considering distributed environment without real-time constraints, Web services approaches have big potential. We are the opinion that a hybrid approach would better fulfil the different requirements rather than trying to

implement one protocol for all cases. Furthermore, we observe progresses in the standardization of communication protocols on the field device level like EtherCAT and OPC UA. Also considering that XIRP might be still too complex for simple I/O devices, it is proposed to develop an integration framework that supports different protocols.

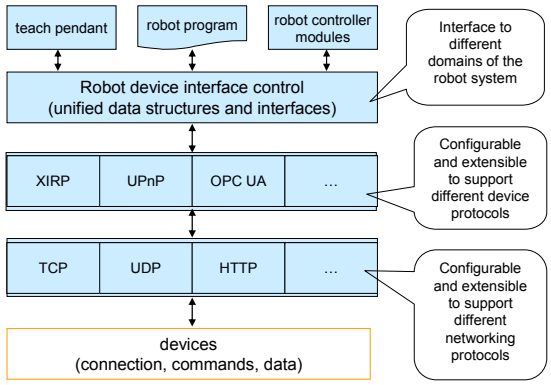


Figure 10: Device integration framework.

Certainly, a careful selection of the protocols should be done so that they really complement each other. Figure 10 only shows the principle and doesn't mean any preferences of protocol selection.

To have different kind of devices running different protocols concurrently with minimal implementation and processing overhead, we have to find the common features of the selected approaches. Also to be investigated are the boundaries of each protocol. Based on the investigations, recommendations for the users respectively application engineers can be worked out.

It is very hard, or even impossible to make a standard that fits to everyone's needs. But a selection of standards, a common understanding of the methods and some guidelines or recommendations would help the users, especially SME's to apply robotics PnP.

ACKNOWLEDGEMENTS

This work has been partly funded by the European Commission's Sixth Framework Program under grant no. 011838 as part of the Integrated Project SMErobot™.

REFERENCES

Nilsson, K., Johansson, R., Robertsson, A., Bischoff, R., Brogårdh, T., Hägele, M., 2005: *Productive robots and the SMErobot™ project.* Third Swedish Workshop on Autonomous Robotics, Stockholm, September 1-2, 2005.

Naumann, M., Wegener, K., Schraft, R.D., Lachello, L., 2006: Robot Cell Integration by Means of Application-P'n'P. In: *Proceedings of the Joint Conference on Robotics,* May 15-17, 2006, Munich

Gauss, M., Dai, F., Som, F., Zimmermann, U.E., Wörn, H., 2006: A standard communication interface for industrial robots and processor based peripherals - XIRP, In: *Proceedings of the Joint Conference on Robotics,* May 15-17, 2006, Munich.

VDMA, 2006: *XML-basiertes Kommunikationsprotokoll für Industrieroboter und prozessorgesteuerte Peripheriegeräte (XIRP)*, VDMA 66430-1:2006-07, 2006

Gecks, Th., Henrich, D., 2004: SIMERO: Camera Supervised Workspace for Service Robots. In: *Proceedings of ASER 2004, 2nd Workshop on Advances in Service Robotics,* Feldafing, Germany, 20-21 May 2004

AUTOMATIC GENERATION OF EXECUTABLE CODE FOR A ROBOT CELL USING UPNP AND XIRP

Alexander Verl and Martin Naumann
Fraunhofer Institute for Manufacturing Engineering and Automation
Nobelstr. 12, 70569 Stuttgart, Germany
naumann@ipa.fraunhofer.de

Keywords: Plug'n'Produce, High-Level Programming, Automatic Code Generation.

Abstract: This paper deals with the concept of a control architecture for robot cells that enables Plug'n'Produce according to Plug'n'Play in the office world. To achieve this, the cell controller needs special functionality located in a software module called "P'n'P-Module". This module takes as input descriptions of devices and processes. These descriptions are then automatically evaluated in order to offer the user device-independent high-level commands to define a task for the robot cell. Based on this task definition an executable code has to be generated. The focus of this paper lies on the descriptions and algorithms necessary to generate this executable code. The presented method will be realized as a test bed within a bin picking cell using UPnP and XIRP.

1 INTRODUCTION

In Germany, 45% of all robots in 2006 have been shipped to the automotive industry, not counting other industry sectors with mass production 0. The tasks robots have to fulfil there are mostly highly repetitive and do not change over an extended period of time. Therefore, the main requirements for robots used in mass production are short cycle times. The goal of the european project SMErobot™ 0 is to broaden the field of applications for robots from mass production to small lot size production, as it is typically encountered in small and medium sized enterprises (SMEs). Because of small lot sizes, fast adaptability of robot and surrounding cell to new products and processes is much more important for SMEs than short cycle times. To make this possible the programming of applications for robot cells and the integration of new devices into these robot cells must be adapted to these new requirements.

2 APPROACH AND SCOPE OF THIS PAPER

In the office world it is very easy to install and use new devices. For example, to install a printer to your PC, you just plug it in. The entire configuration is then done automatically and your application will offer you the service "`print`". This automatic configuration is called "Plug'n'Play". Carried forward to a production environment this would mean that you would connect e.g. a robot to a cell controller and it would offer you the service "`move_to`" on a HMI. Even more advanced, it could mean that you connect e.g. a robot and a gripper to a cell controller and the cell controller would recognize the new possibilities enabled through the combination of two or more devices and offer you the service "`pick and place`". To achieve this, the cell controller needs to know about the functionality of the connected devices and must be able to draw conclusions which services it can offer to a user. The approach pursued in this paper is based on device- and process-descriptions evaluated in order to offer services representing the functionality of the robot cell to a user.

The ability to add devices to a robot cell and to use the functionality of these devices without the need of configuration is called "Plug'n'Produce", according to "Plug'n'Play" in the office world and is provided by a so called "Plug'n'Produce-Module". Plug'n'Produce (P'n'P) can be broken down into three layers depending on the amount of configuration done automatically:

- **Communication Plug'n'Produce:** deals with communication protocols. Automatic setup of a

basic means of communication between cell controller and devices includes discovery and addressing of devices.

- **Configuration Plug'n'Produce:** automatically configures all the settings the users should not need to care about, e.g. bandwidth requirements, default values, …
- **Application Plug'n'Produce:** automatically offers services to the user corresponding to the functionality of the robot cell.

The focus of this paper lies on the Application-P'n'P-layer. Of course, this layer depends on the Configuration- and the Communication-P'n'P-layers in order to get to know which devices are available, to communicate with these devices and to get to know the descriptions of these devices 0. However, the two lower layers will not be within the scope of this paper as they are already realized in available communication protocols like XIRP and UPnP that will be used.

3 STATE OF THE ART

UPnP 0 and XIRP 0are both XML-based client-server communication protocols that both support eventing and in the case of XIRP also cyclic communication. UPnP was mainly developed by Microsoft® for the PC-world while XIRP (XML Interface for Robots and Peripherals) was developed by a consortium of companies within the German public funded project ARIKT.
Both protocols support the definition of device profiles as do also many other communication protocols 0. These device profiles define programming interfaces that have to be supported by a device in order to belong to a certain device category. The functionality of the device can partly be inferred from the programming interface, but it is not itself part of a device profile. Therefore, device profiles do not contain enough information to allow detailed assumptions about the functionalities of devices.

In the domain of knowledge representation, languages have been developed that can be used to describe functionalities of devices in form of a taxonomy plus additional attributes. The most popular of these languages is OWL (Web Ontology language). It was developed as a key technology of the Semantic Web 0 with the goal to add meaning to the information that is today merely displayed in the internet. This additional information can be used to enable knowledge based services that contain several entities.

In the context of home entertainment systems, a function planning module was developed within the SmartKom project. This module tries to serve complex user requests by first determining which devices are necessary and then determining how to control devices based on abstract descriptions of the functionalities of devices 0, 0.

In this paper, the concept of device profiles augmented by a detailed description of the device's functionality with a knowledge representation language is used to infer the functionality of a robot cell within the Plug'n'Produce-Module that adapts concepts of the function planning module of the SmartKom project to the robotic domain to generate executable code for UPnP- and XIRP-devices thus enabling Application-P'n'P.

4 APPLICATION-P'N'P OVERVIEW

Application-P'n'P as the highest P'n'P-layer has the goal to offer the user as easy as possible means of using the functionality of a robot cell. In the context of SMErobot™ this means offering the user as easy as possible means of adapting robot cells to new tasks.

State of the art of defining the sequence for robot cells is to enter commands in the dialog of some sort of a programming system. The entered commands are then uniquely mapped to devices. This is an appropriate way of programming as long as the user has detailed knowledge about the control structure of devices as well as about programming itself. In the context of a SMErobot™-application this cannot be granted. Users of robot cells in SME environments normally know a lot about the processes they have to perform in order to achieve the desired result, but have only minor knowledge about programming devices (a robot is a special kind of device). Therefore, the definition of sequences for robot cells in SME environments should be possible without the need of device programming. Instead, programming should be focused on the processes the user wants to execute. In this paper, this will be called **"process-oriented programming"** and the corresponding commands will be called **"process commands"** as opposed to traditional "device commands".

Process commands trigger whole processes like drilling a hole or gripping a part, while device commands trigger a state change in a single device like setting a digital output or moving a robot from

point A to point B. Process commands are a much more general approach than subroutines because they define the sequence of actions for a process and the required functionalities. They abstract programming interfaces and communication and are therefore independent of specific device properties or communication protocol properties.

The mapping of specific device commands to a process command in order to generate executable code will be described in detail in the following chapters.

The introduction of process commands imposes the following requirements on the robot cell controller:

- The robot cell controller needs information about the functionality of the available devices and must be able to infer the subset of available process commands that can be executed by the current setup of the cell.
- The robot cell controller must be able to automatically generate code to execute the sequence of process commands defined by the user.

To fulfil these requirements the "P'n'P-Module" is introduced. It acts as an intermediate between user and devices. The operating mode of the P'n'P-Module will be described in detail in the following chapters. Figure 1 shows a block diagram of the P'n'P-Module and its environment.

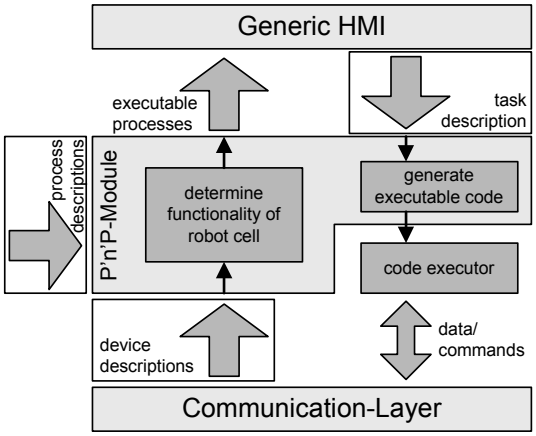


Figure 1: Block diagram of the P'n'P-Module and its environment

5 DESCRIPTIONS

As shown in figure 1, three types of descriptions are necessary:

- **Device Descriptions** containing information about the functionality and the programming interface of devices.
- **Process Descriptions** containing information about the required functionality of a process and the sequence of actions.
- One user-defined **task description** containing information about the sequence of processes and according process parameters.

Information on the determination of the functionality of the robot cell can be found in 0. Therefore, this paper concentrates on the generation of executable code out of descriptions of programming interfaces of devices, the description of the sequence of actions of processes and the user-defined task description.

5.1 Device Descriptions

The description of programming interfaces of devices is realized in form of state charts, called "Device State Charts". Device State Charts can have as many states as necessary, but depending on the functional description of a device certain states are mandatory. If the functional description of a device contains a certain skill, the state chart must contain certain mandatory state(s), e.g. if the functional description of a gripper contains the skill "CanGrip", the Device State Chart of this device must contain the states "open" and "closed". Apart from these mandatory states, the Device State Chart may have other additional states that replicate special properties of the device controller. Figure 2 shows an exemplary Device State Chart of a gripper. The states "configuring", "opening" and "closing" are additional states.

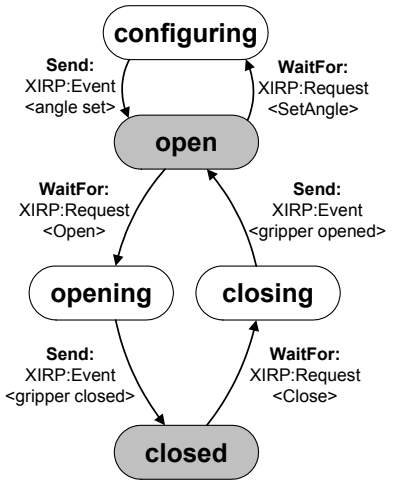


Figure 2: Device State Chart of a simple gripper.

The transitions of the state chart describe the device commands that triggers the state changes. In the case of the bin picking test bed described in chapter 7 UPnP and XIRP will be used as communication protocols. Therefore the transitions of the Device State Charts describe device specific UPnP and XIRP communication to control the devices.

Several languages exist to describe state charts. One of them is SCXML 0. SCXML allows the concurrent execution of parallel state charts and their synchronization and is therefore well suited for the use in Device State Charts.

Device State Charts are a mandatory part of device descriptions in order to generate the necessary sequence of commands to reach certain states – that means, to execute a certain task.

5.2 Process Descriptions

General Process State Charts describe the states the involved devices have to reach, their order and synchronizations that must be taken into account. They are the counterpart of Device State Charts. General Process State Charts have the purpose of describing the sequence of actions of a certain process. "General" means that they describe this sequence independent of the devices actually used and therefore independent of their specific programming interfaces. Therefore, they must describe which states must be reached by the devices in which order to execute a certain action, but they must not describe how these states can be reached as this depends on the devices actually used. General Process State Charts consist of separate state machines for each involved devices. These separate state machines are synchronized where necessary, e.g. to assure that a gripper is closed only after the robot has reached the gripping position. Figure 3 illustrates the General Process State Chart of a picking process.

General Process State Charts are expressed in SCXML, too.

5.3 Task Description

The task description is defined by the user of the robot cell on the Generic HMI as sequence of process commands. The Generic HMI displays all executable processes to the user. The user defines a sequence of processes and enters the corresponding

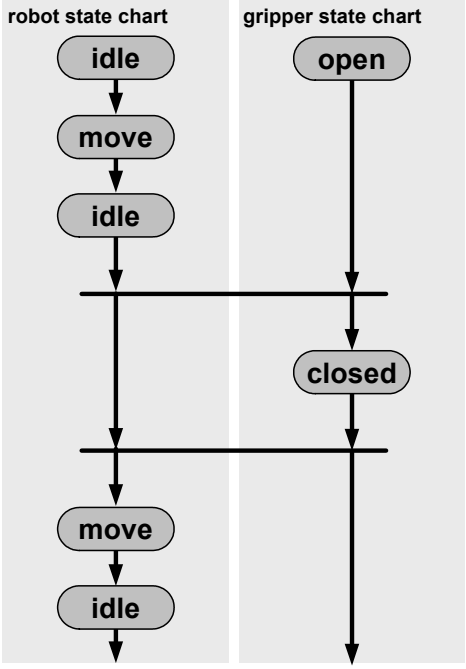


Figure 3: General Process State Chart of a picking process.

process parameters like e.g. gripping force, picking position or robot speed. For that purpose dialogs are automatically generated out of the process descriptions. It is either possible to enter the required process parameters directly on the HMI or, if available, with the help of input devices. Positions could e.g. be taught with lead through programming if the robot is equipped with a force torque sensor and the controller supports lead through programming.

6 GENERATION OF EXECUTABLE CODE

In order to run the task defined by the user code has to be generated that can be executed by the code generator (see figure 1). This code generation consists of two steps. First, General Process State Charts and Device State Charts are combined to (device-) Specific Process State Charts. Second, the Specific Process State Charts are concatenated according to the user-defined task description to a Task State Chart. In this Task State Chart, the user-defined process parameter values are included. Figure 4 illustrates the workflow.

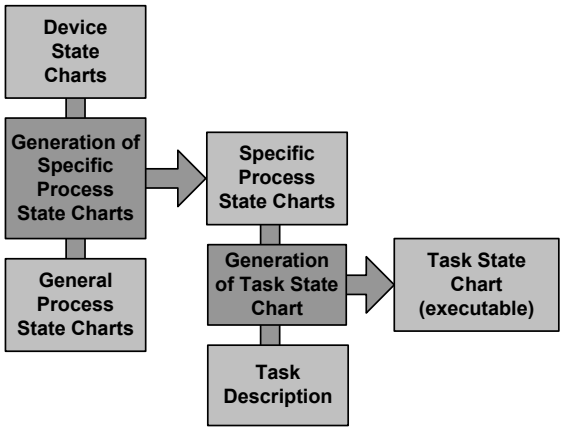


Figure 4: Workflow to generate executable code.

6.1 Generation of Specific Process State Charts

Specific Process State Charts are generated out of General Process State Charts by adding device commands to the transitions.
Therefore, the states of the General Process State Chart are mapped to states of the Device State Charts of the used devices. The mapping is possible because the states of the General Process State Charts and the states of the Device State Charts are related by an ontology. The device commands are then added stepwise by searching a path in the Device State Chart for each Transition in the General Process State Chart. This path including all states, transitions and device commands in between is then inserted into the General Process State Chart. Once this path-search has been done for a whole General Process State Chart, the result is a (device-) Specific Process State Chart. Figure 5 illustrates this approach exemplary using the Device State Chart shown in figure 2 and the Process State Chart in figure 3.

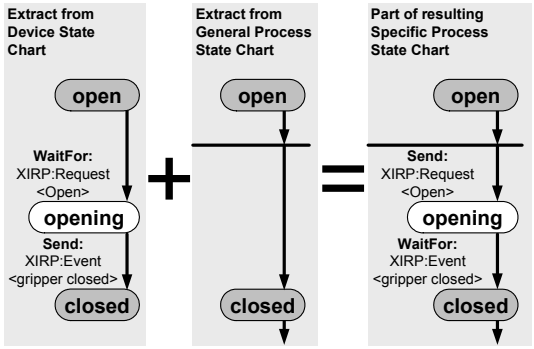


Figure 5: Generation of Specific Process State Chart out of Device State Chart and General Process State Chart.

6.2 Generation of Task State Chart

To generate the Task State Chart, the Specific Process State Charts are concatenated according to the task description. If a process does not involve a device, this device stays in the last state of the previous process. Finally, the user-defined process parameters are included. Result is a state chart containing device commands of the used devices that can be executed. Figure 6 shows an example of a Task State Chart.

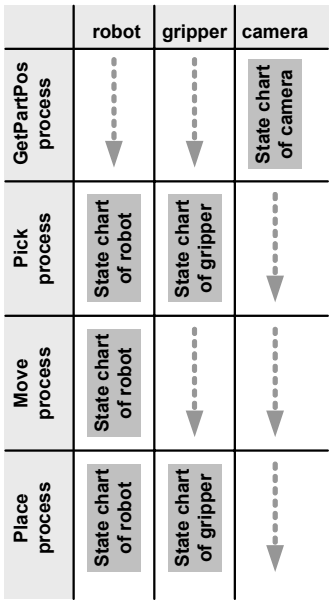


Figure 6: Generation of Task State Chart by concatenating processes.

While concatenating the processes, a basic plausibility check is performed to assure that only processes with matching final and start states are attached. This plausibility check assures e.g. that the gripping process shown in figure 3 cannot be used without in between opening the gripper again in some other process. In this way some errors of the user defined Task Description can be detected.

6.3 Executing the Task State Chart

The execution of the Task State Chart is done by the code executor in the cell controller (see figure 1). For each involved device, a state machine is initialized with the start state of the first process. From then on these state machines check cyclically if the condition of a transition is fulfilled. If yes, a state change is triggered and the state machines

switch to the next state. State changes of a device can either be triggered by an incoming message from that device, by a state change of another device or by a transition without a transition condition.

7 TEST BED BIN PICKING

The presented concept will be realized as test bed in a bin picking robot cell. This cell consists of the following devices:

- Robot
- Gripper
- 3D-Sensor
- PC that runs the bin picking algorithms

All these devices are connected to a cell controller. The cell controller runs the P'n'P-Module with the described functionality. Figure 7 illustrates the underlying control architecture.

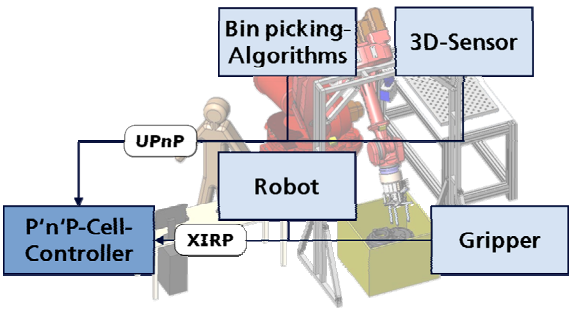


Figure 7: Control Architecture of the bin picking cell.

All devices have their own controller that offers a programming interface to access their functionality. This programming interface is accessible either via XIRP or the UPnP communication protocol. Because both protocols support automatic discovery and initialization of communication, the devices are integrated into the cell controller without manual configuration effort. Then, description files containing the Device State Chart are loaded into the P'n'P-Cell-Controller. The P'n'P-Cell-Controller uses these descriptions to evaluate the cell functionality and – after the user has defined a task – generate and execute code as described in this paper.

8 CONCLUSIONS AND OUTLOOK

The presented concept allows programming of a robot cell without knowing details about the underlying programming interfaces and communication protocols and therefore permits users with little knowledge of (robot) programming to use robots. The user has to combine and parameterize the processes but does not need to use device commands. To facilitate the parameterization, intuitive input devices can be integrated into the cell controller.

The abstraction layers introduced to achieve this goal furthermore allow easy exchange of devices with different programming interfaces and communication protocols as long as they offer the same functionality.

The concept should help users in a SME environment to define typical machine tending or part handling tasks that do not require closed control loops extending over several devices as the present concept cannot cope with real time requirements. One possible solution would be to include mechanisms into the communication layer to support real-time provided that real-time communication protocols are used. Another, more advanced approach would be to establish direct real-time connections between devices that need to exchange time-critical data. This approach would impose new requirements on the devices and the underlying network.

Another possibility to further advance the presented concept is to upgrade the plausibility check described in chapter 6.2. The available information about the meaning of processes and states could be used to not only detect task definition errors, but also make suggestions to the user on how to correct them.

A third advancement of the presented concept could be an upgrade of the code executor. At the moment it executes the generated Task State Chart sending single commands to devices to trigger actions. Because state charts are a very general way of representing programs, the Device State Chart could be used to generate complete programs for single devices using transformation rules. This would allow generating e.g. a program for a robot, downloading it and running it on the robot controller thus significantly reducing the communication effort.

The bin picking test bed will give the opportunity to prove the presented concept, to draw conclusions about its strengths and weaknesses and by this means decide about the next steps.

ACKNOWLEDGEMENTS

This work has been funded by the European Commission's Sixth Framework Program under grant no. 011838 as part of the Integrated Project SMErobot™.

REFERENCES

World Robotics 2007, IFR Statistical Department.
http://www.smerobot.org
Papas homepage: http://www.projekt-papas.de
UPnP Device Architecture; Version 1.0; 8.6.2000. Downloadable from the UPnP-Forum: http://www.upnp.org
VDMA Einheitsblatt 66430-1: XML-basiertes Kommunikationsprotokoll für Industrieroboter und prozessorgestützte Peripheriegeräte (XIRP) - Teil 1: Allgemeine Vereinbarungen.
Riedl, M.; Simon, R.; Thron, M.: EDDL – Electronic Device Description Language. München, Oldenburg Industrieverlag, 2002.
OWL Web Ontology Language Overview, 10.4.2004. Downloadable from W3C: http://www.w3.org/TR/owl-features/
Berners-Lee, T.; Hendler, J.; Lassili, O.: The Semantic Web, Scientific American, 17.1.2001.
SmartKom homepage: http://www.smartkom.org/
Torge, S., Hying, C.: Realizing Complex User Wishes With a Function Planning Module. In: SmartKom: Foundations of Multimodal Dialogue Systems. Berlin. Heidelberg. Springer Verlag, 2006.
Naumann, M.; Wegener, K.; Schraft, R. D.: Control Architecture for Robot Cells to Enable Plug'n'Produce. In: Proceedings of ICRA 2007.
State Chart XML (SCXML): State Machine Notation for Control Abstraction 1.0, W3C Working Draft, 24.1.2006. Downloadable from W3C: http://www.w3.org/TR/scxml/

PLUG-AND-PRODUCE TECHNOLOGIES REAL-TIME ASPECTS

Service Oriented Architectures for SME Robots and Plug-and-Produce

Klas Nilsson
Department of Computer Science, Lund University, Box 118, SE-22100, Sweden
Klas.Nilsson@cs.lth.se

Matthias Bengel
Robot Systems, Fraunhofer IPA, Nobelstr. 12, DE-70569 Stuttgart, Germany
Matthias.Bengel@ipa.fraunhofer.de

Keywords: Plug-and-Produce, Plug-and-Play, real-time computing, flexible manufacturing systems, robot.

Abstract: Plug-and-Produce (with the meaning that devices can be plugged in without any manual configurations needed) is an attractive paradigm for manufacturing systems, and in particular for Small and Medium Enterprises (SMEs) that do not have the expertise of system integrators but do need to be able to reconfigure their systems by themselves. One approach for loosely (in terms of timing) coupled devices is that of Service-Oriented Architectures (SOA). As can be understood from developments with multiple robot arm and online operator interactions, future applications will in some cases need real-time guarantees for performing services. That includes both real-time communication and the need to perform services with a predictable timing. A review of available technologies and inherent limitations of distributed computing leads to the conclusion that the standard SOA approach based on process oriented (like for RPCs and web services) calls similar to distributed object orientation will not be practically useful. Instead, a data or state centric approach should be adapted together with one-way message-based communication.

1 INTRODUCTION

The idea of "just plug in and produce" for manufacturing equipment is very attractive. It is inspired by the Plug-and-Play concept for PCs, which has been developed from something less well working for the old ISA-bus PCs running Windows95, into a quite useful end-user support for plugging in USB and other types of devices. There are sometimes real-time requirements on the communication between PCs and their peripherals, and in manufacturing there are real-time requirements on the communication between different devices.

Such devices in manufacturing can be various types of equipment such as robots, advanced sensors and PLCs, and real-time communication is accomplished via field buses or direct wiring, but then with very limited PnP support. Comparing with the simplicity of connecting appliances to a home PC, one should observe that the PC then is a master device that deals with real-time over dedicated communication lines such as fire-wire for a camera.

The real-time problem is not really there, it just works in a reasonable way when sufficient resources are provided (such as USB2 for a memory stick), and when that is not the case the user has to be a bit more patient.

Large enterprises have technical experts that deal with system integration and setup of communication around robots and other types of machines. The future anticipated wide-spread use of robots in SMEs, however, leads to a situation where non-experts (home PC users expecting PnP to work) need to setup and maintain their robot installations. In this situation, the lack of competence is not a problem; it is a reasonable challenge that automation/computer/software engineers should solve. The problem is, however, the lack of awareness within the involved engineering disciplines concerning the inherent or hidden limitations that pop up as peaks of complexity for the end user, which efficiently prevents the PnP

paradigm to work in practice in SMEs. A certain infrastructure coping with the (then internal) complexity is needed.

Is explicit support for real-time communication part of the needed infrastructure or not? Well, it is typical for infrastructure (phone lines, high-ways, health care, and so on) that:

- Persons (in this case engineers) have different opinions about what need to be provided and what it left to be solved on a case-by-case basis.
- Persons (still engineers, considering real-time and PnP) imagine a cost for additional features, and if the imagined cost is high compared to the imagined (not actual) benefits, there is a resistance of supporting development of the support.
- Lack of a complete supporting infrastructure can result in complete systems being developed in parallel, with deficient overall efficiency, since experienced problems (even if only in few cases) may be severe.
- A well-working infrastructure is not really noticed since it is taken for granted (only noticed when not working), so even persons that actually depend on it may pay no attention to it.

Returning to PCs and IT infrastructure, a comparison with security is relevant. Security should better be built in using solid principles within device drivers, operating systems and in middleware solutions. Since that has not been the case for typical PC systems, the costs in terms of failing systems/enterprises and add-on protective systsems (hardware and software) has been enormous. In development of future automation solutions we should be extra careful since the field by itself is not big enough to cover huge extra costs; instead we have to benefit from low-cost solutions. Therefore, if PCs and low cost devices (hardware and software) do not provide the needed solutions, we should pay extra attention to get the foundations right for future efficient SME usage.

Is then real-time communication and real-time execution something we need; is it something that can be added afterwards; or what should be the foundation for future applications? The standard answer today is that real-time is needed in very few cases, so let's neglect it and focus on other application aspects.

One promising approach is then so called Service-Oriented Architectures (SOA), which could be quite suitable (at least for non-real-time parts) but the implementations tend to be slow and hard to map onto real-time suitable implementations. For instance, both UPnP and web services are implemented on top of http with XML-based information structures that not necessarily map on hardware supported real-time means of communication. We will come back to this issue, but first some preliminaries that our discussion can benefit from.

2 PRELIMINARIES

So called middleware and models of distributed components providing software services typically come from enterprise systems, but is also being used for mobile robots and other system interconnections with no strict real-time requirements. For real-time communication within industrial automation, the current practice is based on field buses. In modern integrated systems with the need for so called vertical integration, there enterprise and device levels need to be able to communicate and the different technologies need to be unified.

2.1 Basic Model of Communication

Software developers today normally use object-oriented programming, and from the beginning they learn how to use method calls for object interaction. Multi-threaded applications, today typically written in Java or C# with language support for synchronized methods, also follow the object-oriented paradigm quite well, which means that two-way synchronous communication is the basis for inter-object communication within a single program.

Developers with experience from computer networking or from data-flow oriented applications with needs for buffering of asynchronous messages may build distributed applications differently than local programs, simply to deal with the quite different and complex communication reality. However, along the lines of hiding complexity, a perhaps more common trend is to stretch the object-oriented paradigm to cover also distributed systems, which by definition are concurrent (but so far we assume no real-time requirements). This is also the basis for several of the middleware approaches that are listed below. The problem is, however, the distributed object-oriented paradigm has limited applicability when assumptions about the networked object interaction do not hold. That is, for realistic applications (such as robot work cells) the networking for object interaction does not scale or does not handle typical deployment contexts. This has been known for over a decade as appropriately described by (Waldo, 1994), and the reader is suggested to read at least sections 2 and 7 of that report before continuing here.

In our case dealing with automation and real-time demands (not to mention safety), the situation is even more difficult, but still some of the dead-end approaches are being promoted. To review the current situation some further details on existing solutions and requirements now follow.

2.2 Fieldbuses

The classical field buses are usually setup in a ring-like structure, which is natural since the original purpose was to minimize cabling. Some of them provide hard real-time capabilities (like Sercos), others just implement soft real-time like Profibus and CAN bus. There, messages can almost be made sure to be delivered in time – depending in their priority. Nevertheless, the CAN bus supports distributed hard real-like control although the communication itself doesn't (CAN, 1991). These field buses make use of their own physical layer. Sercos uses fibre optics, CAN bus relies on a three-wire cable.

Typically a rather limited number of devices can be connected to one network. Profibus and CAN bus for instance support up to 127 participants.

Newer developments don't use their own hardware layer but rely on the Ethernet technology. Well-known examples are Sercos III (Sercos, 2002), ProfiNet (Profibus, 1999) (ProfiNet is part of the Profibus specification since 2003) and EtherCAT (ETG, 2007]. All of them implement real-time capabilities. This is mainly done by replacing some of the ISO/OSI layers (Zimmermann, 1980) in the standard TCP/IP stack. Therefore, the wiring gets less complex and requires less effort. Further more, in some of these technologies – if hard real-time is not required – standard hardware like network switches from the office world can be used.

As these technologies make use of the IP technologies, the number of participants gets larger and is not limited to only a small number. One fairly successful attempt to bring together several protocols under a unifying overall framework is the CIP (ODVA, 2006) initiative. However, such general solutions go with lengthy descriptions and detailed APIs that are not easily adopted.

2.3 Middleware

Apart from the field bus technologies there are several approaches to communicate not only between different automated hardware devices but as well between different programs, spread in the network. Currently, some main directions can be observed:

2.3.1 Web Services

The web services are a mainstream technology in the B2B (business to business) communication and are used there mainly as Enterprise Java Beans (Sun, 2007) and Microsoft's .Net technologies. The web services are usually implemented using the SOAP protocol (W3C, 2007) that communicates via Http and TCP/IP over Ethernet. The main advantage is its flexibility and availability in the intranets and the internet due to using the http port. But in this scope, the main disadvantage is the lack of real-time capability due to the standard network protocols.

2.3.2 OPC/UA

OPC/UA (OPC Unified Architecture) is the newest of all OPC specifications (OPC, 2007). It contains an own communication stack which is scalable from embedded controllers up to main frames.

The architecture follows the SOA paradigm (service oriented architecture) including several logical layers. It supports profiles which can be queried. Therefore, communication partners can query their provided services.

As OPC/UA was invented for communication via the internet, its architecture supports security features like encryption and authorisation. Determinism is not included. Internal tests discovered that round-trip times are short enough to implement even control loops for devices in automation technology.

2.3.3 Corba

Particularly in the research area different Corba implementations are used widely. Corba is the abbreviation for Common Object Request Broker Architecture and is a specification of an object-oriented middleware (Mowbray, 1997). Its core is a so-called object broker which defines platform-independent protocols and services.

Usually the program code for the communication over the network is not written manually. Instead, an abstract language, the Interface Definition Language (IDL) is used. From that, the stubs and skeletons are generated automatically for various programming languages and different operating systems. This is why Corba is platform-independent on the one hand and programming language-independent on the other hand.

Newer implementations of Corba even support real-time, provided that the underlying operating systems and communication channels do as well.
Due to the fact that there are several different Corba implementations it cannot be assumed a priori that all the different implementations interoperate well as

there seem to be some differences in the concrete implementations. The specifications for Corba and IDL can be retrieved from (OMG, 2007).

2.3.4 Universal Plug and Play

The main purpose of Universal Plug and Play (UPnP) is to control devices independent of their manufacturers. UPnP is well-known in controlling routers and multimedia equipment.

Originally, UPnP was introduced by Microsoft, but nowadays, certifications for devices are performed by the UPnP Forum (UPNP, 2007) which at the time of writing consists of 845 vendors.

UPnP can be used on any communication channel supporting IP communication. Basically, UPnP makes use of several protocols for discovery, addressing, description, eventing and so on. Also the technologies IP, UDP, Multicasting, Http, and SOAP are well known and are used in this technology.

Unlike some the other middleware standards described here, UPnP does not support any security features. A good introduction to UPnP can be found in (Jeronimo, 2003).

2.3.5 Representational State Transfer

The standard way of implementing web service (using SOAP as in UPnP) has a number of drawbacks in terms of (Newmarch, 2005):

1. Inefficiency with XML-based RPC-like communication on top of http.
2. Unclear semantics in the use of GET and POST requests.
3. Unclear object model and deficient referencing of attributes in nested data structures.

To overcome these difficulties, the REpresentational State Transfer model was suggested by Fielding (Fielding, 2000) to overcome the above drawbacks.

Technologies used in Microsoft Robotics Studio® (MSRS, see Microsoft.com for latest info) is claimed to include a lightweight REST-style service-oriented runtime, but the Decentralized Software Services Protocol (DSSP) is actually SOAP based. Therefore, even if DSSP is oriented towards exposing device states, it is not clear how the transfer of state information can be mapped to real-time eventing as we aim for.

3 FUTURE COMMUNICATION AND MIDDLEWARE

For PnP automation devices to become a reality, it must be easy and streamlined to develop such devices. The reason is that the strong arguments for interoperability as in telecommunications do not apply to automation, which also is to small an area to cover extensive developments of special solutions. Hence, we must be able to make use of available technologies in a swift manner.

Note that the solution is not standardisation, at least not in the traditional sense with agreements that are negotiated in committees and then maintained as thick documents. There are already a lot of standards; we do not need more of them (unless there is a core new technical solution that calls for some agreements on how to make use of it).

What is the suitable approach then, and what are the requirements?

3.1 Requirements for R&A Systems

To support the desired PnP developments we need middleware providing an API with a suitable expressiveness and simplicity. These two demands are contradictory and it is an open issue if a good solution can be found or not. For instance, even if there are abstractions and APIs for communication channels, the profiles and specifics for fieldbuses may add too much of complexity if the selected abstractions do not map onto the actual setup. Another example is CORBA that should simplify programming of distributed applications, but still adds too much complexity (for programming, for deployment, and for troubleshooting). Thus, abstractions need to be defined on an appropriate (probably medium) level to avoid problems:

1. Too low level: The complexity of networking and field-buses gets exposed and the API gets useless for a majority of the developers.
2. Too high level: Communication setup not reflecting actual needs in terms of timing and resources, and a variety of APIs will evolve.

Hence, the middleware should deal with communication in terms of an application oriented (not networking oriented) API that is designed to map well on the most suitable alternatives for real-time communication. To structure the topic we may separate between:

- Mechanism: How are things accomplished technically, typically locally in a computer node or on the network?
- Policy: How are the mechanisms used and configured locally?
- Deployment: How is an actual system configured before and during run time?

Taking a look at just the mechanisms for communication and connections between devices, there are at least the following to consider, some if it

affecting the software in a less obvious way. For brevity, the following items (that each deserves one paper) are very shortly described without references:

a) **Driven by time or events**: EtherCAT, TTP, and RTNet are time driven while ThrottleNet and normal socket connections naturally forms event driven communication. Combinations?
b) **Exposing services or data**: Configuration methods and operational state, RPC or REST?
c) **Operating peer-to-peer or master-slave**: Is an EtherCAT slave PC node a slave or a peer?
d) **Connecting peer-to-peer or client-server**: Server socket in server, client, or p2p software?
e) **Topology as star or ring**: Implications on predictability, reliability, resources and cabling?
f) **Connection-based or datagram channels**: How to deal with the tradeoff between performance and reliability/simplicity?
g) **Synchronous or asynchronous**: Both events and calls can be both. Best practices and APIs?
h) **Bidirectional or one-way**: Should there be a built-in support for handling event replies, for unreliable low-cost means of communication?
i) **Hot-plugging or reset**: EtherCAT connected to an end-effecter via tool exchanger, is a communication dip during tool changing acceptable?
j) **Predictable or best-effort**: Specification of performance requested or obtained, but what does it mean for the application software?
k) **Dependable or fail-safe**: SME robots only need to be failsafe or are there mission-critical cases?
l) **Guarded or collaborating nodes**: Does human-robot space-sharing imply a need for the 'babbling-idiot' protection as in TTP.

Since not all this complexity should be exposed in a complete API (that nobody would use), we need:

- Tradeoffs such that the most critical and common cases are well supported, for instance by suitable default configurations.
- Abstractions in layers and a guide such that only a few types of configurations are needed in actual scenarios. Integration with model-driven design tools is desirable.
- Ontology-based definitions of the communication model, including formal definitions of items a to l above. Today standards and definitions are only expressed in documents (for humans) and code (for computers), but there is no meaning including semantics that is understandable by both humans and machines, which is necessary for application-level PnP.
- Open source reference implementation working with some generic devices. Different vendors will then adopt the software (or perhaps re-implement in other languages and for other platforms) but agreements and specifications need to be with respect to actual runnable code.

Suitable tradeoffs with respect to embedded distributed software for robot work-cell devices should primarily suit low-cost solutions as needed for SMEs. Our research indicates that the following decisions are appropriate:

A. Real-time data-flows should be based on one-way data streams that from a programming point of view is equal to an event or message stream, but only resource use should be defined programmatically (e.g. by providing a maximum size message and a maximum frequency) and no configuration of communication profiles should go into the application code.
B. There needs to be a binary version of the real-time data flows, with complete description of message types when a connection is established but with only minimal binary information during real-time operation. That way most control messages fit into Ethernet frames and low cost raw Ethernet can be used for predictable communication.
C. Real-time RPC, RMI, CORBA method calls, or web services, should not be permitted, at least not the standard IDL way. If permitted, the underlying asynchronous operation should be explicit, meaning that there is a call object that can be queried for completion, errors, etc.
D. Non-real-time network traffic should be possible to do in the same way as for real-time communication, but in this case synchronous method calls could map (automatically) to RPC calls or web services.
E. All encapsulated entities used by the real-time parts of the application should be resource aware; real-time is just a special case of resource limitations (namely CPU power and the scheduling of it), so also memory usage and IO bandwidth allocation should be taken care of in a structured way.
F. The use of safe languages such as Java and C# should be used for improved modularity and robustness of hand-written code. Unsafe languages such as C++ should not be used for flexible dynamic parts of systems since the risk for dangling pointers and crashes get too high.
G. All systems should (without extra engineering) run on standard desktop computers for simulation and debugging purposes, then without real-time performance but with full concurrency using a virtual time scale.

There will of course be no power to enforce the decisions above and standardization via a committee would not work; freely available implementations of selected abstraction must be the most simple and efficient way of building systems, and thereafter de-

facto standards should evolve. In this perspective compatibility (either directly or via bridges) with major wide-spread middleware solutions such as MSRS must be supported. A variety of initiatives are ongoing, including the Apache CXF Open Source Service Framework (Apache, 2007).

The focus in REST on data rather than methods (or nouns instead of verbs) suits our manufacturing scenario quite well since it is data that is actually transferred over the network and simple mappings of device state to network data should permit tiny devices to be part of the PnP system. Using switched raw Ethernet (Martinsson, 2002) and self-descriptive data packets (Blomdell, 2007) then supports low-cost solutions. To find out if the above technical decisions are appropriate or not, more assessments are needed to get application experiences. That is ongoing but outside the scope of this paper.

4 CONCLUSIONS

Appropriately designed real-time capable middleware and PnP support will most likely simplify for application development rather than being a complication. Support for real-time communication should be built into the abstractions we use for communication between programs and computers. Real-time support means permitting real-time operation (when OS and all involved parts comply), so well-written applications will provide real-time capabilities when deployed on a real-time capable system. Many promising technologies and partial solutions have been developed over the years, but it appears there are no solution with the completeness and scalability that is needed for the future very flexible and modular SME applications. A suitable approach appears to be open-source reference implementations of suitable abstractions for Ethernet-based communication and development of middleware that is compatible with (but also useful independently of) the Microsoft Robotics Studio. Additionally, special attention should be paid to self-descriptive binary communication channels that map well onto raw Ethernet and that can be bridged automatically to high-level XML-based eventing as used in several of the existing standards. Such developments are currently ongoing in the SMErobot consortium.

ACKNOWLEDGEMENTS

This work was supported by the EU FP6 project SMErobot®.

REFERENCES

Apache, 2007: Apache CXF: An Open Source Service Framework, http://incubator.apache.org/cxf

Blomdell, A., 2007:The LabComm Protocol Language, http://torvalds.cs.lth.se/moin/LabComm

CAN, 1991: CAN Specification, Version 2.0, ISO 11898: 1993-11

ETG, 2007: EtherCAT Technology Group website, http://www.ethercat.org

Fielding, R., 2000: Architectural Styles and the Design of Network-based Software Architectures, http:// www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

Jeronimo, M., 2003: UPnP Design by Example. 2003 Intel Press. ISBN 0971786119.

Martinsson, A., 2002: Scheduling of Real-time Traffic in a Switched Ethernet Network, Master thesis, https:// www.control.lth.se/database/publications/article.pike?action=fulltext&artkey=5683

Mowbray, T. J. and Ruh, W. A., 1997: Inside Corba: Distributed Object Standards and Applications. Addison Wesley 1997. ISBN 978-0201895407.

Newmarch, J., 2005: A REST Approach: Clean UPnP without SOAP, Newmarch, J., "A RESTful approach: clean UPnP without SOAP," *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE* , vol., no., pp. 134-138.

ODVA, 2006: The Common Industrial Protocol (CIP™) and the Family of CIP Networks. At *http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00123R0_Common%20Industrial_Protocol_and_Family_of_CIP_Netw.pdf*

OMG, 2007: OMG Specifications. – Middleware Specifications.http://www.omg.org/technology/ documents/spec_catalog.htm

OPC, 2007: The OPC Foundation. http://www.opcfoundation.org/Downloads.aspx?CM=1&CN=KEY&CI=283

Profibus, 1999: Profibus Specification. 1991/1993 DIN 19245, IEC 61158/IEC 61784

Sercos, 2002: SERCOS interface, 2002. IEC/EN 61491.

Sun, 2007: Enterprise Java Beans, Sun Microsystems, 2007. http://java.sun.com/ejb

UPNP, 2007: The UPnP Forum. http://www.upnp.org/

W3C, 2007: SOAP Specifications, 2007. http://www.w3.org/TR/SOAP/

Waldo, J., Wyant, G., Wollrath, A., Kendall, S., 1994: A Note on Distributed Computing. http:// research.sun.com/techrep/1994/smli_tr-94-29.pdf

Zimmermann, H., 1980: OSI Reference Model — The ISO Model of Architecture for Open Systems Interconnection, IEEE Transactions on Communications, vol. 28, no. 4, pp. 425 - 432.

COMMUNICATION, CONFIGURATION, APPLICATION
The Three Layer Concept for Plug-and-Produce

Uwe E. Zimmermann, Rainer Bischoff
KUKA Roboter Gmbh, Zugspitzstraße 140, 86165 Augsburg, Germany
UweZimmermann@kuka-roboter.de, RainerBischoff@kuka-roboter.de

Gerhard Grunwald, Georg Plank, Detlef Reintsema
German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Oberpfaffenhofen, 88234 Wessling, Germany
Gerhard.Grunwald@dlr.de, Georg.Plank@dlr.de, Detlef.Reintsema@dlr.de

Keywords: Plug&Play, robotics, reduced set-up time, determinism, real-time communication, device description.

Abstract: The Plug-And-Play is a synonym for the simple use of computer systems in office applications. This appealing idea also emerges in the branches of robotics and automation. But due to their specific technical and application driven requirements i.e. real-time, determinism, safety… a simple taking over of the technology is not sufficient. Plug-and-Produce extends the "easy to use" for applications in robotics and automation. This will lead to a dramatically reduction of set-up times of work-cells. In order to fulfil the various requirements a three layered Plug-and-Produce architecture is presented. The communication layer addresses all topics regarding the bus-system level; the configuration layer is responsible for the control and operation system level; and the application layer addresses the needs of the user and system programmer. A prototypical implementation is also presented.

1 INTRODUCTION

Technically, the integration of components i.e. a memory stick, a camera, or a disk drive into a computer system is a very complex task. Different kind of knowledge is required. This includes hardware, communication systems, operating systems, programming and application interfaces. Practically, the user wishes the simple addition of his new device without requiring reconfiguration or manual installation of device drivers. "Plug-And-Play"(PnP) is the synonym for this wish. In the early 1980s the NuBus (NuBus, 1983) which was originally developed by MIT and used by Apple Macintosh and Texas Instruments was one of the earliest PnP-busses. Today the Universal Serial Bus (USB) is the widest spread and probably best known PnP-bus (Axelson, 2005).

The Plug-And-Play technology was developed for the simple use of computer systems in office applications. Recently this appealing idea also emerged in the branches of robotics and automation. But due to their specific technical and application driven requirements (details see in section 2) a simple taking over of PnP is not possible. Hence the EU funded project *SMErobot*™ (SMErobot, 2005) created the term „Plug and Produce"(PnProduce) to express these peculiarities and to delimit from the office applications.

The potential of PnProduce is very obviously: The set-up time of robot cells is time-consuming and thus an important cost factor. It take days or even weeks just to get all peripheral systems communicating and working with each other, even though the system was planed and designed very detailed in advance. Costs could be dramatically reduced, if it is possible to reduce the installation time. Especially future robotics systems will require much more flexibility due to frequent task and equipment changes. It should be possible to connect a new, even a priori unknown device to the robot controller and to use it immediately without the need of a long and error prone manual configuration.

Besides the reduction of costs a further important factor is the "easy to use" feature. In future, more and more robots will find their way into small and medium enterprises, into the public and the health sector. Also service robots or assistants for entertainment and at home are emerging. These systems will be in common, that non-robotic experts

will have to deal with such complex systems. Adding of new tools, exchanging of faulty devices or upgrading the system has to be manageable without the need of an expert.

By means of scenarios and use cases section 2 motivates the details of PnProduce and specifies the inherent technical requirements. In section 3 we introduce the three layer concept of PnProduce and. Some first implementation results are presented in section 4.

2 PLUG AND PRODUCE REQUIREMENTS

PnProduce concepts for industrial applications especially robotics can adapt existing PnP concepts from IT. But there are some special issues that have to be covered and which are not solved by available solutions. By means of use cases some features of PnProduce are discussed and also the specific technical requirements.

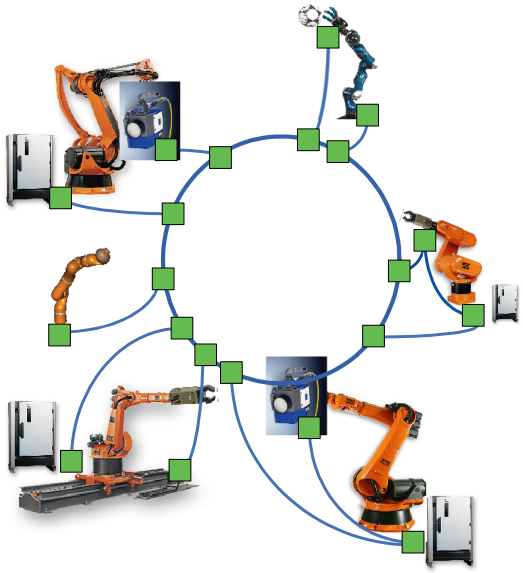


Figure 1: Scenarios for PnProduce.

2.1 Use Cases

Figure 1 shows a generic setup of a work cell with multiple, different robots, tools, grippers, hands, sensors, and controllers. They are connected via real-time bus systems i.e. EtherCAT, Sercos …. Also the single components inside a robot should be connected via PnProduce.

2.1.1 Device Exchange

A robot component i.e. a drive unit has to be exchanged or upgraded. The servicing staff powers off the robot, exchanges the devices. In case of PnProduce the robot has only to be switched on being operational again. The central control unit automatically identifies the new device, reads the actual device status information, configures it and integrates its functionality into the application.

2.1.2 (Automatic) Tool Change

Especially for versatile robot assistants frequent tool changes are a common requirement. A device driven approach is just a subversion of the previous use case, where a new tool (e.g. driller) is plugged to the robot and the associated tool service (e.g. “drill”) is available for the user.

In a service driven approach the user first selects the service (e.g. “drill”) and then the tool that could perform the task is automatically selected by the robot system itself. This means that an automatic tool changer system has to be set up.

2.1.3 Start Up of a New System/Application

In general an application can be characterized by the potential functionality of the connected components and devices. When powering on the functionalities of the devices are communicated with the central controller or a programmer. Bus, devices and robot controller are already configured for the immediate use. The application itself can be described by the detailed device configurations needed for running the task. This could be done manually or (semi-automatic).

2.1.4 Start Up of an Existing System/Application

An application as described above may be stored in a file. When switching on the system the controller gets all status information from the devices which are connected via the real-time bus. These data can be compared with the stored description of the planned application. The controller can verify whether all devices are connected and properly configured. If not appropriate feedback to the user can be given.

2.2 Deterministic and Real-time Communication

The communication in robotics and automation must guarantee some features to facilitate a proper and

safe operation of the system. One of the most important features are real-time and determinism. As there are multiple specifications on the term real-time, we will use in the context of PnProduce: "The timing constraints of the system must be guaranteed to be met. Guaranteeing timing behaviour requires that the communication is predictable". The term deterministic tightens the requirement in real-time communication. It is not sufficient to guarantee the data within a time frame. The data have to be delivered at a precise time.

2.3 Static and Dynamic Parameters

In the past, the application programmer differentiates cyclic and acyclic communication. Cyclic data are i.e. the sensor values from a force torque sensor or the nominal and real value of the robot joints. These data types are also called "process images" (CAN, 2007), that are a complete or partial local data base. The application or the devices operate only on this local data, whereas the bus system is responsible to keep the distributed data consistent. The advantage is that communication and application are highly decoupled. Another important factor is the deterministic exchange of data as the bus load is constant and could be analyzed a priori.

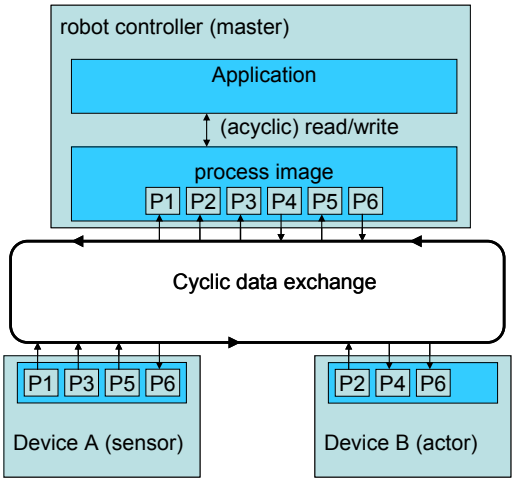


Figure 2: Cyclic/acyclic data exchange.

Also there is an acyclic communication regarding the application. Data types are commands or events. Commands are triggered by the application program and are explicitly programmed by the user. Examples are "gripper open" or "sensor on". They are typically for actuators and tools that are connected to the robot controller. The philosophy of commands is very similar to the Service Oriented Architecture (SOA) approaches. Most existing robot languages are command based and therefore PnProduce concepts could be integrated seamlessly.

Events are used for the asynchronous transfer of small data sets to inform the control system or the devices on their actual status. They may also be used for setting parameters of the attached devices and/or software.

The parameters may be classified in static and dynamic parameters. The latter may be set during system start-up or in case of configuration purposes. But they cannot or should not be changed during run-time. Thus there is no need to include these parameters in the process data sets. In general, static parameters are device specific. Dynamic parameters dynamically lead to more flexible systems. For example, if a gripper with an adjustable gripping force is used, the force parameter can be set once (configuration data) before the application starts or can be changed depending on what kind of object has to be gripped (process data).

Commands and events are still called acyclic even it is communicated in a cyclic manner on the bus level. The process images as well as the data exchange between the local data bases has to be configured, which is a complex task and leads to high demands for a PnProduce system. This is especially true if the data exchange should be dynamically changed.

2.4 Plug-And-Play Variants

There are different grades in the performance of Plug-And-Play systems. The simplest version is *Cold PnP:* All devices and components are switched off and get connected. Thereafter the entire system is switched on.

The most difficult variant is *Hot PnP*. The entire system is in operational mode. The user may remove or attach a device without further intervention and most important without disturbing the running application.

An intermediate level is *Coordinated PnP* which is a preliminary stage of Hot PnP. The chance nature of attaching/removing devices is eliminated. This guarantees that the application itself is not disturbed. The attaching/removal is user or program controlled.

Another classification scheme for PnP is complete, semi-automatic, or configurable.

Complete PnP:

- The attached device is automatically recognized and may be used without further intervention.

- Device descriptions are stored in a data base; drivers are automatically configured and integrated to the application system.

Semiautomatic PnP:
- The device is attached and device type/class is recognized.
- User has to integrate the appropriate drivers (CD, DVD, Internet …)
- No further configuration is needed

Configurable PnP:
- Additionally to semiautomatic the user has to configure the drivers, devices manually.

2.5 Abstraction and Descriptions

Hot-PnP, Complete-PnP, and the other variants presuppose the description of devices, configurations, and applications. The purpose is to encapsulate some device/configuration/application dependent functionality thus it can be used in a modular fashion.

Device Descriptions are a key technology for PnProduce. There are a several approaches which will be shortly presented.

The first is EDDL, the Electronic Data Definition Language and its predecessor the GSD, the Generic Station Description. Both of them are used in the PROFIBus Setting (EDDL, 2007).

The GSD is the obligatory “ID card” for every Profibus device. It contains the key data of the device, details relating to its communication capabilities and other information relating, for example, to diagnosis values. In the device integration process the GSD is sufficient to employ for the cyclic exchange of measurement values and manipulated variables between the field device and the automation system.

The EDDL is a textual and OS independent format to describe field devices. It gives hints on how to model the GUI, which is used to configure the devices. This results in a standardized GUI for devices of every manufacturer. The EDDL holds fields for metadata, such as ordering and maintenance information. The standardization is IEC 61804-2.

Another example for a description language is EDS and DCF. EDS stands for electronic data sheet and is used with the CAN bus, mainly with CANopen and DeviceNet. EDS is just a template, DCF, device configuration file, is a concrete instance of this template.

EDS is also a plain ASCII or XML file, standardized in ISO 15745. By means of an EDS, a device can be described with respect to the content of its object dictionary. User defined data types (records), their value(s), simple values of predefined type, arrays, "funktion pointers” and similar data is stored there and therefore described in the EDS. There are a lot of EDS Files predefined for devices supporting the CAN bus (CAN, 2007).

FDCML is the field device configuration markup language. It describes identity, logical and physical communication facilities (bus independence), functionality, and configuration facilities. Also it supports the multilingual documentation of the device. It's flexible in respect to future developments and is capable of describing dependencies between device parameters. It is possible to describe the different types of devices which can be connected to this device, the needed resources to use this device, its structure and so on (FDCML, 2007).

The descriptions used in UPnP are written in XML syntax (therefore again multilingualism possible). UPnP describes identity, provided services (functional units within devices), actions (functional units within services) and state variables related to these services. For each service the description contains the type and URI's for eventing and controlling. The device itself contains a URI for its presentation. With these URIs it is possible to interact with the service and to examine the device (UPnP, 2007).

The actions are parameterized with supplied arguments, which are defined within the service. These arguments can be in or out arguments, relate to any state variable within the service and have a defined return value.

The state variables are typed, have a value range and can send events. It is a very abstract approach, so it doesn't include any description of physical connections and therefore bus-independent.

Used in our setting the abstract approach and the missing definition of the physical connection display the need for extensions of the description.

Configuration descriptions are another category of *descriptions*. They describe how a unique instance of a device is actually used. Configuration descriptions are only valid for exactly one device instance and often are dependant from a single application. A configuration description is more or less just a “snapshot” of the actual device state, e.g. the values of the static parameters. They can be used to make device data persistent, to check if the actual configuration is valid and for automatic configuration of devices after a device exchange.

The application description contains the devices and services needed to run an application. It is used

to check, if all conditions are met to start the application or if there are modifications regarding the connected devices.

3 THREE LAYER CONCEPT

It is obviously to see that PnProduce has different levels of abstraction. You can identify three "Plug-And-Produce"-layers to meet the requirements of robotics and automation systems:

1. Communication and bus-system level: log-in and log-out of subscribers;
2. Control and operation system level: integration and release of components in the environment;
3. Application level: the offered functionalities of the subscribers are used.

The German national project PAPAS: Plug And Play for Automation Systems (PAPAS, 2006) addressed this complexity and suggested a layered structure similar to the famous ISI/OSI layers for communication. Figure 3 relates the PAPAS structure both to the ISO/OSI model and the typical field bus approach.

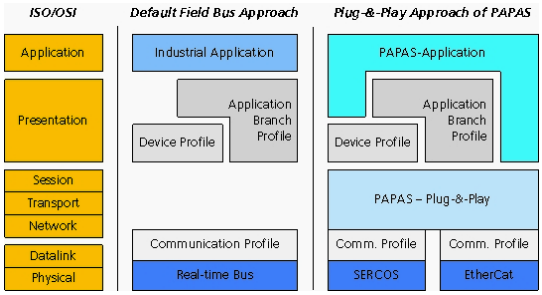


Figure 3: The PAPAS Plug-And-Play layer architecture (Plank et.al. 2006).

The Datalink and Physical Layer are directly related to the communication bus and its physics. Examples of a field-bus are CAN or Profibus. PAPAS investigated among others Sercos (SERCOS, 2007) and the Ethernet-based real-time busses EtherCAT (ETG, 2007) and Ethernet-Powerlink (EPSG, 2007). The two layers are the only, clear delimited layers regarding the ISO/OSI specification. The layers Session, Transport, and Network are not used in the general field-bus approaches. In all three models the application and presentation layers are oriented towards the application and/or the configuration of the application. The limits of these ISO/OSI layers are indistinct.

3.1 Communication PnProduce

The basis of PnProduce is the communication layer, as it addresses the potential (i.e. bandwidth, load, determinism, clock …) of the communication itself. It is directly related to Layer 1 to Layer 5 of the ISO/OSI reference model. The PAPAS Plug&Play layer addresses here the PnP-functionality regarding the bus-systems and their specific communication protocols. At this level the independency of the bus-system is implemented.

When these layers are successfully switched on the systems knows all the participants by its network address but not by its function. The transferring data types are not known yet which guarantees the independence of the application.

3.2 Configuration PnProduce

The effect of this layer is twofold. It configures the system with regard to the communication as well as to the application.

When the communication system is operational, the participants are known and the master is able to read their device descriptions (see 2.5). Typical device information is:

- Synchronous, asynchronous communication
- Minimal/maximum bandwidth for sending and/or receiving data i.e. joint value, force/torque value, image data …
- Minimal/maximum clock for sending and/or receiving data

According to their requirements and the needs of the specific application the communication system can be configured.

Analogue the devices and controls have to be configured regarding their functionality. Following tasks can/should be done in this layer:

- Determining which devices i.e. sensor are available;
- Determining which functions, services, and parameters are available (see device descriptions);
- Automatic reconfiguration of a device after it has been exchanged because the old one was defect;
- Reading device status information i.e. error log, software release, operating hours, and other statistical data.

3.3 Application PnProduce

The intention of this layer is to structure the open ended world of applications thus you can reuse robot programs or tasks (synonym for robot applications)

in another context or environment. A robotic task could be i.e. to “drill a hole”. This task is composed by a robot, a sensor, a drilling machine, a piece of wood, and software commanding the active devices. You can repeat the task with the same components but also with i.e. another robot. The application remains the same. Only the functionality of the connected devices is of interest.

The following scenarios describe a graded structure, which can be distinguished within Application PnProduce.

- Standardised Functions/Services: If functions, services and parameters are standardised like CANopen profiles (Pfeiffer et.al. 2003, CAN, 2007), they can be used automatically and called from the system or the application.
- User integration: If a new device with an unknown functionality is attached, the system is able to autonomously detect and integrate it at the communication level. But it is not able to integrate its services or functions. It will inform the user on the new component its offered functionalities.
- Semantic information: Another possibility is to integrate semantic knowledge into unknown functionality. But this is a very difficult and not completely solved task. Besides the syntactic description a service the meaning must also be represented.

Whereas the application PnProduce layer is fully independent from the communication PnProduce layer and vice versa, the configuration PnProduce layer can have dependencies with both, communication and application.

3.4 PnProduce Layer Model

The following will explain the PnProduce layers and their interdependencies in more detail.

The new high performance communication systems must meet the modern robotics and automation demands:

- Distributed automation
- Close and coordinated coupling of devices with different functions, timing, and data loads
- High clock speed
- Short reaction time
- Reduction of costs

The actual serial real-time bus systems (i.e. EtherCAT, Ethernet-Powerlink, and Sercos) comply with these demands. But they differ in transfer modes and protocols. Thus the integration of different devices equipped with different bus systems into one robotics application is very difficult and needs a lot of expert knowledge. In order to reach the Communication PnProduce functionality and thus a transparent and deterministic access to the bus system level, an abstraction of the PnP-behaviour from the bus-system is needed.

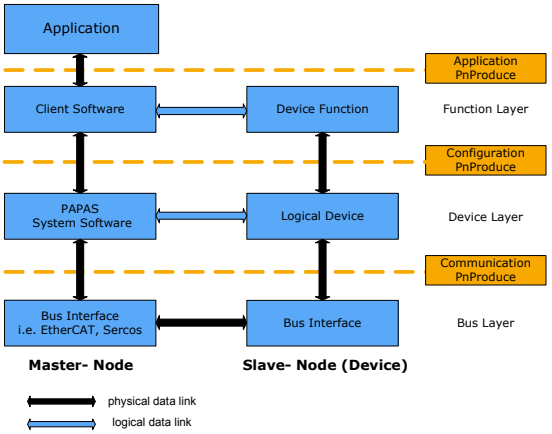


Figure 4: The PnProduce layer model. Each transition interface provides a specific kind of PnProduce-capability.

PnProduce assumes a master/slave architecture which is commonly used in robotic and automation systems. The PnProduce layer model (Figure 4) shows the most important components and the data flow.

- Master-Node: i.e. robot controller;
- Slave-Node: i.e. sensor device in its entirety;
- Device Function: description of the input/output characteristics i.e. sensor values;
- Client Software: API of the device;
- Logical Device: Basic interface, which facilitates the master an unified handling of the varying devices;
- PAPAS System Software: Specific w.r.t. the operation system but independent of the client software and the bus interface;
- Bus-Interface: hardware i.e. EtherCAT, Sercos

Master and slave are composed of three layers. The Bus Layer provides the physical and packet oriented connection. The Device Layer contains all the system information needed for the device configuration. The Function Layer describes the logical interconnection between master and slave.

The master puts a *request* on the bus. If the needed communication resources are available either the master or the slave sends its data. The success is indicated by *acknowledge*. The PnProduce protocol provides four data transfer modes:

1. *Control*: configuration of new attached devices and device control. All devices have to support this mode.
2. *Interrupt*: Asynchronous data transfer of small data sets. The protocol guarantees the data transfer in device specific service intervals.
3. *Bulk*: Asynchronous data transfer of large data sets. The protocol guarantees the data transfer as soon as the needed bandwidth is available.
4. *Isochronous*: This mode guarantees fixed data rates and a maximum latency. When a new device is attached the master verifies whether the requested communication bandwidth is available.

4 EXPERIMENTAL SYSTEMS

This section describes in short the first version of an implemented PnProduce system. Here the focus is mainly on the communication PnProduce functionality. The application addressed within the PAPAS experimental setup is based on a robot assistant for human working in industrial environments (see Figure 5). One chosen task of the robot is a typical example of an industrial manufacturing process, which is still executed almost completely manually today, with a very low degree of automation. The assembly algorithms developed and tested on a set of planar objects with complex, non-convex geometric forms are described within (Stemmer et.al, 2006).

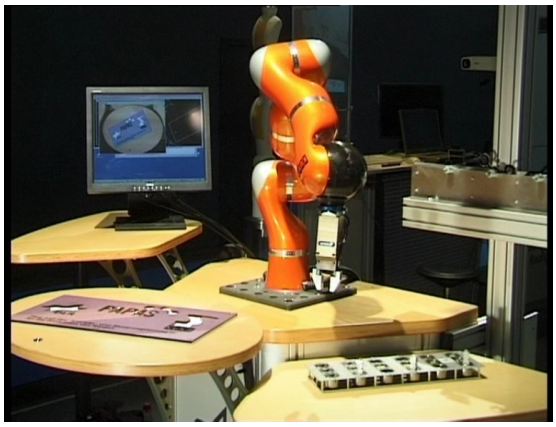


Figure 5: A light-weight robot arm as an assistant for human-like working in industrial environments the PAPAS experimental setup is designed to demonstrate a typical example of an industrial manufacturing process.

For verification of the PnProduce approach a set of typical industrial tools has been prepared to demonstrate hot plugging. Typical tools are a Programmable Focusing Optics (PFO), a compliant force-torque sensor or a parallel gripper (see Figure 6). The PFO of Trumpf Laser flexible welds spots and seams without moving the workpiece, neither the focusing optics having to be moved at all. The PFO focuses the laser beam to any defined location in the working area.

Figure 6: Typical classes of slave tools and devices for robotics and automation applications: a gripper with parallel jaw motion, a force torque sensor and a programmable focusing optics (PFO) which have been prepared to interact as an EtherCAT slave device.

Especially for assembly and part mating with industrial robots, a compliant force-torque sensor was developed. The compliant sensor does not only yield the forces and torques but also the positional/rotational displacements of a tool, e.g., under the influence of gravity.

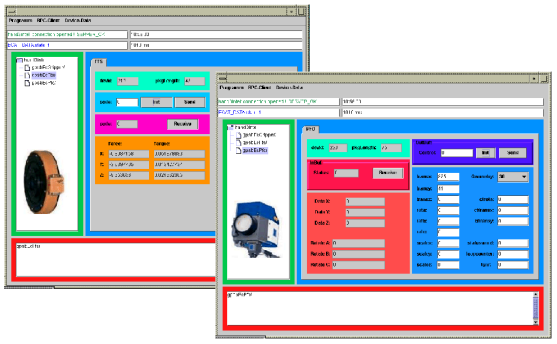


Figure 7: JAVA-based tools can be used to monitor the plug-and-play process, to display recorded device data and to command plugged devices using a general purpose class-specific interface.

With the *Ethernet Control Automation Technology* (EtherCAT) a new Ethernet-based field bus system was chosen at the drive or I/O level (ETG, 2007) for real-time transportation of process data.

The experimental set up uses a line structure configuration. A chain of EtherCAT slaves can be connected in any sequence to the EtherCAT master which monitors the Ethernet bus for plugged devices. Inserting a new device into an expansion slot forces the master to trigger the PnProduce process of the overlaying PnProduce layer without

requiring reconfiguration or manual installation of device drivers.

5 SUMMARY AND OUTLOOK

The integration of industrial components i.e. gripper, force/torque sensors, intelligent tools… into a robot work-cell is a very complex task. Different kind of knowledge is required. This includes hardware, communication systems, operating systems, programming and application interfaces. Plug-and-Produce extends the very well known concept of Plug-And-Play towards the use in industrial environments. Due to their specific technical and application driven requirements i.e. real-time, determinism, safety… a simple taking over of the PnP technology is not sufficient. Plug-and-Produce extends the "easy to use" for applications in robotics and automation. This will lead among other things to a dramatically reduction of set-up times of work-cells.

In order to fulfil the various requirements a layered Plug-and-Produce architecture was presented. The key feature is to properly encapsulate the functionalities of one layer with regard to the neighbouring. Standardized interfaces i.e. the device description will permit the interchangeability of hardware and software

Three "Plug-And-Produce"-layers were specified to meet the requirements of robotics and automation systems:

1. Communication and bus-system level: log-in and log-out of subscribers;
2. Control and operation system level: integration and release of components in the environment;
3. Application level: the offered functionalities of the subscribers are used.

The first implementation results especially the Communication PnProduce are very promising. The future activities of Application PnProduce may be influenced by the research in Service Oriented Architectures (SoA). Here we have to investigate whether these concepts satisfy the PnProduce requirements and how they can be integrated.

ACKNOWLEDGEMENTS

This work has been partially funded by the German Collaborative project PAPAS under grant no. 02PH2060 and the European Commission's Sixth Framework Programme under grant no. 011838 as part of the Integrated Project SMErobot™.

REFERENCES

NuBus, 1983. *NuBus Specification,* Texas Instruments.

Axelson, J. 2005. *USB Complete: Everything You Need to Develop Custom USB Peripherals*. Lakeview Research, Madison WI.

SMErobot, 2005. *SMErobot™* - The European Robot Initiative for Strengthening the Competitiveness of SMEs in Manufacturing, www.smerobot.org, Integrated project funded under the European Union's Sixth Framework Programme (FP6).

CAN, 2007. *Controller Aera Network*, www.can-cia.de.

EDDL, 2007. *The Electronic Device Description Language (EDDL)*, www.eddl.org.

FDCML, 2007. *Field Device Configuration Markup Language,* www.fdcml.org.

UPnP, 2007. *Universal Plug and Play*, The UPnP™ Implementers Corporation, www.upnp-ic.org.

PAPAS, 2006. *Plug-And-Play Antriebs- und Steuerungskonzepte für die Produktion von morgen* (in German), www.robotic.de/PAPAS.

Plank, G., Reintsema, D., Grunwald, G., Otter, M., Kurze, M., Löhning, M., Reiner, M., Zimmermann, U., Schreiber, G., Weiss, M., Bischoff, R., Fellhauer, B., Notheis, T., Barklage, T. 2006. PAPAS Abschlussbericht (in german), http://www.dlr.de/rm-neu/Portaldata/52/Resources/dokumente/papas/PAPAS-Abschlussbericht.pdf

ETG, 2007. *EtherCAT Technology Group: „Technology Introduction and Overview"*, http://www.ethercat.org.

SERCOS, 2007. *SERCOS interface*, www.sercos.de.

EPSG, 2007. *Ethernet POWERLINK Standardization Group,* www.ethernet-powerlink.org.

Pfeiffer, O., Ayre, A., Keydel, C. 2003. *Embedded Networking with CAN and CANopen.* RTC Books

Stemmer, A., Schreiber, G., Arbter, K., and Albu-Schäffer, A. 2006. *„Robust Assembly of Complex Shaped Planar Parts Using Vision and Force"*, in IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI2006), 3.-6. September 2006, Heidelberg, Germany.

TOWARD ONTOLOGIES AND SERVICES FOR ASSISTING INDUSTRIAL ROBOT SETUP AND INSTRUCTION

Mathias Haage, Anders Nilsson and Pierre Nugues
Department of Computer Science, Lund University, Box 118, 22100 Lund, Sweden
{*mathias.haage, anders.nilsson, pierre.nugues*}*@cs.lth.se*

Keywords: Industrial robot, ontology, web services, multimodal dialogue.

Abstract: Achieving rapid, intuitive, and error-free robot setup and instruction is a challenge. We present our work towards an assistive infrastructure for robot setup and instruction that attempts to address it. In this paper, we describe the ongoing development of a system that automatically generates multimodal dialogue interaction from product and process ontologies. The prototype currently generates two modalities, digital paper and spoken dialogue.

1 INTRODUCTION

Traditional robotics supports long-batch production and requires skilled personnel to handle setup and instruction. On the contrary, new robot markets often involve shorter series and small and medium enterprises. This means that the shift of products is faster and the change-overs often need to be carried out by non-experts. This sets new challenges for the robot user interfaces to be more intuitive and user friendly in order to reduce number of errors and cost/time (Haegele, 2007). Such challenges outline the need for assistive systems within the robot cell to make the operator less dependent on expert knowledge and turn complex tasks feasible. Examples that could benefit from assistance include calibration of tools, fixtures, and workpieces; usage of CAD/CAPP/CAM software such as task planners; configuration of process-specific software packages, such as the ABB palletizing PowerPac.

In this paper, we present an assistive infrastructure for robot setup and instruction that attempts to address these challenges. We introduce the ongoing development of a system based on semantic web technologies that automatically generates multimodal dialogue interaction. We also describe a prototype that currently generates two modalities, digital paper and spoken dialogue.

The purpose of an assistive system is to enhance the usability and usefulness of the robot and its connected resources (sensors, CAD/CAPP/CAM systems) through:

- The use of semantic standards in information exchange, such as RDF/S, OWL, SPARQL, and SWRL;
- Production documents such as product and process data including a semantic layer;
- The definition of compatible semantic layers so that they can be used across the relevant tasks, such as aiding cell calibration and robot instruction.
- Increased automation of tasks using the semantic layer, such as finding calibration sequences to make sure nothing is forgotten.

The roadmap we follow to implement the assistive infrastructure is based on the use of an ontological network to encapsulate knowledge about the product data and manufacturing processes. It requires the derivation of ontology concepts that will serve as the main data source to generate — or refer to — the complete specifications and the operating instructions used to automate information management necessary for task planning and execution.

2 MULTIMODAL FORM-BASED DIALOGUE

From the user viewpoint, the operation starts with an initial selection command from the operator to tell the machine which work piece to produce and possibly

from positions and equipment data sensed by devices on the floor.

After the initial selection, the system extracts data from the ontology that enables the operator to configure the task and the product, and to prepare the task execution. The configuration step uses a multimodal interface that lets the operator fill in the different options. It ends with the monitoring and execution of the configured task.

The process flow uses conversion tools such as transformation rules, inference rules, and the JastAdd compiler (JastAdd, 2007) to select and convert portions of the ontology. This results in process and product data divided into configurable and nonconfigurable parts (Figure 1). The extracted data are first formatted as an XML document corresponding to a production sketch that we call the XML appconf.

From this sketch, the system automatically generates user interfaces with multiple input modalities for all the parameters. A first demonstration prototype will be available in Spring 2008.

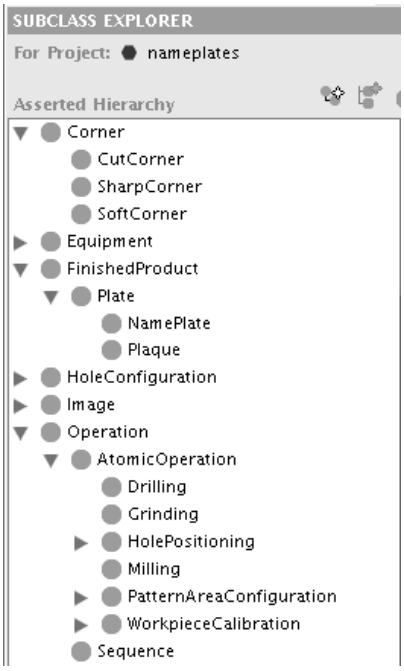


Figure 2: An excerpt of the ontology detailing the operation hierarchy.

3 ONTOLOGY MODELING

In computer science, ontologies correspond to hierarchical models to represent concepts, objects, and their relationships. They enable systems to (Buitelaar, 2007):

- Encode and interpret data using a rich hierarchical and relational structure.
- Extract data and integrate them into applications.
- Share data with a common format.

As ontology modeling language, we have chosen the web ontology language (McGuinness and van Harmelen, 2007) and the Protégé toolkit (Protégé, 2007) as a data entry and validation tool. Both are well-established standards in their domain with a large developer's community. We are currently using them to build the ontology of a specific domain shown in Figure 2 that serves in the demonstration prototype. This ontology acts as an advanced data repository for the product configuration and the production operations. In the future, we will populate ontologies from manual modeling, specification databases, and 3D models. In addition to what we develop within SMErobot, the data model we will use could also possibly benefit from work being carried out at LTH for the SIARAS project (SIARAS, 2007) on production ontologies.

The conversion pipeline shown in Figure 1 uses W3C recommendations associated with the semantic web such as XSLT or SWRL. The choice of these tools needs some clarification. We first summarize the concepts that are around OWL and then explain the conversion principles.

3.1 Resource Description Framework - RDF

OWL is based on the resource description framework, RDF (RDF, 2007). RDF models statements as triples in the form of a subject, a predicate (a verb), and an object. As an example, the statement *the milling process starts with a calibration* can be split into a subject, *the milling process*, a predicate, *starts with*, and an object, *a calibration*. Such triples are also named, respectively, the resource, the property, and the value. RDF is restricted to binary predicates.

This framework can use two encodings. The first one, called Notation 3, consists of sequences of textual triples and the second one adopts an XML syntax. The subject – the resource – must be a URI. The predicate or property, which is also a resource, is a URI too. The object or value can be a resource or a literal. To represent the example above, we use the lrc namespace – Lund Robotics Core – and URI http://cs.lth.se/lrc/ontologies/1.0/. This URI is still nonexistent when this article is being written. Using Notation 3, we can represent the example above as:

```
@prefix lrc: <http://cs.lth.se/lrc/ontologies/1.0/>.
<#milling_process> lrc:starts_with "calibration";
```

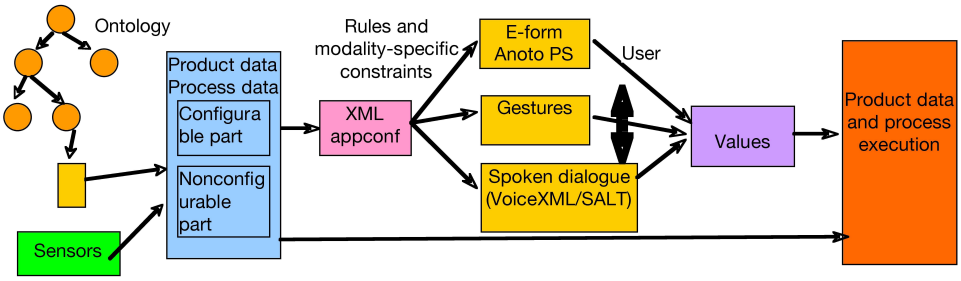


Figure 1: The process workflow.

And in XML syntax as:

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:lrc="http://cs.lth.se/lrc/ontologies/1.0/">
  <rdf:Description rdf:about="#milling_process">
    <lrc:starts_with>calibration</lrc:starts_with>
  </rdf:Description>
</rdf:RDF>
```

More generally, triples (subject, predicate, object) can be encoded in Prolog or Datalog as *predicate(subject, object)* that is, with the sentence above as:

```
starts_with(milling_process, calibration)
```

3.2 RDF Schema – RDFS

RDF schema, RDFS, is built on RDF and defines two predicates that enable the programmer to build an ontology: *rdfs:Class* and *rdfs:subClassOf* (RDFS, 2007). The *rdfs:Class* element allows to declare a RDF resource as a class and the *rdfs:subClassOf* element allows to declare subclasses of a class and build a hierarchy. When a resource has been declared as a class, we can use *rdf:type* to create individuals member of this class. In addition, RDFS comprises similar predicates to build a property hierarchy.

In addition, RDFS has constructs to type the subject and the object of the triples. It corresponds to the domain and range of a function, and in the case of RDFS applies to properties. RFDS uses the constructs *rdfs:domain* for the subjects and *rdfs:range* for the objects to restrict the values of the two arguments of a property.

3.3 OWL

Although the combination of RDF and RDFS forms an ontology language, it lacks some features to build large, realistic ontologies. They include cardinality restrictions, Boolean operations on classes, etc. In addition, RDF and RDFS are not well coupled to logic and reasoning tools.

The web ontology language, OWL, is an extension of RDF and RDFS that attempts to complement them with better logic foundations and a support for practical reasoning (McGuinness and van Harmelen, 2007). It comes with three flavors of increasing expressivity – light, description logic, and full – that are upward compatible. Only the two first ones are guaranteed to be tractable in practice.

Important constructs of OWL include the *owl:Class*, which is derived from the *rdfs:Class*, two properties, *owl:ObjectProperty* and *owl:DatatypeProperty*, that relate objects to respectively another object or a data type value like a string, an integer, a float, etc., *owl:Restriction* that enables the programmer to use existential and universal quantifiers and cardinality.

4 PROTOTYPE SETUP

4.1 Nameplate Manufacturing

As described in (SMErobot, 2007b), the ontology programming approach uses automatically generated forms to select and configure both the task and the product. The prototype selected to demonstrate the concepts associated with our approach corresponds to the manufacturing of wood nameplates.

Figure 3 shows a configuration form where the left column configures the shape and looks of the plate. The right column configures the process for manufacturing the plate. In this example it is possible to skip steps, execute in a stepwise manner, and choose data acquisition methods for steps involved. The left column can be filled out at an earlier date while the right column is filled out close to task execution time. The upper right barcode identifies the process and is possibly unique to individualize the sheet.

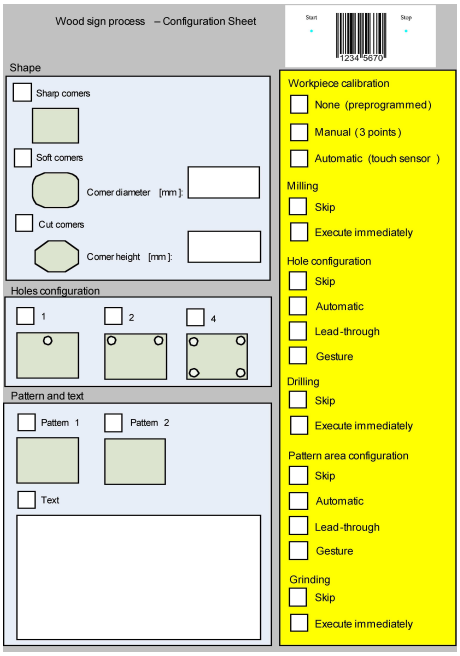


Figure 3: Prototype form for manufacturing wood nameplates.

4.2 Nameplate Manufacturing Ontology

We have built a prototype ontology to encode the process templates and we are developing well-defined conceptual interfaces toward work cells (equipment, capabilities, communication) and process data, assisting construction of process templates, and assisting (manual/automatic) work cell reconfiguration.

The ontology is restricted to the prototype domain and Figure 2 shows an excerpt of it. It describes the finished products its components, together with possible features as well as the operations involved in the manufacturing process of the product.

4.3 Intermediate Appconf Representation

Given a product to manufacture, the conversion process extracts an intermediate, flat representation from the ontology. This representation is designed to be modality-independent, which makes it easier to build user views (forms, speech, gestures). We call it the application configuration – appconf.

As an example, in the application prototype we are building, the sheet requires the operator to supply data, such as the corner shape, the hole configuration, and the pattern and text (person name for instance). All these items are shown as input areas on the sheet in Figure 3. The intermediate appconf representation has corresponding elements representing all these configurable items, for instance the corner shape.

We give an idea of how to represent the corner shape options in the XML code snippet below. This code replicates the possible options, sharp, soft, or cut corners, with the images to display in the e-form using the *img* element and the messages to utter using the *snd* element in the case of a spoken dialogue.

```
<shape>
    <one-of>
        <option>
            <name>sharp corners</name>
            <command code="sharp.cd"/>
            <img src="sharp.jpg"/>
            <snd src="sharp.wav"/>
        </option>
        <option>
            <name>soft corners</name>
            <command code="soft.cd"/>
            <img src="soft"/>
            <param name="diameter" unit="mm"/>
        </option>
        <option>
            <name>cut corners</name>
            <command code=".cd"/>
            <img src="cut.jpg"/>
            <param name="height" unit="mm"/>
        </option>
    </one-of>
</shape>
```

Using this configuration sketch, presentation rules, and modality specific constraints, the conversion process produces displayable forms or spoken dialogues so that the operator can supply the missing parameters. Once the operator has filled in the data, the corresponding XML fragment is:

```
<shape>
    <name>sharp corners</name>
    <command code="sharp.cd"/>
    <img src="sharp.jpg"/>
    <snd src="sharp.wav"/>
</shape>
```

4.4 Methods and Languages to Extract Information from Ontologies

The conversion pipeline extracts and infers information from the ontology and generates the user input modalities. The appconf configuration sketch is an XML intermediate document between the ontology and the user interfaces. Unlike the sketch, the ontology is a structured and hierarchical representation, where features are shared and inherited across a variety of pieces and processes. This means that extraction is not trivial because the representation lan-

guages involve three complex and intricate layers: RDF, RDFS, and OWL.

Such a setting requires specific query languages and techniques. In addition to the ontology management, we need to process other XML documents in the processing chain such as the appconf sheet to convert them into forms or dialogue programs. We review here techniques and their application in the management of information along the conversion chain. They include accessing XML nodes, querying RDF triples, and reasoning about the ontology knowledge. Most difficulties come from the apparent masses of "solutions". Wikipedia lists not less than 11 different RDF query languages and ten OWL reasoners! We focus here on what have become the (likely) standards in their respective ecosystems.

4.4.1 XSLT

The simplest way to access and transform XML documents is to use the combination of XPath and the extensible stylesheet language transformations, XSLT (XSLT, 2007). XPath enables programmers to express a path and access nodes in an XML tree, while XSLT defines conversion rules to apply to the nodes. A typical application of XSLT is the transformation of XML documents into XHTML files destined to be read by web browsers.

Provided that the amount of paraphrasing (syntactic variation) is limited, XSLT XPath is fairly usable to run the conversions. From studies we have done, this is the case for the conversion of the appconf sketch to the user modalities. We are completing the implementation and integrating it in the prototype.

However, this is not the case for ontologies. They are built on OWL, which is built with RDF triples, which allows reformulating similar structures using different constructs. Querying ontologies require either query languages or reasoning rules. For a justification, see (Antoniou and van Harlmelen, 2004), pp. 100-102.

4.4.2 SPARQL

SPARQL (SPARQL, 2007) is a RDF query language. It enables the programmer to extract RDF triples using the SELECT keyword where the variables are denoted with a question mark prefix using a set of conditions defined by the WHERE keyword. It is also possible to build a new graph using the CONSTRUCT keyword. SPARQL's syntax is similar to that of the SQL language. The query below extracts all the pairs where *?subject* is a subclass of *?object*:

```
SELECT ?subject ?object
WHERE {
    ?subject rdfs:subClassOf ?object. }
```

However, as SPARQL makes the join operation implicit, it bears some resemblance with Prolog as in this query:

```
SELECT ?subject ?config
WHERE {
    ?subject rdfs:subClassOf <#FinishedProduct> .
    ?prop rdf:type owl:ObjectProperty .
    ?prop rdfs:range ?config . }
```

SPARQL is becoming a de facto standard for RDF. It is a stable language with a quality implementations from various sources. Competitors include XQuery, a generic XML query language, which has not gained acceptance in the RDF community.

4.4.3 SWRL

While SPARQL enables a programmer to extract information from an ontology, it is only designed for RDF. In addition, it cannot easily derive logical consequences from its results. To exploit fully the ontology knowledge, one needs a reasoning or inferencing mechanism. This is the purpose of a language like the semantic web rule language, SWRL (SWRL, 2007). SWRL rules have a Prolog-like structure. They consist of an antecedent corresponding to a conjunction of conditions (predicates) and a consequent. When the conditions are true, the consequent is also true and can be asserted. In addition to being a inference language, SWRL features an extension that lets it act like a query language, SQWRL.

SWRL is supported from the 3.4 version of Protégé in the form of a development environment with editing tools. This means that we can write, modify, and to a certain extent validate rules. However, version 3.4 is still in the beta stage at the time we are witing this paper. In addition, Protégé does not include a full-fledged inference engine. This means that it cannot by itself execute the rules. It just supplies a bridge that connects to an external module. So far, Protégé supports only one inference engine, Jess (Friedman-Hill, 2007).

4.4.4 JastAdd

JastAdd is not a query language in itself, but a general compiler construction tool with some very useful features; aspect oriented programming and attribute grammars. Using results from earlier work (Malec et al., 2007) we can automatically create a parser for an OWL ontology. Utilizing the aspect-oriented feature of JastAdd, we can then implement queries in the form of aspect modules that will be weaved in with the generated parser at compile time.

While it does not possess the expressive power of SWRL, it will enable the user to extract almost any information from the ontology with just a few 10s of lines of Java code.

4.4.5 Prolog

Prolog – or Datalog – is a last example of reasoning tool that could be used to extract information from the ontology. Some Prolog implementations have a RDF interface like SWI Prolog that has been used with success in semantic web applications (Wielemaker et al., 2007). It is then possible to query an ontology from a logic program and to run inference rules on the result.

As Prolog predicates and rules have much in common with SWRL, logic programs written in Prolog and SWRL would be very similar and with equivalent performances. Difference would come from the location of the bridge between the ontology and the inference engine, at the RDF level for Prolog, at the OWL level for SWRL.

However, although Prolog is more expressive than other languages and has a proven record of industrial applications, Protégé does not support it. It is not standardized within the context of the semantic web either. This makes its choice, at the moment, riskier than the other options.

4.5 From an Ontology to the Appconf Sketch

The first step of the conversion pipeline generates the XML Appconf sketch from the ontology. As the SWRL formalism is more flexible and powerful, as well as adopted by the semantic web community, we are using it for this step in the demonstration prototype. As inference engine, we are using the built-in bridge that is for now only coupled to Jess.

However, SWRL is a new feature of Protégé 3.4 and athough it already supports many *abox* and *tbox* built-in predicates, it is still under active development. Many of the predicates are not yet implemented. The beta version status of SWRL pose timetable problems and we are also using SPARQL to query the ontology and write the rules.

We wrote rules using both formalisms to extract information from the ontology. We show below an example of SWRL rule that finds all the properties of the subclasses of FinishedProduct and Figure 4 shows a screenshot of the editing window in Protégé. We also show its counterpart in SPARQL. We embedded the rules in the Java prototype using the Protégé API that resembles SQL drivers.

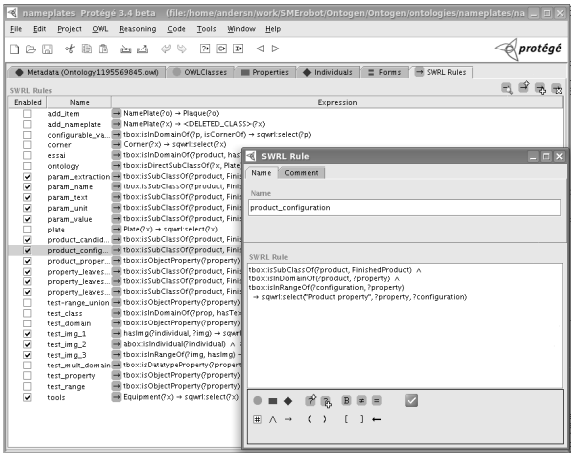


Figure 4: A screenshot of the SWRL interface in Protégé and an example of a rule.

```
PREFIX list: <http://jena.hpl.hp.com/ARQ/list#>
SELECT ?product ?configuration
WHERE {
?product rdfs:subClassOf <#FinishedProduct> .
?property rdf:type owl:ObjectProperty .
{{?property rdfs:domain ?product} UNION
{?property rdfs:domain ?union .
?union owl:unionOf ?list .
?list list:member ?product}} .
?property rdfs:range ?configuration}
```

4.6 From the Appconf Sketch to Input Modalities

The second step of the conversion pipeline generates user input interfaces from the XML Appconf sketch. As final products frequently need to be customized according to the order, the manufacturing operator will be able to enter a part of the specifications at production time. In the demonstration prototype, we will investigate three configuration modalities that are core to the SMErobot project (SMErobot, 2007a), namely E-forms, gestures, and spoken dialogue.

We are developing tools to generate automatically the work piece production forms, the gesture tracking and interpretation module, or the dialogue specifications from the XML Appconf. Before the piece is manufactured, the operator fills in the remaining data corresponding to the final piece using the modality of her/his choice. To ease the interaction, we are investigating a framework to combine simultaneously the different modalities so that the operator can use speech and gestures at the same time for instance.

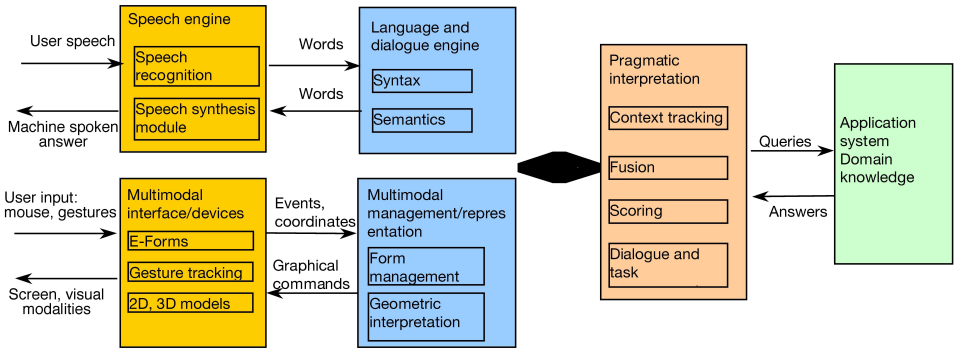


Figure 5: Multimodal dialogue architecture.

4.6.1 Transformation Language

As transformation language to produce the forms and the dialogue specifications, we are using XSLT. XSLT enables to apply transformations to Appconf nodes accessed via the XPath language. It can produce XML documents in formats like XHTML for the forms or VoiceXML for the dialogues. In addition, we are investigating the XSL-FO page-formatting standard where the description of a document content uses objects such as blocks, tables, footnotes, etc. It is richer than HTML-like descriptions and can be converted to PDF. The conversion of an XSL-FO document to a PDF uses a sequence of transformations that builds the XML tree, produces graphical objects, renders the objects as text areas with their pixel positioning, and finally generates PDF.

4.6.2 VoiceXML

The demonstration prototype includes a voice modality that enables the operator to configure the product through a dialogue and hence have his hands free while s/he fills in the manufacturing options. The dialogue uses a system-initiative scheme, which means that the dialogue structure resemble a form filling procedure where the user answers questions posed by the system.

We have chosen the Voice Extensible Markup Language, VoiceXML, to generate the dialogues. (VoiceXML, 2007) is markup language that enables a programmer to build form-based, goal-oriented dialogues (system-initiative mostly). The user fills fields in forms using speech, where the field input can be constrained with a grammar. It is designed to be integrated in a speech server and supports IP telephony. As VoiceXML is a standard, the programs should be portable to any platform that supports this language.

4.7 Perspectives: Multimodal Management

We will examine possible designs for the multimodal management architecture such as the one shown in Figure 5. It is mainly derived from a previous work of a member of the LTH team (Bersot et al., 1998; Godéreaux et al., 1998) and a recent system developed by the Bell labs (Ammicht et al., 2007; Potamianos et al., 2007). One input channel corresponds to the speech recognition module, which transcribes the user's speech into a word stream. The language engine then processes the character flow dealing with syntax, which is constrained by the VoiceXML structure, and semantics. The semantic module converts words into a semantic representation that is common to speech and other types of interaction. The other channel corresponds to form filling, which are processed by the multimodal manager. The pragmatic module merges data from both modalities and keeps track of the context and the application goals. The resulting answers are either converted into speech to the user by a speech synthesizer or presented visually by a visualizing module.

The multimodal architecture will use a client-server architecture and instantiate some of the modules shown in Figure 5. We are currently defining them. In the future, it could evolve into an integration platform enabling partners to plug their applications. As a use case we consider interactions where the user fills in the data using one modality, the corresponding client sends the data to the server, and the server updates the context of all the modalities. There are then continuous visual or audio updates of the current context. For instance, an audio message is synthesized each time the user has selected an option with the form, to confirm or remind the next actions. Modality switching could be carried out manually by the user or automatically.

5 CONCLUSIONS

We have described the design and implemention of an assistive infrastructure based on the use of an ontological network to encapsulate knowledge on the product data and manufacturing processes. We have implemented a prototype ontology that serves as the main data source to automatically generate digital forms and voice dialogues to configure a wood nameplate manufacturing process. As a perspective, we intend to synchronize modalities for a more flexible, efficient user input. A first prototype will be available for demonstration in Spring 2008.

ACKNOWLEDGEMENTS

The work presented in this paper is being financed mainly by the EU FP6 projects SMERobot and SIARAS. Partner companies within SMErobot with which we have close cooperation in this work include; ABB AB, Rinas ApS, and Visual Components Oy.

The respective developer teams behind the software tools we are using also deserve to be acknowledged: Protégé, Jess, and JastAdd.

REFERENCES

Ammicht, E., Fosler-Lussier, E., and Potamianos, A. (2007). Information seeking spoken dialogue systems part I: Semantics and pragmatics. *IEEE Transactions on Multimedia*, 9(3):532–549.

Antoniou, G. and van Harlmelen, F. (2004). *A semantic web primer*. The MIT Press.

Bersot, O., Guedj, P.-O. E., Godéreaux, C., and Nugues, P. (1998). A conversational agent to help navigation and collaboration in virtual worlds. *Virtual Reality*, 3(1):71–82.

Buitelaar, P. (2007). On the role of natural language processing in a data-driven approach to the ontology life-cycle. Keynote talk at TALN, Toulouse, France (http://olp.dfki.de/ontoselect/).

Friedman-Hill, E. (2007). Jess, the rule engine for the Java platform. http://herzberg.ca.sandia.gov/, site accessed December 2007.

Godéreaux, C., Guedj, P.-O. E., Revolta, F., and Nugues, P. (1998). *Virtual Worlds on the Internet*, chapter Ulysse: An interactive, spoken dialogue interface to navigate in virtual worlds. Lexical, syntactic, and semantic issues, pages 53–70, 308– 312. IEEE Computer Society Press, Los Alamitos, California.

Haegele, M. (2007). White paper on trends and challenges in industrial automation.

JastAdd (2007). The JastAdd Compiler-Compiler System. http://jastadd.cs.lth.se, site accessed December 2007.

Malec, J., Nilsson, A., Nilsson, K., and Nowaczyk, S. (2007). Knowledge-Based Reconfiguration of Automation Systems. In *Proceedings of IEEE CASE*, pages 170–175. IEEE.

McGuinness, D. L. and van Harmelen, F. (2007). OWL web ontology language. http://www.w3.org/TR/owl-features/, site accessed December 2007.

Potamianos, A., Fosler-Lussier, E., Ammicht, E., and Perakakis, M. (2007). Information seeking spoken dialogue systems part II: Multimodal dialogue. *IEEE Transactions on Multimedia*, 9(3):550–566.

Protégé (2007). The Protégé ontology editor and knowledge acquisition system. http://protege.stanford.edu, site accessed December 2007.

RDF (2007). Resource description framework. http://www.w3.org/RDF/, site accessed December 2007.

RDFS (2007). RDF schemas. http://www.w3.org/TR/rdf-schema/, site accessed December 2007.

SIARAS (2007). Skill-based inspection and assembly for reconfigurable automation systems.

SMErobot (2007a). The European robot initiative for strengthening the competitiveness of SMEs in manufacturing.

SMErobot (2007b). Preliminary description of concepts for high-level programming methods. Technical Report Deliverable DR2.6. Deliv.

SPARQL (2007). SPARQL protocol and RDF query language. http://www.w3.org/TR/rdf-sparql-query/, site accessed December 2007.

SWRL (2007). SWRL: A semantic web rule language combining owl and ruleml. http://www.w3.org/Submission/SWRL/, site accessed December 2007.

VoiceXML (2007). Voice extensible markup language (VoiceXML) 2.1. http://www.w3.org/TR/voicexml21/, site accessed December 2007.

Wielemaker, J., Hildebrand, M., and van Ossenbruggen, J. (2007). Using Prolog as the fundament for applications on the semantic web. In *Proceedings of the ICLP'07 Workshop on Applications of Logic Programming to the Web (ALPSWS-2007)*, Porto, Portugal.

XSLT (2007). XSL transformations (XSLT) version 2.0. http://www.w3.org/TR/xslt20/, site accessed December 2007.

PLUG-AND-PRODUCE TECHNOLOGIES
On the Use of Statecharts for the Orchestration of Service Oriented Industrial Robotic Cells

Germano Veiga and J. Norberto Pires
Mechanichal Engineering Department, University of Coimbra
Rua Luís Reis dos Santos, Coimbra, Portugal
gveiga@robotics.dem.uc.pt, norberto@robotics.dem.uc.pt

Keywords: Service Oriented Architectures, SCXML, Industrial robotic cells.

Abstract: Programming industrial robotic workcells is a challenging task, namely because it means dealing with several types of machines, mange the data flow between them and orchestrate their basic functionality into a working program. In this work service oriented architectures are used for the task of programming robotic workcells along with managing the communication between cell components, and a statechart model engine is implemented to orchestrate the system logic. The objective of this paper is to focus in merging service oriented architectures with StateCharts XML, and in evaluating that robotic workcell programming approach using a simple laboratory test bed.

1 INTRODUCTION

The integration of different components in an industrial robotic cell is a time consuming task. Nowadays, industrial automation is using technologies originally designed for wider range of non-traditional industrial companies. The evolution of vision systems, 3D scanners, intelligent sensors PLC's, with their special languages and programming environments, etc., originated an enormous collection of interesting and powerful devices which are easier to program, although harder to integrate in their full extent. Consequently, porting plug-n-play concepts from PC's to the industrial automation environment is a promising opportunity. Even though similar to plug-n-play, in many aspects, the plug-n-produce concept (Veiga et al. 2007) has to deal with some specifics from the industrial automation world. One of these specifics is related with the presence of many highly programmable devices. Considering an industrial robotic cell, connecting a sensor to a robot controller can be compared to connecting a mouse to a PC, but integrating a programmable vision system or a PLC with an industrial robot requires some orchestration logic. To materialize this plug-n-produce concept regarding highly programmable devices, service oriented architectures (SOA) have been pointed as a promising approach (SMErobot™ 2007-2009). service oriented architectures are composed by autonomous services and are extensively event driven.

Finite-state automatons are very commonly used in the task of modeling the behavior of industrial automation systems. Due to their discrete event nature, industrial systems are well described by states and event driven transitions.

Harel StateCharts are a widely used extension to the finite-state automata model, and SCXML (Barnett et al. 2007) is a modern standardized way of specifying them.

The purpose of the current paper is to evaluate a SCXML-derived language to program service-oriented industrial robotics cells. This paper also presents a software application that materializes this concept, and the results obtained using a simple laboratory robotic test bed.

2 SOA - UPNP

With the advent of internet, service oriented architectures (SOA) emerged to increase the degree of decoupling between software elements. A SOA relies on highly autonomous but interoperable systems. The definition of a service is ruled by the larger context; this means that all technological details are hidden, but also that the concept which

supports the service is more business (or process) oriented (instead of being technology oriented). SOA enables software engineers to focus more on the business logic and less on the interconnection details. At the device level, service-oriented architectures are emerging as the way to deal with the increasing amount of embedded devices present in our homes and offices.

In manufacturing, the inherent complexity (necessarily hidden from the user) associated with the presence of many types of devices and machines makes the concept of service-oriented architectures particularly attractive (SIRENA 2005, SODA 2006, SOCRADES 2006). In fact, it leads to the idea that each workcell programming block (that is, not only physical devices) should be considered as a potential device (SOA device style) that offers programming services.

Considering a *holonic* workcell structure (Gou, 1998), with *holons* composed by automation devices, like an industrial robot or a vision system, one can classify as uncommon the need to have real-time in the communication framework. The majority of the component connections can instead be described in terms of coarse-grained services, with synchronous calls for setup and asynchronous events for operation.

Considering an industrial robotic cell ecosystem, past work (Veiga et al, 2007) revealed that service-oriented architectures can provide a suitable platform to a plug-n-produce environment.

Nevertheless, there are several approaches to SOA, namely, if we consider only the four most relevant platforms: Jini (Jini 2006), UPnP (UPnP, 2007), DPWS (Chan, 2006) and DSSP (Nielsen, 2007).

Jini is an architecture proposed by Sun Microsystems based on Java. This fact makes it platform independent but language dependent. It also carries a large memory footprint, due to the presence of a virtual machine and extensive libraries, making it less appropriate for very small devices.

UPnP and DPWS rely extensively on standard network protocols such as TCP/IP, UDP, HTTP, SOAP, XML, and the web technology. This makes them platform and language independent, which is a major advantage for their adoption. XML formats are broadly used and accepted and provide modern data interchange mechanisms and communications. Their style is close to the one defined in the enterprise world with the pair WSDL/SOAP.

Although similar in many aspects, the UPnP and DPWS architectures use different languages for device description and different protocols for discovery and event notifications. There is an enormous dynamics around DPWS. Nevertheless, the new Microsoft operating system, Microsoft Vista, supports both technologies under the name plug-and-play extensions for Windows (PnP-X, 2006)

DSSP is a simple SOAP-based protocol that defines a lightweight, REST-style service model (Nielsen, 2007) that also relies extensively on web technology. Paired with concurrency and coordination runtime (CCR) it constitutes the major parts of the Microsoft Robotics Studio (MSRS) platform.

DSSP architecture style is radically different from the WSDL/SOAP model. UPnP and DPWS are very similar technologies which mean that concepts and design styles can be easily ported between each other.

In this work UPnP was selected due to representativeness of the platform, the quantity and quality of the tools available, and our experience with the UPnP based services.

3 SCXML

StateChart XML (SCXML) can be described as an attempt to render *Harel StateCharts* in XML. The aim of this standard is to provide a basis for future standards in the area of multimodal dialogue systems. Even though this effort is being carried by the W3C group for voice technologies, SCXML provides a generic state-machine based execution environment and a modern (XML) state machine notation for control abstraction. In fact, SCXML is a candidate for control language within multiple markup languages coming out of the W3C.

Harel StateCharts are an extension to finite-state automata. These extensions are needed in order to make finite-state automata useful, and they include:

Hierarchy – StateCharts may be hierarchical, i.e., a state may contain another statechart down to an arbitrary depth.

Concurrency – Two or more statecharts may be run in parallel, which means that their parent state is in two states at the same time.

History – A state holds information that allow a "*pause and resume*" behavior.

Consider for example the microwave oven model presented in Figure 1.

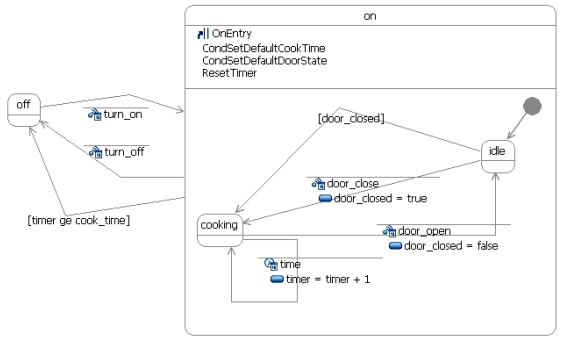


Figure 1: Microwave oven (Adapted from Barnett et al 2007).

The equivalent SCXML specification is:

```
<?xml version="1.0"?>
<scxml xmlns=
       "http://www.w3.org/2005/07/scxml"
       version="1.0"
       initialstate="off">

  <state id="off">
    <!-- off state -->
    <transition event="turn_on">
      <target next="on"/>
    </transition>
  </state>
  <state id="on">
    <initial>
        <transition>
            <target next="idle"/>
        </transition>
    </initial>
    <onentry>
         …
    </onentry>
    <transition event="turn_off">
        <target next="off"/>
    </transition>
         <transition cond="${timer ge
         cook_time}">
        <target next="off"/>
    </transition>
    <state id="idle">
         <transition
         cond="${door_closed}">
        <target next="cooking"/>
      </transition>
         <transition event="door_close">
         <assign name="door_closed"
         expr="${true}"/>
        <target next="cooking"/>
      </transition>
    </state>
    <state id="cooking">
      …
    </state>
  </state>
</scxml>
```

Has it can be seen in this example, an SCXML statechart can be divided in two major parts: the first composed by the machine states and correspondent transitions, and the other composed by the executable content.

The SCXML executable content consists of actions that are performed as part of taking transitions and entering and leaving states. The executable content is responsible for the modification of the data model, for raising events and invoking functionality on the underlying platform. It's worth noting that executable content cannot cause a state change, or fire a transition, except indirectly, by raising events that are then caught by transitions. This separation in the specification leaves room for platforms to add additional executable content corresponding to special features.

4 EXPERIMENTS

4.1 Experimental Setup

The robotic cell used in this demonstration is composed of an industrial robot (ABB IRB 140), equipped with the modern IRC5 controller, a conveyor controlled by a PLC (Siemens S7-200) and a USB web camera (Figure 2).

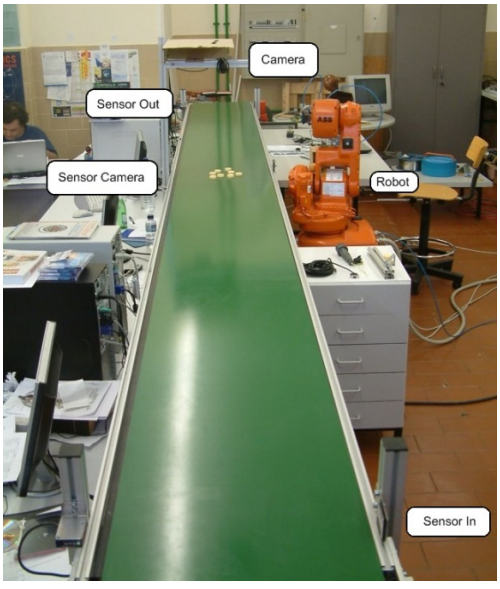


Figure 2: Equipment for experimental setup.

Basically, the conveyor transports sample pieces over the machine vision system which calculates the

number and position of the pieces. The results are sent to the robot controller to command the robot to pick them from the conveyor and place them into a box.

Two different applications were developed to operate the cell: a speech interface and a PDA interface.

4.2 Previous Work

In previous work (Veiga et al, 2007) this Industrial Robotic cell test-bed was used in order to validate SOA in a plug-n-produce environment. The cell components were represented in the network by UPnP devices that provided services as way to expose their functionality (Figure 3).

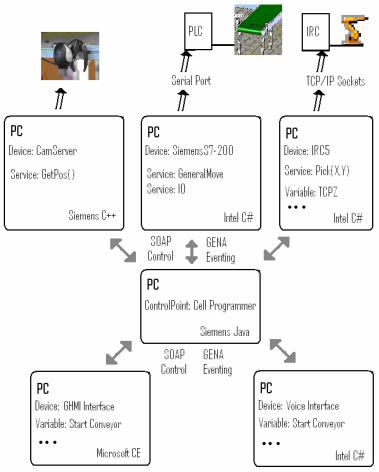


Figure 3: UPnP Network of the industrial test-bed.

Five UPnP devices representing five workcell components were developed. In some of these devices an extra layer was needed because native UPnP support could not be implemented. Detailed description can be found in (Veiga et al, 2007).

The *Cell Programmer Interface* (Figure 4) is a software application developed to control the flow of high level tasks in a manufacturing cell. Basically, it's an UPnP *control point*, with some tools suitable to build a generic stack. This stack represents the control flow of process related tasks. In the left side of this interface a tree shows all UPnP devices founded on the network.

Clicking over them it is possible to get additional information (access the presentation page, for example). Using the "arrow" button, actions or events are added to the stack. Furthermore, when running the resulting program and the program counter is pointing to an event, it means that the program is "waiting" for that event to occur. Inversely, if the program counter is pointing to an action, it means that it is calling that action and waiting for the return. There is also the possibility of defining auxiliary variables to store values that can be used as arguments in later stack steps.

4.3 Analysis

The simple stack approach revealed to be very limited to handle more complex systems. These systems often have concurrent tasks, multiple transition events and many other orchestration requirements which are impossible to model with a simple stack system.

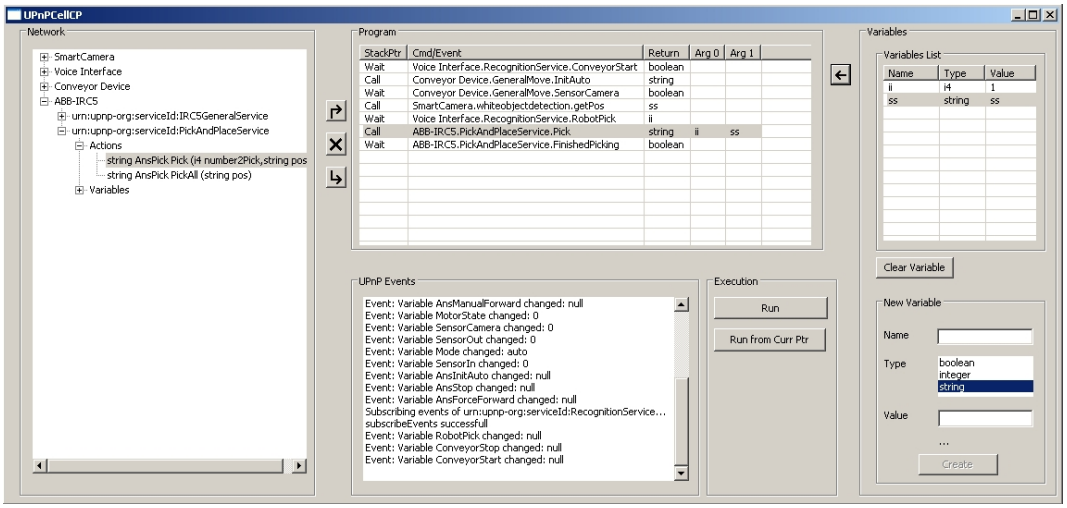


Figure 4: Simple orchestration system (Veiga et al, 2007).

To address this problem, this paper proposes a more powerful orchestration model that relies on standard technologies (SCXML) and solves many of the described problems.

Another issue with the previous approach was related with the use of the auxiliary variables and the need to check the presence of remote functionality on distributed systems. A robot cell program is only valid if all the services needed are available, but also if the used variables have the desired value. If this value is obtained by previous steps it's not possible to check if that value is still valid (corresponding to a live device, for example). To address this issue the solution proposed only relies on UPnP state variables to manage data, discarding the use of auxiliary variables

4.4 Implementation

The software developed can be divided in two distinct parts: the implementation of the statechart engine; and the user interface itself.

4.4.1 StateChart Engine

Actually there are too few SCXML implementations available, and the most notable effort is *CommonsSCXML* (CommonsSCXML 2007). Since *CommonsSCXML* is still in a 0.x version, and there was the need to extend the standard functionality, it was decided here to develop an SCXML engine from scratch.

The application presented in this paper was developed in C# following the guidelines of (Samek, M. 2002) including the basic part of the SCXML language. Considering the W3C standard (Barnett et al. 2007), our implementation doesn't include the *Extensions to the basic State Machine Model* and the *Executable Content.*

This approach was taken not only for simplicity but also with the objective of keeping the cell program as simple as possible. This objective of simple orchestration programs is sustained by the *holonic* cell structure referenced earlier, where devices expose high-level functionality services and the cell orchestration programs are reduced in terms of flow control, managing data etc. As such, all executable content within the cell orchestration program is always related with processing UPnP events or UPnP actions.

4.4.2 User interface

The *UPnPSCXMLCellProgrammer* is a graphical user interface that allows the composition of workcell orchestration programs relying on UPnP devices and the SCXML derived model (Figure 5). To behave like this the StateChart engine has the added capability to recognise whether an event or executable content are UPnP or not.

When the engine gets an UPnP event it converts it into an SCXML event. When the statechart engine finds an UPnP action inside an executable content block it just calls that action with the parameters (UPnP arguments) enclosed.

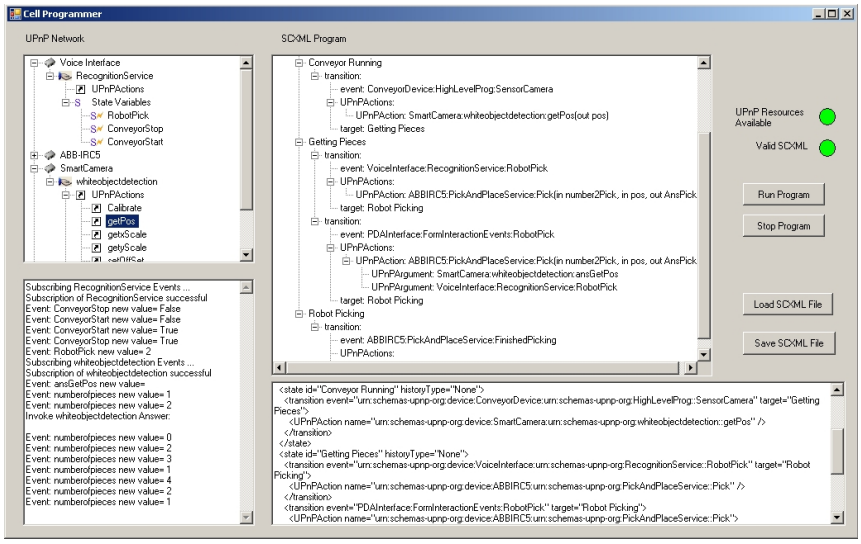


Figure 5: UPnPSCXMLCellProgrammer Interface.

In this application the user can drag and drop UPnP actions and UPnP events from the UPnP network and place them into the SCXML program. It's worth noting that the events in this program are always network events, and that only UPnP actions can be assigned to the *OnEntry* and *OnExit* handlers and to the executable content of the transitions, which are executed between the *OnExit* handler of the source state and the *OnEntry* handler of the current state.

UPnP events and UPnP actions are defined by their complete Unified Resource Name (URN) that includes the name of the action or service, plus the URN of the owning service, plus the URN of the device.

5 EXPERIMENTS AND RESULTS

The test bed used to experiment this new approach that merges UPnP with SCXML is the same used previously (Figure 2). The program logic is a little bit more complex with the objective to show some of the new possibilities (Figure 6).

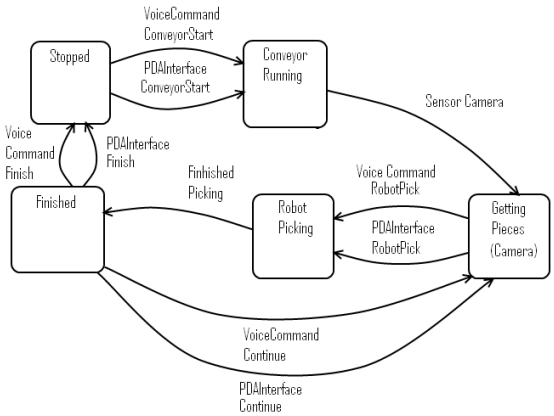


Figure 6: Program statechart.

In fact, the new program logic allows the user to give alternatively speech commands or PDA commands, and re-picking leftovers before asking the conveyor to run again.

In comparison with the previous situation much more complex orchestration schemes can be obtained. Statecharts provide a very nice way to model systems logic but are very limited when dealing with data processing. With this combined approach pairing statecharts with SOA all data processing is made within SOA, leaving statecharts for modeling systems states and logic. Considering the experiments made so far we can point statecharts as a suitable model to orchestrate *holonic* automation workcells, due to their capabilities in dealing with events and states in opposition of dealing with data processing.

6 CONCLUSIONS

This paper reports results from an ongoing research work. Experiments done so far revealed that the added features enable the cell programmer to define powerful and more complex orchestrations that can handle complex systems.

Consequently, merging service oriented architectures with statecharts proved to be an interesting approach to model the workcell orchestration logic. Future work will focus on the evaluation of this approach with more complex systems and in providing a more user friendly graphical interface.

ACKNOWLEDGEMENTS

This work has been mainly funded by the European Commission's Sixth Framework Program under grant no. 011838 as part of the Integrated Project SMErobot™.

REFERENCES

Abb, 2005 ABB IRC5 Documentation, ABB Flexible Automation, Merrit, 2005

Ahn S. C., Kim J.H., Lim K., Ko H.,Kwon Y and Kim H., 2005 UPnP Approach for Robot Middleware *P Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona, Spain, April 2005*.

Barnett, J. et al, 2007. State Chart XML (SCXML): State Machine Notation for Control Abstraction. http://www.w3.org/TR/2007/WD-scxml-20070221/

Chan, S., Conti, D., Kaler, C., Kuehnel, T., Regnier, A, Roe, B., Sather, D., Schlimmer, J., Sekine, H., Thelin, J., Walter, D., Weast, J., Whitehead, D., Wright, D., and Yarmosh, Y. (2006). "Devices Profile for Web Services." http://schemas.xmlsoap.org/ws/2006/02/devprof/

CommonsSCXML, 2007, available from: http://jakarta.apache.org/commons/scxml

James, F. and H. Smit ,2005 Service Oriented Paradigms for Industrial Automation. In: *IEEE Transactions on Industrial Informatics, Vol. 1, no. 1 February 2005*.

Gou L., P. Luh, and Y. Kyoyax (1998). Holonic Manufacturing Scheduling: Architecture, Cooperation Mechanism, and Implementation. *'97., IEEE/ASME International Conference on Advanced Intelligent Mechatronics* Vol 37, 213-231,

Harel D. 1987. StateCharts: A Visual Formalismfor Complex Systems. *Science of Computer Programming* 8, North Holland.

Nielsen, H. and G. Chrysanthakopoulos. (2007) Decentralized Software Services protocol – DSSP/1.0

PnP-X (2006): Plug and Play Extensions for Windows Specification. Available: www.microsoft.com/whdc/Rally/pnpx-spec.mspx.

Samek, M. (2002). *Pratical StateCharts in C/C++*, CMPBooks

Schlimmer J., S. Chan, C. Kaler., T. Kuehnel, R. Regnier, B. Roe, D. Sather, H. Sekine, D. Walter, J. Weast, D. Whitehead, and D. Wright (2004) Devices Profile for Web Services: A Proposal for UPnP 2.0 Device Architecture.Available: http://xml.coverpages.org/ni2004-05-04-a.html.

SIRENA Project (2005), Service Infrastructure for Real-time Networked applications, Eureka Initiative ITEA. Available: www.sirena-itea.org.

SOCRADES. (2006). "Service-Oriented Cross-layer infRAstructure for Distributed smart Embedded devices." http://www.socrades.eu/

SODA. (2006). "Service Oriented Device and Delivery Architecture." http://www.soda-itea.org/

UPnP forum (2004).Available: http://www.upnp.org

Veiga G., Pires JN, Nilsson K.. On the use of SOA platforms for industrial robotic cells: Intelligent Manufacturing Systems Proceedings IMS2007, Spain, 2007

SMErobot™ (2007-2009), The European Robot Initiative for Strengthening the Competitiveness of SMEs in Manufacturing, www.smerobot.org

SPECIAL SESSION ON MULTI-AGENT ROBOTIC SYSTEMS

CHAIRS:
PETER SAPATY
JOAQUIM FILIPE

USING ROBOTIC SYSTEMS IN A SMART HOUSE FOR PEOPLE WITH DISABILITIES

Viorel Stoian and Cristina Pana
CCMR Craiova - University of Craiova, Decebal Street No. 107, Craiova, Romania
stoian@robotics.ucv.ro, cristina@robotics.ucv.ro

Keywords: The assistive technology, smart house, robots and mobile robots, edutainment, robot control, artificial potential field method.

Abstract: We present in this paper several ideas about the usability of the robotic arms and mobile robots as an assistive technology in a smart house where people with disabilities daily live. First, psychological and social aspects of smart home technology are presented and after that the modularity and standardization processes are discussed. Next we propose a smart house plan, equipped with a mobile robot which has a manipulator arm. This robotic system is used to help vulnerable persons, the handicapped men vehicle seat being equipped with a robotic arm which can manipulate objects by a hyper-redundant gripper. For the control of the processes in the smart house, we propose a hierarchical control system and for the mobile robot we use the artificial potential field method. Also, this paper points out the edutainment concept (EDUcation and enterTAINMENT) by robotics. Finally, some applications are presented.

1 INTRODUCTION

Technology can play a major role in assisting process of the people in their daily life. Designing smart environments is a goal that appeals to researchers in a variety of disciplines, including artificial intelligence, pervasive and mobile computing, robotics, middleware and agent-based software, sensor networks, and multimedia computing (Cook, Das, 1989). Because smart environment research is being conducted in real-world, physical environments, design and effective use of physical components such as sensors, controllers, and smart devices are vital.

We define a smart environment as one that is able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment (Youngblood et al., 2005).

Systems are required to be robust and reliable as the person with disabilities will rely on the installed devices and they will become internalized within their self-concept (Dewsbury, Edge, 2000, 2001), (Lupton, 2000).

Some of the properties of the environment need to be captured and they can be measured thus: motion properties (position, velocity, angular velocity, acceleration), presence (tactile/contact, proximity, distance/range, motion), biochemical (biochemical agents), physical properties (pressure, temperature, humidity, flow), contact properties (strain, force, torque, slip, vibration), identification (personal features, personal ID) (Lewis, 2004).

The information required by smart environments is measured by sensors and collected using sensor networks. These sensor networks are responsible for acquiring and distributing data needed by smart buildings, utilities, industries, homes, ships, and transportation systems. Sensor networks need to be fast, easy to install and maintain, and self-organizing.

There are many potential uses for a smart environment. With the maturing of smart environment technologies, at-home automated assistance can allow people with mental and physical challenges to lead independent lives in their own homes. Pollack (Pollack, 2005) categorizes such assistive technology as meeting the goals of assurance (making sure the individual is safe and performing routine activities), support (helping individual compensate for impairment), and assessment (determining physical or cognitive status) (Cook and Das, 1989).

Pineau, et al. (Pineau, 2003) demonstrate the benefits of robotic assistants in nursing homes, while Helal, et al. (Helal, 2005) provide a visitor-

identifying front door, inhabitant-tracking floor and a smart mailbox to volunteer seniors living in the Gator Tech Smart Home. Kautz, et al. (Kautz, 2002) show that assistance is not limited to a single environment. Using their activity compass, the location of an individual can be tracked, and a person who may have wandered off can be assisted back to their goal (or a safe) location.

Finally, smart environments can be used to actually determine the cognitive impairment of the inhabitants. Carter and Rosen (Carter and Rosen, 1999) demonstrate such an assessment based on the ability of individuals to efficiently complete kitchen tasks.

While performance measures can be defined for each technology within the hierarchical architecture, performance measures for entire smart environments still need to be established.

(Mann and Bendixen, 2007) makes a distribution of the assistive technology in a smart house on eight levels, from the lowest level (basic communications) to eighth level (household arrangements).

Most people see robotics as being a vital technology for providing society with the assistive solutions that it needs in present and will need in the future. The purpose of Assistive Technology (AT) is to provide assistance, without to be a substitution for personal care, to enable people to lead a better quality of life. This technology was applied to devices for personal use created specifically to enhance the physical, sensory and cognitive abilities of people with disabilities and to help them function more independently in environments oblivious to their needs (Story, Mueller and Mace, 1998). People with disabilities are the principal beneficiaries of the technological growth.

2 PSYCHOLOGICAL AND SOCIAL ASPECTS OF SMART HOME TECHNOLOGY

The use of technology appears to present dramatic compromises in social activities, role definition, and identity (Gitlin, 1995).

Approximately all older persons and people with disabilities might feel that they are not included in discussions on technology, as it is perceived as irrelevant to their needs.

Isolation is a major problem for any person who is older or has a debilitating disability (Marshall 2000).

People who are incapacitated in some way are at the mercy of others to provide the simple basic needs. People who do not have disabilities should not to be concerned with food, shelter or human contact as they are part of every day life. It is there essential that people with disabilities are not given substandard care packages that do not meet their needs in all areas: social, psychological, physical, social and emotional. Similarly, care packages should not be over technologies so that the person is reduced to being the slave of technology (Dewsbury, Edge and Taylor, 2001; Dewsbury, 2001).

3 EDUTAINMENT BY ROBOTICS FOR PEOPLE WITH DISABILITIES

Edutainment is a neologism with is derived from the expression “EDUcation by enterTAINMENT” (Muscato and Longo, 2003). It means “Learning and playing”. In the edutainment systems or products are included different elements that have been designed to teach or to train persons and at the same time to entertain those persons. For young people with disabilities is very important to learn reading and writing. In the future is very important toad to these processes initiation and learning new assistive technology and devices (computer science, internet, telecommunications, robotics, flexible automation etc.) with will be present inside each smart home. Edutainment has a great success, especially, to young people. A person with disabilities can get through 5 levels of the edutainment which cover a large period of time, from pre-school level to researchers’ and practitioners’ level: pre-school, kindergarten, school, university, and applications/ research (Stoian, Bizdoaca and Pana, 2006). On the last level the researchers design systems and applications for the others levels.

4 MODULARITY AND STANDARDIZATION

(Virk, 2003) focuses on the state of play of component modularity and standardization in a number of application sectors that have good potential for adopting the robotics technology in the near future. In a smart house for people with disabilities there are many and different technological systems. Because the design of such

mechatronic systems is very complex, is necessary to split this design problem into specific areas of mechanics, sensor systems, actuators and powering systems, communication interfaces and hardware and software components of the computing process. In this mode is easier to develop a generic methodology.

The modular design methodology supposes to enable the individual modules to be designed as black boxes that interact with one another via an interaction space (data buses, intelligent actuators, intelligent sensors, intelligent power supply, mechanics, and controllers). The design process should include aspects of standardization so that wider issues of open components can be determined. This can be done by looking for specific application areas and establishing the status in each from the viewpoint of where the technologies are and what the status as regards standards is and what are the future requirements.

5 A PROPOSAL FOR A SMART HOUSE DESIGN

Here we propose a map of a smart house where live vulnerable people (Figure 1). HMVS means *Handicapped Men Vehicle Seat* and MR means *Mobile Robot*. These devices with locomotion facilities are controlled by smart systems (controllers or computers) and implement some methods or algorithms lake in Section 7.

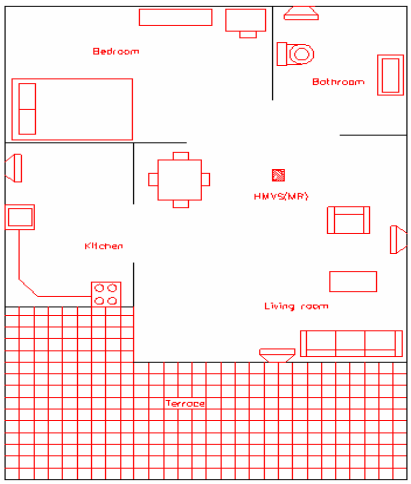


Figure 1: A smart house map.

The external areas can be compound from garden, terrace, drive way, entrance, and stairs. The internal areas can be termed circulation and external for the others and can be compound from: kitchen, living room, bedroom, bathroom, and general. This area has minimal physical barriers between the rooms. Technological systems could be allocated to these functional areas. Some systems (for example, motorized windows or doors), may be the same (physically and functionally) in more than one functional area. This is especially the case for people with long-term degenerative conditions whose quality of life can be enhanced by judicious introduction of this technology (Edge, Taylor, Dewsbury and Groves, 2000).

Systems map to one or more rooms (functional areas). A system that is not mapped to any functional area is not required. Also many of these systems will interact with each other. Some systems may be sufficiently interconnected that they would be better treated as two parts of one bigger system.

It is concluded than that there are two basic types of mapping: either a system will map to one or more rooms (functional areas) or a functional area will map to one or more systems.

6 THE MOBILE ASSISTANT ROBOTS

In this section we propose two solutions which presuppose the use of the robotic systems. First, we propose the installation of a robotic arm on the handicapped men vehicle seat (Figure 2).

Figure 2: Handicapped men vehicle seat with robotic arm.

This arm can execute different actions and different functions which the vulnerable persons are

deprived of. It is endowed with a hyper-redundant gripper. The gripper can manipulate different objects with different forms (Figure 3).

Figure 3: The hyper-redundant gripper of the arm.

Second, we propose a mobile robot with anthropomorphic arm which is endowed with an anthropomorphic manipulator (Figure 4).

Figure 4: Mobile robot with anthropomorphic arm.

This device can run inside of internal and external areas and can satisfy many needs of the resident. For example, it can grip and bring a cup of tee, milk or coffee.

7 MOBILE ROBOT CONTROL BY ARTIFICIAL POTENTIAL FIELD METHOD

7.1 The Artificial Potential Field Approach

In order to avoid the difficulties associated with the dynamical model, the control law is based only on the gravitational potential and a new artificial potential. It is shown that to drive the mobile robot to a desired point in an unconstrained movement is necessary the artificial potential to be a potential functional whose point of minimum is attractor for the system. Also, this method is used for a constrained movement in the environment with obstacles. The target position is represented by an artificial attractive potential field and obstacles by corresponding repulsive fields, so that the trajectory to the target can be associated with the unique flow-line of the gradient field through the initial position and can be generated via a flow-line tracking process. This approach is suitable for real-time motion planning of robots since the algorithm is simple and computationally much less expensive than other methods based on global information about the task space. It is difficult in the artificial potential field framework to regulate the transient behaviour of the generated trajectories such as the movement time to the target and the shape of the velocity profile. For example, even if the potential function without local minima is used, it is difficult to estimate the movement time required for reaching beforehand.

Potential field was originally developed as on-line collision avoidance approach, applicable when the robot does not have a prior model of the obstacles, but senses them during motion execution (Khatib, 1986). Using a prior model of the workspace, it can be turned into a systematic motion planning approach. Potential field methods are often referred to as “local methods”. This comes from the fact that most potential functions are defined in such a way that their values at any configuration do not depend on the distribution and shapes of the obstacles beyond some limited neighbourhood around the configuration. The potential functions are based upon the following general idea: the robot should be attracted toward its goal configuration, while being repulsed by the obstacles.

In order to make the robot be attracted toward its goal configuration, while being repulsed from the obstacles, $\prod$ is constructed as the sum of two elementary potential functions:

$$\prod(\mathbf{x}) = \prod_A(\mathbf{x}) + \prod_R(\mathbf{x}) \qquad (1)$$

where: $\prod_A(\mathbf{x})$ is the *attractor potential* and it is associated with the goal coordinates and it isn’t dependent of the obstacle regions.

$\prod_R(\mathbf{x})$ is the *repulsive potential* and it is associated with the obstacle regions and it isn’t dependent of the goal coordinates.

In this case, the force $\mathbf{F}(t)$ is a sum of two components: the *attractive force* and the *repulsive force*:

$$\mathbf{F}(t) = \mathbf{F}_A(t) + \mathbf{F}_R(t) \qquad (2)$$

7.2 Attractor Artificial Potential Field

The artificial potential is a potential function whose points of minimum are attractors for a controlled system. It was shown (Takegaki and Arimoto, 1981; Douskaia, 1998; Masoud, and Masoud, 2000; Tsugi, Tanaka, Morasso, Sanguineti and Kaneko, 2002, Mohri, Yang, and Yamamoto, 1995) that the control of robot motion to a desired point is possible if the function has a minimum in the desired point. The attractor potential $\prod_A$ can be defined as a functional of position coordinates $\mathbf{x}$ in this mode:

$$\prod_A(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n}\left[k_i\left(x_i - x_{Ti}\right)^2 + k_{n+i}\dot{x}_i^2\right] \qquad (3)$$

The function $\prod_A(\mathbf{x})$ is positive or null and attains its minimum at $\mathbf{x}_T$, where $\prod_A(\mathbf{x}_T) = 0$. $\prod_A(\mathbf{x})$ defined in this mode has good stabilizing characteristics (Khatib, 1986), since it generates a force $\mathbf{F}_A$ that converges linearly toward 0 when the robot coordinates get closer the goal coordinates:

$$\mathbf{F}_A(\mathbf{x}) = k(\mathbf{x} - \mathbf{x}_T) \qquad (4)$$

Asymptotic stabilization of the robot can be achieved by adding dissipative forces proportional to the velocity $\dot{\mathbf{x}}$.

7.3 Repulsive Artificial Potential Field

The main idea underlying the definition of the repulsive potential is to create a potential barrier around the obstacle region that cannot be traversed by the robot trajectory. In addition, it is usually desirable that the repulsive potential not affect the motion of the robot when it is sufficiently far away from obstacles. One way to achieve these constraints is to define the repulsive potential function as follows (Latombe, 1991):

$$\Pi_R(\mathbf{x}) = \begin{cases} \frac{1}{2}k\left(\frac{1}{d(\mathbf{x})} - \frac{1}{d_0}\right)^2 & \text{if } d(\mathbf{x}) \le d_0 \\ 0 & \text{if } d(\mathbf{x}) > d_0 \end{cases} \qquad (5)$$

where k is a positive coefficient, $d(\mathbf{x})$ denotes the distance from $\mathbf{x}$ to obstacle and d_0 is a positive constant called *distance of influence* of the obstacle. In this case $\mathbf{F}_R(\mathbf{x})$ becomes:

$$\mathbf{F}_R(\mathbf{x}) = \begin{cases} k\left(\frac{1}{d(\mathbf{x})} - \frac{1}{d_0}\right)\frac{1}{d^2(\mathbf{x})}\frac{\partial d(\mathbf{x})}{\partial \mathbf{x}} & \text{if } d(\mathbf{x}) \le d_0 \\ 0 & \text{if } d(\mathbf{x}) > d_0 \end{cases} \qquad (6)$$

For those cases when the obstacle region isn't a convex surface we can decompose this region in a number (N) of convex surfaces (possibly overlapping) with one repulsive potential associated with each component obtaining N repulsive potentials and N repulsive forces. The repulsive force is the sum of the repulsive forces created by each potential associated with a sub-region.

We propose the mobile robot to move from initial point (x, y) = (0, 0) to final point (x_T, y_T) = (7, 5). If any obstacles are not between the two point, the trajectory is a straight line. If we consider that there is a dot obstacle, in the point (x_R, y_R) = (4, 3), with *distance of influence* d_0 = 0.4, the trajectory is like in Figure 5.

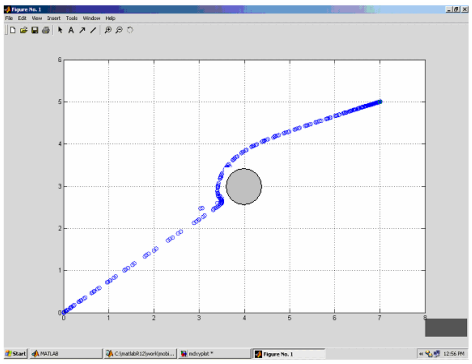


Figure 5: The constrained robot trajectory by one obstacle.

8 CONCLUSIONS

Most people see robotics as being a vital technology for providing society with the assistive solutions that it needs in present and will need in the future. The purpose of Assistive Technology (AT) is to provide assistance, without to be a substitution for personal care, to enable people to lead a better quality of life. This technology was applied to devices for personal use created specifically to enhance the physical, sensory and cognitive abilities of people with disabilities and to help them function more independently in environments oblivious to their needs. People with disabilities are the principal beneficiaries of the technological growth.

REFERENCES

Cook, D., Das, S.K., 2007. How smart are our environments? An updated look at the state of the art. *Journal of Pervasive and Mobile Computing* **3**:2, pp. 53-73.

Youngblood, G. M., Cook, D. J., Holder L. B., Heierman, E. O., 2005. Automation intelligence for the smart environment. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Dewsbury, G., Edge, H.M., 2000. Designing the Home to Meet the Needs of Tomorrow … Today: Deconstruting and rebuilding the home for life, *ENHR 2000 Gavle*, http://www.rgu.ac.uk/subj/search.

Dewsbury, G., Edge, H.M., 2001. Designing the Home to Meet the Needs of Tomorrow … Today: Smart technology, health and well-being, *Open House International*.

Dewsbury, 2001. THE SOCIAL AND PSYCHOLO-GICAL ASPECTS OF SMART HOME TECHNOLOGY WITHIN THE CARE SECTOR, Published in *New Technology In The Human Services*, Vol 14, No 1-2, 2001, pp 9-18, ISSN 0959 0684.

Lupton, D., Seymur, W., 2000. Technology, selfhood and physical disabilit", *Social Science & Medicine*. 50.1852.

Lewis, F. L., 2004. Wireless sensor networks. In D. J. Cook and S. K. Das, editors, *Smart Environments: Technology, Protocols, and Applications*. Wiley, 2004.

Pollack, M. E., 2005. Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment. *AI Magazine*, 26(2):9–24.

Pineau, J., Montemerlo, M., Pollack, M. Roy, N., and Thrun, S., 2003. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*, 42(3-4).

Helal, A., Mann, W., El-Zabadani, H., King, J., Kaddoura Y. and Jansen, E., 2005. The gator tech smart house: A programmable pervasive space. *IEEE Computer*, 38(3):50–60.

Kautz, H., Arnstein, L., Borriello, G., Etzioni, O. and Fox. D. , 2002. An overview of the assisted cognition project. In Proceedings of the *AAAI Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, pages 60–65.

Carter J., and Rosen, M., 1999. Unobtrusive sensing of activities of daily living: A preliminary report. In *Proceedings of the 1st Joint BMES/EMBS Conference,* page 678.

Mann, W.C. and Bendixen, R.M., 2007. Smart Homes of the Future, http://www.agingwithdisability.org/PDF

Story M., Mueller J & Mace R, 1998, The Universal Design File: Designing for people ofall ages and abilities, *The Center for Universal Design*, NC State University.

Gitlin L., 1995. Generations: *Journal of the American Society of Ageing*: Why Older People Accept or Reject Assistive Technology, 19, 1.

Marshal, M., 2000. Astrid: A Social & Technological Response to Meeting the Needs of Individuals with Dementia & their Carers, *Hawker Publication*, London.

Dewsbury, G., Edge, H.M, Taylor, B., Designing Safe Smart Home Systems For Vulnerable People, Workshop on Dependability in Healthcare Informatics, 22nd-23rd March 2001, pp. 65-70.

Muscato, G. and Longo, D., 2003. CLAWAR WP3 applications – edutainment and service robots, *Proceedings of the 6th International Conference on Climbing and Walking Robots,* Vol. 1, Professional Engineering Publishing Limited, London, UK, 17-19 sept. 2003, pp. 1033-1042.

Stoian, V., Bizdoaca, N.G., Pana, C., "*Edutainment and Robotics - a Great Relationship*", Proceedings of the 17-th International Annual Conference on Innovation in Education for Electrical and Information Engineering (EAEEIE 2006), Craiova, Romania, June 1st-3rd, 2006, vol. I, pp. 93-98, ISBN: (10) 973-742-350-X, (13) 978-973-742-350-4.

Virk, G.S., 2002. Clawar Network and its Role for Advanced Robotics Dissemination", *Proceedings of the CLAWAR Workshop*: The role of CLAVAR in education, training, working conditions and safety, Madrid, Spain.

Edge, H.M., Taylor B., Dewsbury, G., Groves, M., 2000. The Potential for 'Smart Home Systems in Meeting the Care Needs of Older Persons and People with Disabilities, *Seniors Housing Update*, vol. 10. no. 1, 6-8.

Khatib., O, 1986. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Robot. Res*., vol. **5**, no. 1, pp. 90-98.

Takegaki, M. and S. Arimoto, 1981. A new feedback methods for dynamic control of manipulators. *Journal of Dynamic Systems, Measurement and Control*, pp. 119-125.

Douskaia, N.V., 1998. Artificial potential method for control of constrained robot motion. *IEEE Trans. On Systems, Man and Cybernetics*, part B, **28**, pp. 447-453.

Masoud, S.A., A.A. Masoud, 2000. Constrained motion control using vector potential fields, *IEEE Trans. On Systems, Man and Cybernetics*, part A, **30**, pp. 251-272.

Tsugi, T., Y. Tanaka, P.G. Morasso, V. Sanguineti and M. Kaneko, 2002. Bio-mimetic trajectory generation of robots via artificial potential field with time base generator. *IEEE Trans. On Systems, Man and Cybernetics*, part C, **32**, pp. 426-439.

Mohri, A, Yang, X. D. and Yamamoto, 1995. A Collision free trajectory planning for manipulator using potential function", *Proceedings 1995 IEEE International Conference on Robotics and Automation (ICRA95)*, pp. 3069-3074.

Latombe J.C., 1991. *Robot Motion Planning*, Kluwer Academic Publishers, Boston.

ROBOTIC SOCCER: THE GATEWAY FOR POWERFUL ROBOTIC APPLICATIONS

Luiz A. Celiberto Junior and Jackson P. Matsuura
Instituto Tecnológico de Aeronáutica (ITA)
Praça Marechal Eduardo Gomes, 50, Vila das Acácias, 12228-900, São José dos Campos, SP, Brazil
celibertojr@itandroids.com, jackson@itandroids.com

Keywords: Robotics, Robotic Soccer, Artificial Intelligence, Image Processing, Intelligent Agents.

Abstract: From the RoboCup goal of having a fully autonomous humanoid soccer team, it is possible to see many applications of the research in the Humanoid Soccer, such as the development of mechanical legs and arms and exoskeletons. The onboard vision algorithms for multi target tracking and the cooperative decision making of some Soccer Leagues can be used in squadrons of autonomous vehicles in a variety of missions. The algorithms for image processing of the Small Size League can be used in aerial or satellite images to track vehicles. The Simulation League allows the development of many intelligent agents applications. The formations and team play positioning of the Simulation League can be used to optimize the positioning of a squadron of autonomous vehicles. The research of Robotic Soccer fosters and strengthens the research in Robotics, allowing and contributing to the development of many powerful applications which can great benefit the mankind.

1 INTRODUCTION

The RoboCup initiative is an attempt to foster AI and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined. RoboCup chose to use soccer game as a central topic of research, aiming at innovations to be applied for socially significant problems and industries. The ultimate goal of the RoboCup project is by 2050, develop a team of fully autonomous humanoid robots that can win against the human world champion team in soccer (The RoboCup Federation, 2007).

Although clearly stated by RoboCup that in order for a robot team to actually perform a soccer game, various technologies must be incorporated including: design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics, and sensor-fusion (RoboCup, 2007), some people do not understand why RoboCup choose the soccer and not another robotic application with real and direct benefits to the mankind as its central topic of research. Some robotics researches even do not recognize the Robotic Soccer research as a serious one. And even among RoboCup researches there is some which are so concentrated in developing competitive Soccer Teams that do not really realize the real potential of their research in Robotic Soccer.

With the introduction of the RoboCup Rescue and RoboCup Leagues, part of the research for RoboCup Competitions can be direct and immediately applied to some robotic applications, but the research in Robotic Soccer may also foster the research in more advanced and specific topics which lead to great advances in Robotics, both in hardware and software.

The objective of this work is to show some of the many relevant and important applications which can be derived directly or indirectly from the research in Robot Soccer.

Starting from the main RoboCup goal of having a fully autonomous humanoid soccer team, it is possible to see many direct applications of the research in the Humanoid Soccer League, such as the development of mechanical robotic legs and arms for the cripple and exoskeletons for paralytic. The onboard computer vision algorithms for multi target tracking and the cooperative decision making of Humanoid, Middle Size and Four Legged Soccer Leagues can be used by squadrons of sea, ground or aerial un-

manned vehicles in search-and-rescue, surveillance, recognition or even attack missions. The algorithms applied in the image processing of the Small Size League for image segmentation and multiple targeting of fast moving objects can be used in aerial or satellite images to track vehicles or boats and even aircrafts and isolate each target and its velocity and attitude. Finally even the Simulation League research is very important and besides helping the development of algorithms for the other Leagues, can allow many intelligent agents applications to be developed in many areas. In conjunction with the multi target algorithms and cooperative decision making algorithms developed in other RoboCup Leagues the defensive and offensive formations and team play positioning of the Simulation League can be used to optimize the group positioning and area coverage of a squadron of unmanned autonomous vehicles.

Some research topics of the RoboCup Soccer Leagues are presented in the next section and in section 3 the relations among the research in the RoboCup Soccer Leagues and powerful real world applications are explained in detail, leading to the conclusion that the research of Robotic Soccer in the various RoboCup Soccer Leagues foster and strengthen the research in Robotics, allowing and contributing to the development of many powerful hardware and software which can great benefit the mankind.

2 ROBOCUP SOCCER LEAGUES AND SUB LEAGUES

The RoboCup Soccer Competition has a total of five senior Leagues, some with Sub Leagues, but for our purpose we should consider only four different approaches for Robotic Soccer Research. The first of these four approaches to be considered is the Humanoid Soccer League, were anthropomorphic autonomous robots must be developed to play soccer, perform penalty kicks and accomplish some technical challenges related with soccer playing skills. The second embodies the Four Legged and the Middle Size Soccer Leagues were autonomous robots must coordinate their actions to play a soccer game. The third approach is the Small Size Soccer League, where a unique program controls an entire team of robots using the information provided by a camera that has a satellite like view of the entire field of play, including all robots, field marks and the ball. The fourth and last approach is the Simulation League where team play algorithms must be developed to autonomous intelligent agents play soccer coordinating their efforts.

2.1 Humanoid Soccer

In the RoboCup Humanoid Soccer League, autonomous robots with a human-like body plan and human-like senses play soccer against each other. In addition to soccer games, penalty kick competitions and technical challenges take place. Dynamic walking, running, and kicking the ball while maintaining balance, visual perception of the ball, other players, and the field, self-localization and team play are among the many research issues investigated in the Humanoid League (RoboCup Humanoid League, 2007). Figure 1 shows some of the Humanoid Robots which participated in the RoboCup 2006 at Bremen.

Figure 1: Robots of the Humanoid Soccer League at the RoboCup Championship - Bremen2006.

To ensure that the humanoid robots perform well all these activities a wide range of technology must be researched and adapted. Some of the research topics in the RoboCup Humanoid League are:

- The design and assembly of anthropomorphic robots;
- The development of optimal and robust control algorithms that optimize the speed of the movements keeping a robust stability of the robots;
- The development of real time image processing algorithms, capable of tracking moving objects and even anticipates actions of adversary robots;
- The development of team play algorithms that allow the coordination of the robots actions according with the state of the game.

Some RoboCup teams fully design and assembly their robots (Santos et al, 2006), (Behnke, S. et

al, 2006), while other augment some commercial robots and there is even Robotic Companies which adapt their robots to test their capabilities in the Humanoid League (Faconti, 2006).

2.2 Autonomous Multi-Agent Soccer

For our purpose of establish the relations of the research in the RoboCup Soccer Leagues and real world applications the Four-Legged and the Middle Size RoboCup Leagues can be explored together. In the Four-Legged League teams consisting of four Sony Aibo robots each play on a field of 6 m x 4 m. The robots operate fully autonomously, i.e. there is no external control, neither by humans nor by computers (Four-Leegged, 2006). In the Middle Size League two teams of mid-sized robots with all sensors on-board play soccer on a field; relevant objects are distinguished by colors; communication among robots (if any) is supported on wireless communications and no external intervention by humans is allowed, except to insert or remove robots in/from the field (RoboCup2004 Middle Size League, 2007).

The great difference among these two leagues is the hardware. In the Four-Legged League, all teams are limited to the Sony Aibo robot, while in the Middle Size the teams have the freedom to design and build their robots according to some dimensions and weight limitations. But both leagues have some research challenges in common, which are also Humanoid Soccer challenges:

- The development of real time image processing algorithms, capable of tracking moving objects and even anticipates actions of adversary robots;
- The development of team play algorithms that allow the coordination of the robots actions according with the state of the game.

But as the concern about stable biped walking doesn't exist and the design of the robots is easier or also nonexistent in these leagues and usually the processor power of the Aibos and mainly of the Middle Size Robots are far better than the humanoid ones, more advanced and complex techniques can be developed and applied in these two leagues. Such techniques can latter be used by the humanoid robots when their processor power reach better standards.

2.3 Small Size League

The Small Size Soccer League focuses on the problem of intelligent multi-agent cooperation and control in a highly dynamic environment with a hybrid centralized/distributed system. A Small Size robot soccer game takes place between two teams of five robots each.

Although local on-board vision sensors are permitted, most teams use a global vision system, where an overhead camera and an off-field PC are used to identify and track the robots. The off-field PC also performs most of the processing required for coordination and control of the robots (Small Size Robot League, 2007). Figure 2 shows the structure of the control loop for the robots using the global vision system and Figure 3 shows a typical image acquired by the overhead camera.

Fast moving multi-target tracking and multi-robot coordination are some of the big research challenges in the Small Size Soccer League.

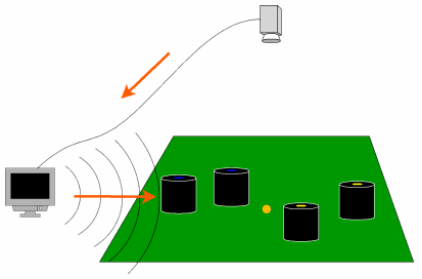


Figure 2: Control Loop Structure for the Global Vision System.

Figure 3: Typical Small Size image (Manzuri-Shalmani et al, 2006).

2.4 Soccer Simulation League

The RoboCup Soccer Simulation League have two main objectives, the first is to develop a simulation environment where it is possible to research the software aspects of RoboCup, allowing a fast development of new techniques and algorithms. The second is to present by itself a challenge multi-agent problem, for that it enables for two teams of 11

simulated autonomous robotic players to play soccer (The RoboCup Soccer Simulation, 2006) .

So, the main challenge in the Soccer Simulation League is to develop efficient team play algorithms for autonomous intelligent agents. The offensive and defensive formations of the players are one of the researches topics associated with this challenge.

3 REAL WORLD APPLICATIONS

RoboCup not only create student and media interest in the research of Robotics (Brãun, 1999) and RoboCup Soccer Competitions are not limited to just make students to work in practical solutions. The research in the Robotic soccer teams should allow a great advance in the Robotics field and besides the most obvious applications of the RoboCup research being of military or space exploration use (Kitano et al, 1998) the methods, techniques and algorithms developed to make robots play soccer can be used in many real world applications which can benefit all the mankind.

To start understand the extension and potential of the research in Robotic Soccer it is easier to imagine that the main RoboCup objective of develop a team of fully autonomous humanoid robots that can win against the human world champion team in soccer was achieved. The year is 2050 and the RoboCup humanoid team is able to win against the human soccer world champion team. What more these robots should do? What more should be done with the technology used in these robots?

From the electromechanical point of view, having such humanoid soccer players will help many cripple and paralytic people. Using the same technology and components used to assembly the soccer robots it will be possible to assembly mechanical legs and arms so or more efficient than the humans members. Also, the same algorithms used in the robots to walk, run, jump, kick and pick a ball should be used to control these robotic prosthesis. With some changes the robotics prosthesis can be turned in orthosis and even full body exoskeletons can be assembled.

The vision algorithms and the cameras should be used to help blind people and to monitor everything. The robots will be able to accurate track the ball, all the teammates and adversaries and all the field landmarks. If the image processing techniques used in the robots should do that the will also be able to guide blind people in a crowded metropolis. The vision processing will be able to tell when an obstacle is approaching, the best way to avoid a collision and even anticipate movements of other people. But we do not need to wait until 2050 to see some of the powerful applications that can be derived direct or indirect from Robotic Soccer.

3.1 Humanoid Research, Prosthesis, Orthosis and Exoskeletons

Although not all direct related with Robotic Soccer much researches are already being done and some important results are already being obtained with robotic prosthesis, orthosis and Exoskeletons.

It is true that one of the main usages of exoskeletons or lower part exoskeletons (Chu; Kazerooni; Zoss, 2005), (Low et al, 2006) is the enhancement of human soldiers, improving their endurance, speed and load carrying ability, but there are also other uses for them. Figure 4 shows the BLEEX (Berkeley Lower Extremity Exoskeleton). Exoskeletons may be used by paralytic people to perform all actions that any another people should do. Construction workers, miners, firefighters and rescue agents should also use exoskeletons to do their jobs more safely and efficiently. In the case of firefighters and rescue agents and exoskeleton may be the difference for saving an human life.

Exoskeletons or orthosis, like ankle-foot orthosis (Agrawal, 2005), (Ferris, 2005) may also be used in the rehabilitation of patients and help in physiotherapy.

Figure 4: The Berkeley Lower Extremity Exoskeleton - BLEEX.

In the research directed related with humanoid soccer robots one should cite the use of reinforcement learning for humanoid robots (Latzke; Behnke; Bennewitz, 2007) and studies of dynamic stabilization techniques for humanoid robots (Renner; Behnke, 2006). Both research topics are very im-

portant to have humanoid robots capable to adapt themselves to adverse conditions.

3.2 Image Processing, Self-Localization, Sensor Fusion

Two common problems in most Soccer Leagues are the image processing and the coordination of multi-robots. The solutions for both problems can be applied in a wide range of robotic and even just monitoring applications. Recognizing and tracking objects and using images for self-locating are general image processing research topics and are not limited to the Robotic Soccer.

Algorithms used to track a moving ball and predict changes in its movements due to contact with robots (Li; Zell, 2007) can be used to track vehicles in a road or the trajectory of any moving object, anticipating possible collisions. Algorithms developed to differentiate opponent robots from teammate robots (Lange; Riedmiller, 2007) can be used in traffic cameras to easy identity a suspect or robed car and in military operations in armored vehicles and aircrafts to avoid friendly fire.

Although some tracking and localization algorithms used in the RoboCup Soccer Leagues still relays on colored artificial landmarks (Iocchi, 2007) there is already research in the Four Legged League for self localization of the robots based on field features and not in colors (Herrero-Pérez, 2007). These algorithms can be use in any kind of unmanned vehicles to detect features in the terrain around it and self locating its position. They can also be used in conjunction with terrain data bases for better localization.

In the RoboCup Soccer Leagues there are also researches in fusion the visual information of the robots for better ball and robot localization on the field (Nisticó et al, 2007). These sensor fusion algorithms and techniques can be extended to sensor fusion and movement coordination in the target tracking of a squadron of unmanned vehicles (Ludington, 2006) the target should be a fugitive, a suspicious car, an airplane or even a spaceship or rocket which must be followed with precision.

3.3 Global Image Processing, Team Coordination

The Global Vision System of the RoboCup Small Size Soccer is very similar to the satellite imaging and surveillance aircraft image systems, where the images is collected from a point high above the ground, resulting in a practically 2D image. The image processing algorithms used in the Small Size Soccer League to track multiple fast moving objects can be used in satellite tracking and with aerial video. Figure 5 shows a unmanned aircraft vehicle with its camera field of view and a sample image.

Figure 5: UAV camera field of view (left) and sample image (right) (Arrambel et al, 2004).

There are also research in RoboCup to autonomous extract relevant information from robot marks and used this information to anticipate robots actions (Umemura, 2007). The same algorithms can be used to extract relevant information from any moving target and also anticipate its movements.

Bruce and Veloso (Bruce; Veloso, 2007) extended a motion planning algorithm primary developed and used in the Small Size Soccer robot navigation to an unmanned aircraft vehicle.

The Team coordination algorithms of the Small Size League and of the Simulation League can be used for teams of ground, sea or aerial unmanned vehicles. Search and Rescue, Patrol, Surveillance, and Escort Missions among others will need team coordination. If an off-field computer has the global view of the field of interest the Small Size decision Algorithms should be used, but if each vehicle has to take its own decision on coordinating efforts the Soccer Simulation algorithms and team positioning strategies should be used.

A good example for the use of the team positioning and team coordinating algorithms are the

search for missing people. A Squadron of UAVs must cover a wide forest area searching for missing tourists. A defensive positioning for covering a wide area without blanks like the one used in Soccer Simulation to block passes should be assumed by the UAVs squadron. Also, when one of the UAVs leaves the formation to see an area of interest the others UAVs should close the formation to cover the space left, like to close a defensive formation when one player is not available for the defensive action.

Multi-agent coordination techniques can also be applied in Air Traffic Management to ensure safer and more efficient operation of civilian aircrafts (Nguyen-Duc, 2003). And finally the Soccer Simulation league should be used to explore many intelligent agent cooperation techniques, which should be used in any of the Intelligent Agent applications fields, like Process Control, Manufacturing, Air Traffic Control, Information Management, Electronic Commerce, Business Process Management, Patient Monitoring, Health Care, Games or Interactive Theater and Cinema (Jennings; Wooldridge, 1998).

4 CONCLUSIONS

The RoboCup Soccer Competition is more than just an attractive for students and media and more than one place to test hardware and software outside the laboratories. The research done to create fully autonomous soccer robots can really be applied in many useful robotics applications and allow and foster the development of even more powerful and important applications to the mankind.

From prosthesis and orthosis to cripple, passing to image processing algorithms which can save lives and arriving in multi-agent cooperation algorithms and decision making which will optimize the actions of squadrons or even swarms of robots or intelligent agents, the researches in Robotic Soccer can really allow and foster the development of powerful robotic applications.

REFERENCES

The RoboCup Federation, 2007. http://www.robocup.org (2007). *RoboCup Humanoid League*. http:// www.humanoidsoccer.org/.

Santos, V. et al. RoboCup 2006, 2006. Humanoid League: UA Team Description. *In Proceedings of the RoboCup 2006 Symposium* CDROM, TDP .

Behnke, S. et al, 2006. NimbRo KidSize 2006 Team Description. In *Proceedings of the RoboCup 2006 Symposium* CDROM, TDP.

Faconti, D. et al, 2006. Technical description: Pal Technology. In *Proceedings of the RoboCup 2006 Symposium* CDROM, TDP.

Four-Legged League, 2007 *http://www.tzi.de/4legged/bin/view/Website/WebHome*

RoboCup2004 Middle Size League, 2007 *http://www.er.ams.eng.osaka-u.ac.jp/rc2004msl/*.

Small Size Robot League, 2007 *http://small-size.informatik.uni-bremen.de/*.

Manzuri-Shalmani, M. T. et al, 2006. Team Description Paper of the Robo-Cup Small-Size League 2005 Sharif CESR. In *Proceedings of the RoboCup 2006 Symposium* CDROM, TDP.

The RoboCup Soccer Simulator, 2007. *http://sserver.sourceforge.net/*.

Bräunl, T. Research Relevance of Mobile Robot Competitions, 1999. *IEEE Robotics and Automation Magazine*, v. 6, n. 4 32-37.

Kitano, H. et al. RoboCup: Robot World Cup, 1998. *IEEE Robotics and Automation Magazine*, v. 5, n. 3 30-36.

Chu, A., Kazerooni, H., Zoss, A, 2005. On the Biomimetic Design of the Berkeley Lower Extremity Exoskeleton (BLEEX). In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* - ICRA 2005 4345- 4352.

Low, K. H. et al, 2006. Locomotive Control of a Wearable Lower Exoskeleton for Walking Enhancement. *Journal of Vibration and Control*, v. 12, n. 12 1311-1336.

Agrawal, A. et al, 2005. Design of a Two Degree-of-freedom Ankle-Foot Orthosis for Robotic Rehabilitation. In *Proceedings of the 9th International Conference on Rehabilitation Robotics* - ICORR 2005 41- 44.

Ferris, D.P., Czerniecki, J.M., Hannaford, B., 2005. An ankle-foot orthosis powered by artificial pneumatic muscles. *Journal of Applied Biomechanics*, v. 21 189-197.

Latzke, T., Behnke, S., Bennewitz, M. Imitative, 2007. Reinforcement Learning for Soccer Playing Robots. RoboCup 2006: Robot Soccer World Cup X, *Lecture Notes in Artificial Intelligence*, Springer Verlag, Berlin . To Appear.

Renner, R., Behnke, S., 2006. Instability Detection and Fall Avoidance for a Humanoid using Attitude Sensors and Reflexes. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* 2967-2973.

Li, X., Zell, A. H., 2007. Filtering for a Móbile Robot Tracking a Free Rolling Ball. RoboCup 2006: Robot Soccer World Cup X, *Lecture Notes in Artificial Intelligence*, Springer Verlag, Berlin. To Appear.

Lange, S., Riedmiller, M., 2006. Appearance-Based Robot Discrimination using Eigenimages. RoboCup 2006: Robot Soccer World Cup X, *Lecture Notes in Artificial Intelligence*, Springer Verlag, Berlin. To Appear.

Iocchi, L., 2007 Robust Color Segmentation through Adaptive Color Distribution Transformation. RoboCup 2006: Robot Soccer World Cup X, *Lecture Notes in Artificial Intelligence*, Springer Verlag, Berlin. To Appear.

Herrero-Pérez, D., Martínez-Barberá, H., 2007 Robust and Efficient Field Features Detection for Localization. RoboCup 2006: Robot Soccer World Cup X, *Lecture Notes in Artificial Intelligence*, Springer Verlag, Berlin. To Appear.

Nisticó, W. et al., 2007 Cooperative Visual Tracking in a Team of Autonomous Mobile Robots. RoboCup 2006: Robot Soccer World Cup X, *Lecture Notes in Artificial Intelligence*, Springer Verlag, Berlin. To Appear.

Ludington, B. et al., 2006. Target Tracking with Unmanned Aerial Vehicles: From Single to Swarm Vehicle Autonomy and Intelligence. In *Proceedings of the 14th Mediterranean Conference on Control and Automation* 1-6.

Arambel, P. et al., 2004. A Multiple-Hypothesis Tracking of Multiple Ground Targets from Aerial Video with Dynamic Sensor Control. In *Proceedings of the 7th International Conference on Information Fusion* 1080-1087.

Umemura, S.; Murakami, K. and Naruse, T., 2006. Orientation Extraction and Identification of the Opponent Robots in RoboCup Small-Size League. RoboCup 2006: Robot Soccer World Cup X, *Lecture Notes in Artificial Intelligence*, Springer Verlag, Berlin. To Appear.

Bruce, J. and Veloso, M., 2007. Real-Time Randomized Motion Planning for Multiple Domains. RoboCup 2006: Robot Soccer World Cup X, *Lecture Notes in Artificial Intelligence*, Springer Verlag, Berlin. To Appear.

Nguyen-Duc, M., Briot, J.-P., Drogoul, A., 2003 An application of Multi-Agent Coordination Techniques in Air Traffic Management. In *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology* 622-265.

Jennings N. and M .Wooldridge M., 1998 Agent Technologies: Foundations, Applications and Markets. Springer.

COLLABORATION VS. OBSERVATION: AN EMPIRICAL STUDY IN MULTIAGENT PLANNING UNDER RESOURCE CONSTRAINT

Emmanuel Benazera
LAAS-CNRS, Université de Toulouse, 7 av. du Colonel Roche, 31077 Cedex 4, Toulouse, France
ebenazer@laas.fr

Keywords: Multiagent planning, Decentralized Markov Decision Processes, Resource constraints, Empirical study.

Abstract: A team of robots and an exploratory mission are modeled as a multiagent planning problem in a decentralized decision theoretical framework. In this application domain, agents are constrained by resources such as their remaining battery power. In this context, there is an intrinsic relation between collaboration, computation and the need for the agents to observe their resource level. This paper reports on an empirical study of this relationship.

1 INTRODUCTION

Among formal models for the control of collaborative multiagent systems, decision-theoretic planning has focused on Markov Decision Problems (MDPs) (Boutilier et al., 1999). There exist several multiagent extensions to the MDP framework. The decentralized MDP framework (DEC-MDP) represents agents whose knowledge is partial and that act relatively to their local models of the world. An even more general framework is the decentralized partially observable MDPs (DEC-POMDPs) where individual agents do not fully observe their portions of the world (Bernstein et al., 2002). The multiagent control problem where agents have both stochastic actions and internal continuous state-spaces can be represented as decentralized hybrid MDP (DEC-HMDP). By hybrid it is meant that it involves both continuous and discrete variables. DEC-HMDPs are related to DEC-POMDPs with the difference that the former decide in the observation space whereas the later decide in the belief space.

Many real-world planning applications that involve teams of agents can be modeled as DEC-HMDPs. Our application domain is that of teams of exploratory robots. The interest in building and controlling teams of agents is motivated by an expected increase in both the overall capabilities and the robustness of the system. In our application domain, the continuous state-space of an agent represents its available level of resource, such as battery power and remaining time for the mission. A consequence of the resource constrained nature of the agents is that each of them is rarely able to achieve all the tasks in a mission. It follows that agents have to pay close attention to their resource levels before taking decisions such as achieving one task or the other. In this context, what the designers of multiagent robotic missions and systems may not foreknow is that there is an intrinsic relation between collaboration, computation and the need for the agents to observe their local world. In other words, the amount by which the agents constrain each others, that is the level of collaboration allowed, affects the need for observation of the agent local worlds, as well as the difficulty of computing an optimal controller for the team. This has potential consequences on the design of both missions and robots themselves.

Modern algorithms allow solving DEC-HMDPs with a small number of agents (Becker et al., 2004; Petrik and Zilberstein, 2007). This paper's focus is not on the computational techniques for DEC-HMDPs but rather on the form of their solutions and the light they shed on the relation between collaboration, computation and the need for agents to observe their local world. Through simulations and tests, empirical evidences are given of the structured relationship between collaboration, computation and the need for observation. The first half of the paper lay the required background for understanding the resource constrained DEC-HMDP framework. The second half reports on a series of experiments and empirically establishes a few useful facts that connect collaboration, computation and the need for observation.

2 BACKGROUND

We give with a brief overview of decentralized hybrid Markov decision processes, and its resource constrained variant. The reader interested in more detailed description is referred to (Becker et al., 2004).

2.1 Decentralized Hybrid Markov Decision Process and Resource Constraints

A team of m agents is modeled as a DEC-HMDP. It is defined by a tuple (N,X,A,ϕ,R,N_0). N is a set of m discrete variables N_i,$i = 1,\cdots,m$, that refer to each agent i discrete component, and n_i denotes a discrete state in N_i. $X = \bigotimes_{i=1}^{m} X_i$ is the continuous state space, and x_i denotes a continuous state in state-space X_i. $A = A_1 \times \cdots \times A_m$ is a finite set of joint actions. $\phi = \phi_1 \times \cdots \times \phi_m$ are joint transition functions. ϕ is decomposed into the discrete marginals $P(n' \mid n,x,a)$ and the continuous conditionals $P(x' \mid n,x,a,n')$. For all (n,x,a,x') it holds $\sum_{n' \in N} P(n' \mid n,x,a) = 1$ and $\int_{x \in X} P(x' \mid n,x,a,n')dx = 1$.

$R_n(x)$ denotes the reward obtained in joint state $(n;x)$ where $n \in N, x \in X$. N_0 is the initial discrete state, with initial distributions $P_0(x_i)$ for each agent $i = 1,\cdots,m$ and $P_0(x) = \bigotimes_{i=1}^{m} P_0(x_i)$.

In our application domain, continuous variables model non-replenishable resources. This translates into the assumption that the value of the continuous variables is non increasing. Each resource is internal to an agent and is thus independent of other agent resources. It is thus assumed that an agent action has no effect on other agent resource states. In this work we rely on the stronger assumption that the DEC-HMDP is *transition independent*, that is an agent action has no effects on other agent discrete state as well (Becker et al., 2004). This assumption greatly simplifies the computation and adds useful properties to the framework.

An m-agents transition independent DEC-HMDP (TI-DEC-HMDP) is a DEC-HMDP such that for $a \in A$, $n \in N$, $x \in X$, ϕ is such that

$$\forall i = 1,\cdots,m, \begin{cases} P(n_i' \mid n,x,a) & = P(n_i' \mid n_i,x_i,a) \\ P(x_i' \mid n,x,a,n') & = P(x_i' \mid n_i,x_i,a,n_i'). \end{cases}$$

In the rest of the paper, we consider an m-agents resource constrained TI-DEC-HMDP (RC-TI-DEC-HMDP).

2.2 Goals, Policy and Reward

Agents operate in a decentralized manner, and choose their actions according to their local view of the world. They do not communicate but are cooperative, i.e. there is a single value function for all agents. We assume a set of identified global goals $\{g_1,\cdots,g_k\}$, each of which is known and corresponds to an achiement by one or more agents. Each $g_j \in N$ is such that $g_j = \{g_{ij}\}_{i=\alpha_j^1,\cdots,\alpha_j^{q_j}}$ where $g_{ij} \in N_i$ and $\alpha_j^q \in \{1,\cdots,m\}$. For simplifying notations, we note $i \in g_j$ the fact that $g_{ij} \in g_j$, i.e. agent i is involved in goal state g_j. The reward function for a RC-TI-DEC-HMDP is decomposed into two components for each goal j: a set of individual local reward functions for each agent, the $R_{g_{ij}}(x_i)$; a joint reward $c_j(x)$ the team receives and that depends on the actions of more than one agent.

The joint reward articulates the interaction among agents. In general agents seek to maximized the local and joint reward functions. In this case negative c_j such as in our case study (see 2.3) can be seen as a penalty put on some agent interactions. Positive c_j naturally favor certain other interactions.

Given a RC-TI-DEC-HMDP, we define a policy $\pi = \{\pi_1,\cdots,\pi_m\} : (N,X) \to A$ to be a mapping from the spate space to the action space. A global value function $GV : (N,X) \to \Re$ gives the expected total reward of the system starting from an initial state and acting according to the policy π until termination. Termination occurs whenever all goals are achieved or all agents have run out of resources. Similarly, the local value function $V^i : (N_i,X_i) \to \Re$ gives the expected total reward for agent i, and the joint value function $JV : (N,X) \to \Re$ gives the joint expected total reward of the system. The joint value function is given by

$$JV(x \mid \pi_1,\cdots,\pi_m) = \sum_{j=1}^{k} c_j(x) \prod_{i \in g_j} P_{g_{ij}}(x_i \mid \pi_i) \quad (1)$$

where the c_j are the joint rewards, and $P_{g_{ij}}(x_i \mid \pi_i)$ is the probability agent i has to achieve goal j according to the policy π_i. Often the joint rewards are in fact penalties. The global value function is given by

$$GV(x \mid \pi_1,\cdots,\pi_m) = \sum_{i=1}^{m} V_0^i(x_i) + JV(x \mid \pi_1,\cdots,\pi_m) \quad (2)$$

where $V_0^i(x_i)$ is the local value function for the initial state of agent i. The optimal joint policy is noted $\pi^* = \{\pi_1^*,\cdots,\pi_m^*\}$, given by

$$\pi^* = \underset{\pi_1,\cdots,\pi_m}{\operatorname{argmax}} E_X[GV(x \mid \pi_1,\cdots,\pi_m)] \quad (3)$$

where E_X denotes the expectation over state space X. The optimal value function is $GV^*(x \mid \pi_1^*,\cdots,\pi_m^*) = \max_{\pi_1,\cdots,\pi_m} GV(x \mid \pi_1,\cdots,\pi_m)$. Note that action Abort_i ends the policy of agent i.

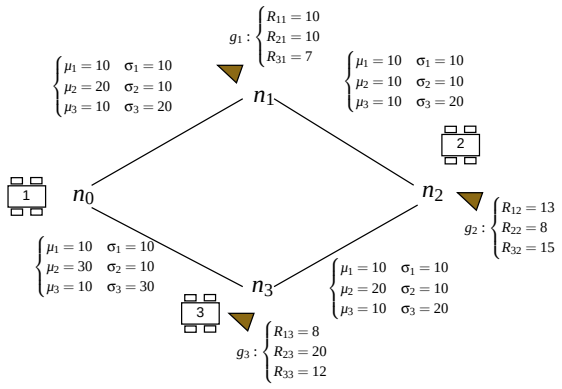


Figure 1: Case-study: a 3 agents / 3 goals problem.

2.3 Case-study

Consider the problem on figure 1. Three rovers share three rocks over four locations. The (μ_i, σ_i) are the standard mean and variance of the Gaussian distribution that models resource consumption $P(x_i' \mid x_i, n, n', a)$ of rover i on each path (n, n'). $R_{g_{ij}}$ is the reward function for achieving goal j and agent i. Each rover starts from a different initial location.
We study a joint value of the form

$$JV(x \mid \pi_1, \cdots, \pi_m) = -\beta \max_{i \in \{0,\cdots,m\}} \sum_{j=1}^{k} R_{g_{ij}}(x) \prod_{i \in g_j} P_{g_j}(x_i \mid \pi_i) \quad (4)$$

where $\beta \in [0,1]$. In other words, the joint reward signal $c_j(x)$ for goal j is a negative factor, or penalty, of value the maximum reward possibly obtained by an agent for that goal. The rational behind this model is that it allows to parametrize the collaboration among agents. Thus for $\beta = 1$ no collaboration is beneficial, and each agent has a high incentive of avoiding goals already achieved by other agents. When $\beta < 1$, there is incentive for all agents to consider all goals, with the amount collaboration inversely proportional to β. For this reason, in the following, we refer as *the collaboration factor* of a team of agents as a function that is *inversely proportional to* β.

2.4 Oversubscription, Conditional Policies, Branches and Observations

2.4.1 Oversubscription in Goals

In a RC-TI-DEC-HMDP, the resource constrained nature of each agent gives rise to *over-subscribed* planning problems. These are problems in which not all the goals are feasible by each agent under its internal resource constraints and the initial distribution over its resources. Their solutions are characterized by the existence of different achievable goal sets for different resource levels. In our application domain, it is assumed that each goal can be achieved only once (no additional utility is achieved by repeating the task).

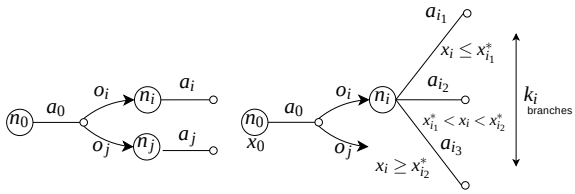


Figure 2: MDP policy (left) and HMDP policy (right).

2.4.2 Conditional Policies

As defined earlier, the policy solution to a RC-TI-DEC-HMDP is a set of individual policies, one per agent. In fact, an agent policy is solution to an underlying HMDP (Becker et al., 2004). There is no need to define this HMDP here. What we are interested in is the form taken by an HMDP policy, in general.

Most traditional planners assume a discrete state-space and a small number of action outcomes. When the model is formalized as an MDP, the planner can decide on discrete action outcomes. The policy *branches* on discrete action outcomes. A policy thus reads *from state n_0 and action a_0, when action outcome is o_i, do action a_i; else when action outcome is o_j, do action a_j; ....* A MDP solution policy is pictured on the left-hand side of figure 2.

When the model includes continuous resources and is modeled as an HMDP, a consequence of oversubscription is that a HMDP policy is conditional upon resources. Thus the planner must branch not only on the discrete action outcomes, but on the availability of continuous resources. In this case, a policy reads *from discrete state n_0, continuous resource x_0 and action a_0, when action outcome is o_i, then if continuous resource is $x_i \leq x^*_{i_1}$, do action a_{i_1}, else if continuous resource is $x^*_{i_1} < x_i < x^*_{i_2}$, do action a_{i_2}, etc...; else when action outcome is o_j....* The right-hand side of figure 2 pictures a portion of an HMDP policy.

2.4.3 Observation, Collaboration and Computation

Now, the important point is that each branch of an HMDP policy calls for an observation of the agent resource state. Each observation is to be carried out at execution time. Because of the oversubscribed nature of the planning problem, each agent has to make a

Table 1: Empirical measures and related information on the underlying planning problem.

Empirical measure	Information
OCS	Dependency of one agent on the other agents' policies
Local plans studied	Dependency of agent on the other agents' policies. Complexity of the local decision problem.
Joint policies studied	Computational difficulty of finding of the joint controller. Dependencies among all agents.
Branches in the optimal joint policy	Observations of their local resource states by the agents.
Size of the optimal joint policy	Repartition of goals (rocks) among agents.
Discretization (number of pieces)	Complexity of the decision problem (local or global).

certain number of observations before deciding which goals to achieve. In the multiagent framework, the collaboration among agents and its possible penalties affects the repartition of goals, and thus the need for observation of its resource state by each agent. As a consequence, this also affects the computational weight of finding an optimal policy for a team of agents. The rest of this paper reports on the results of a series of simulations and tests that yield empirical evidences of the relation between collaboration, observation and computation.

3 COMPUTATION AND COMPLEXITY

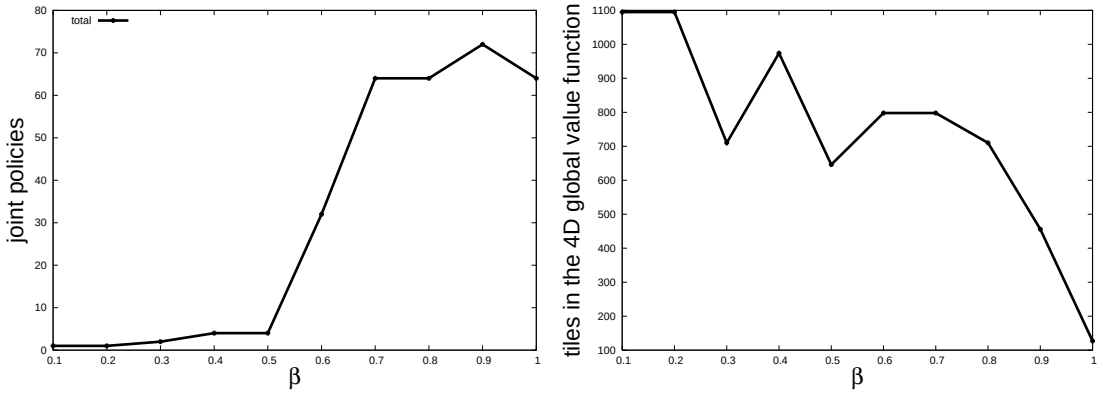
3.1 Solving RC-TI-DEC-HMDPs

Here we give a little background on the solving of an *m*-agents RC-TI-DEC-HMDPs. The Cover Set Algorithm (CSA) is an efficient algorithm that finds optimal policies (Becker et al., 2004; Petrik and Zilberstein, 2007). It is a two steps algorithm. The first step consists in finding a set of policies for each of $(m-1)$ agents, called the optimal cover set (OCS). Each agent's OCS is such that for any of the other agent's policies it contains at least a policy that is optimal. In other words, the OCS of an agent is guaranteed to contain the optimal policy for this agent that belongs to the optimal policy for the team. In computing the OCS for an agent, the CSA has to study a number of competing local policies for this agent. This number yields an information on the dependency of the agent w.r.t. the other agent policies. The second step iterates all combinations of policies in the $(m-1)$ OCS, computes an optimal policy for the *m*-th agent, and returns the combination of *m* policies that yields the maximal expected global value (2). Table 1 sums up the empirical measures and their information on the underlying planning problem.

Computationally, the challenging aspect of solving an HMDP is the handling of continuous variables, and particularly the computation of the so-called Bellman optimality equation. At least two approaches, (Feng et al., 2004) and (Li and Littman, 2005) exploit the structure in the continuous value functions of HMDPs. Typically these functions appear as a collection of humps and plateaus, where the later correspond to a region in the continuous state space where similar goals are pursued by the policy. The steepness of the slope between plateaus reflects the uncertainty in achieving the underlying goals. The algorithms used for producing the results analyzed in this paper exploit a problem structure where the continuous state can be partitioned into a finite set of regions. Taking advantage of the structure relies on grouping those states that belong to the same plateau, while dynamically scaling the discretization for the regions of the state space where it is most useful such as in between plateaus. It follows that the dynamic discretization of the continuous state-space reflects the complexity of the decision problem: the less discretized pieces, the easiest the decision, see Table 1.

3.2 Empirical Evidences

This section reports on planning for our case-study. It helps understanding the relation between collaboration and computation. Figure 3 reports on the computation of the optimal joint policy. Figure 3(a) shows the number of joint policies studied for selecting the optimal joint policy. This number jumps with the reduction of the collaboration factor among agents that is implicitely carried by the joint reward structure. One hypothesis is that the problem becomes globally more computational when the amount of collaboration among agents is reduced. In fact, this hypothesis is confirmed by the results on figure 3(b). The number of discretized regions in the optimal four-dimensional global value function reflects the discretization of the optimal value functions of individuals. The finer the discretization, the more complex and thus the more



(a) Number of studied joint policies.

(b) Dynamic discretization of the optimal global value function.

Figure 3: Case-study: joint policy computation.

Table 2: Case-study: Optimal joint policy.

β	local policies studied	joint policy size	branches	unused agents
1	113256	17	5	1
0.9	89232	20	5	0
<0.9	68640	> 20	5	0

difficult the decision problems at the level of individual agents. The very clear drop in the number of regions with the reduction of the collaboration factor among agents corroborates our hypothesis: low collaboration puts the stress on the global controller and relieves the individuals. On the opposite, when β moves toward 0 and collaboration is high, each agent has an incentive to visit all rocks.

Fact 1. *The computational difficulty of finding the global controller for a team of resource constrained agents is a decreasing function of their collaboration factor.*

Table 2 characterizes the optimal joint policy for our case-study. These numbers confirm the trend observed in other figures: agents involved in less collaborative problems (i.e. $\beta \approx 1$) are more dependent on the strategies of others since they are forced to avoid the goals possibly achieved by others.

Interestingly, for $\beta = 1$, agent 1 is useless, that is its policy is empty. In fact, agent 1 is dominated, i.e. the two other agents do what it does, and do it better. This is an indication that in larger problems, with more agents, heavy computations might lead to a empty optimal policies for certain agents. The next section studies the number of branches and the relation between collaboration and observation.

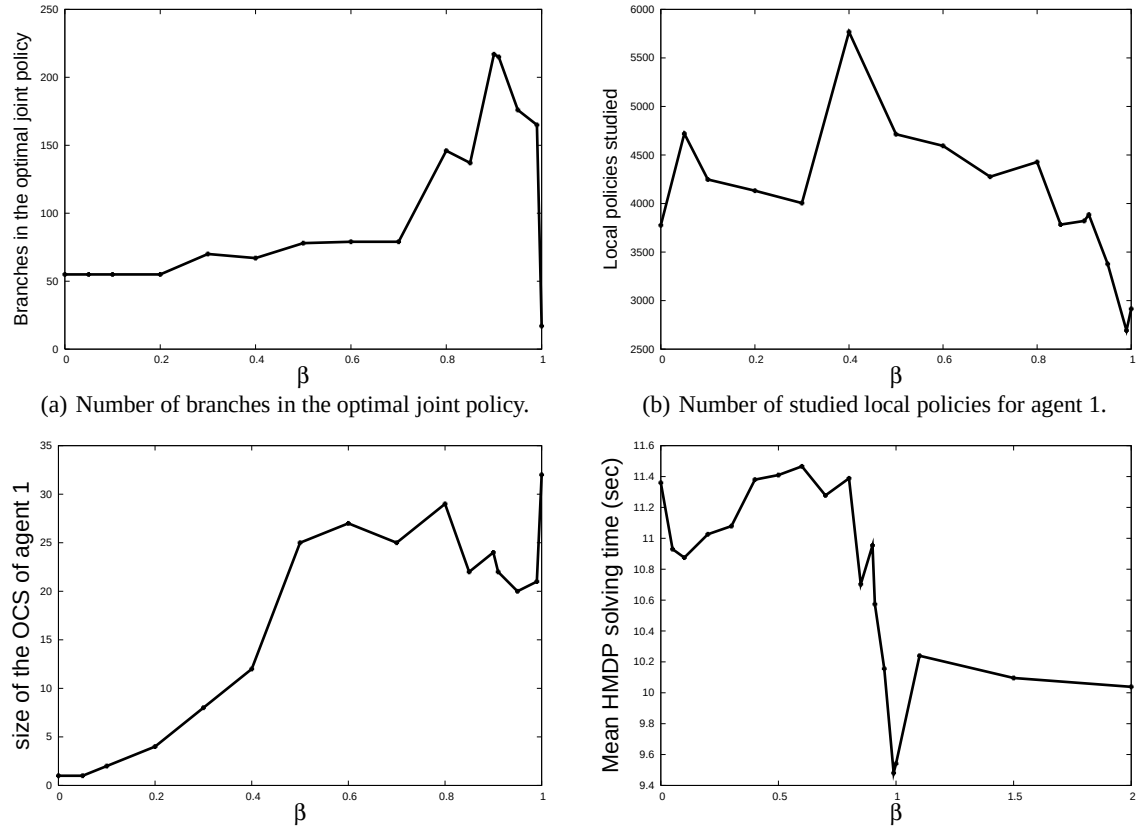
4 COLLABORATION AND OBSERVATION

In this section we empirically study the relation between collaboration and the need for observation at agent level. To this end, we have considered a five rocks, two agents problem in the Mars rovers domain. We have varied the collaboration factor β for this problem. Figure 4(a) reports on the number of branches in the optimal joint policy for $\beta \in [0, 1]$. The number of branches is the number of times the policy asks for an observation of the level of continuous resources before acting. We see a four times increase of the number of branches, which reflects a growing need for the individual agents to observe their internal resource state and cast away uncertainty[1].

Fact 2. *The number of observations required by the optimal policy of a team of resource constrained agents is a decreasing function of their collaboration factor.*

Now, consider figure 4(b) that shows the number of local policies that are studied decreases when β increases. This number is a function of two variables: i/the discretization of the continuous space; ii/the structural dependency on other agent strategies. Figure 4(c) shows that the size of the coverage set of agent 1. It shows that the number of policies in the OCS of agent 1 augments, and does not significantly decreases when β increases. This means that agent 1 grows more dependent on agent 2 when β increases. This is because when collaboration becomes more pe-

[1]The ratio (branches/number of actions in the optimal joint policy) remains rather constant, thus reflecting the structure of the problem: in most cases decision is taken before navigation to a rock.



(a) Number of branches in the optimal joint policy.

(b) Number of studied local policies for agent 1.

(c) Number of policies in the OCS of agent 1.

(d) Mean time needed to solve agent 1 local policies.

Figure 4: Collaboration vs. Observation: report on a 2 agents, 5 rocks problem from the Mars rovers domain.

nalized, agent 1 has to be increasingly aware of agent 2's strategy before it takes action, thus mitigating its potentially negative effect on the global reward.

It follows from i/ and ii/ that this is the discretization of the continuous space that becomes less dense when β increases. Equivalently, this indicates that the underlying local decision problems are less complex. Less formally, this means that the world becomes increasingly sharper for the individual agents, with value functions that exhibit more plateaus and less slopes. Metaphorically, the world, as seen by each individual agent, turns into a *black & white* decisional space, where rocks must be clearly partitioned among agents, and collaboration tends to be avoided. In other words, there is less room for uncertainty, and risk is agressively eliminated, as early as possible. This correlates naturally with the higher number of observations required by the optimal joint policy. Each observation disambiguates the reachability of each rock and sharpens the view of an otherwise very stochastic world.

Fact 3. *The complexity of the agent local decision problems is an increasing function of their collaboration factor.*

In parallel, since local decision is sharper, it increasingly needs to be articulated with that of other agents. A consequence is that with increasing β, the decisional stress is increasingly shifted to the global controller. We had already noticed this behavior in section 3. Here we choose to observe the side-effect that is a relief of the computational weight that is put on individual agents. Figure 4(d) shows the mean time needed to solve an augmented HMDP for different values of β. The sudden decrease indicates the decisional shift from the local controllers to the global controller.

Fact 4. *A decrease in the collaboration factor for a team of resource constrained agents implies a shift of the computational weight from the local controllers to the global team controller.*

To summarize, agents, each with eclectic abilities, acting in a specialized world in which collaboration is not well valued, are forced to agressively decide upon their objectives more often, while the final computational burden is shifted to the global controller that governs them.

5 CONCLUSIONS

We have reported on an empirical study of the connections between collaboration, computation and the need for observation in optimal policies for resource constrained multiagent problems. These problems well model number of real-world situations for modern teams of robots. This includes our application domain, that of team of exploratory rovers.

We have defined the collaboration as the positive value given to interactions among agents in a team. Interestingly, we could show that the need for observation is a decreasing function of the collaboration among agents. We can sum up our empirical finding by considering a world where the division of labor is extreme, and collaboration not much valuable. In this world, resource constrained individuals with eclectic abilities (i.e. that are equally able with every task), are stressed to take sharp decisions, more often, and based on recurrent observations of their own resources.

REFERENCES

Becker, R., Zilberstein, S., Lesser, V., and Goldman, C. (2004). Solving transition independent decentralized markov decision processes. *Journal of Artificial Intelligence Research*, 22.

Bernstein, D., Givan, R., Immerman, N., and Zilberstein, S. (2002). The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4).

Boutilier, C., Dean, T., and Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11.

Feng, Z., Dearden, R., Meuleau, N., and Washington, R. (2004). Dynamic programming for structured continuous Markov decision problems. In *Proceedings of the Twentieth International Conference on Uncertainty In Artificial Intelligence*, pages 154–161.

Li, L. and Littman, M. (2005). Lazy approximation for solving continuous finite-horizon mdps. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*.

Petrik, M. and Zilberstein, S. (2007). Anytime coordination using separable bilinear programs. In *Proceedings of the Twenty Second National Conference on Artificial Intelligence*.

TRILATERATION LOCALIZATION FOR MULTI-ROBOT TEAMS

Paul M. Maxim[1], Suranga Hettiarachchi[2], William M. Spears[1], Diana F. Spears[1]
Jerry Hamann[1], Thomas Kunkel[1] and Caleb Speiser[1]
[1] *University of Wyoming, Laramie, Wyoming 82070, U.S.A.*
[2]*Eastern Oregon University, La Grande, Oregon 97850, U.S.A.*
{*paulmax, wspears, dspears, hamann, caleb485*}*@uwyo.edu, http://www.cs.uwyo.edu/ ~wspears*
suranga.hettiarachchi@eou.edu

Keywords: Localization, trilateration, formations, distributed, outdoor.

Abstract: The ability of robots to quickly and accurately localize their neighbors is extremely important for robotic teams. Prior approaches typically rely either on global information provided by GPS, beacons and landmarks, or on complex local information provided by vision systems. In this paper we describe our trilateration approach to multi-robot localization, which is fully distributed, inexpensive, and scalable (Heil, 2004). Our prior research (Spears et. al, 2006) focused on maintaining multi-robot formations indoors using trilateration. This paper pushes the limits of our trilateration technology by testing formations of robots in an outdoor setting at larger inter-robot distances and higher speeds.

1 INTRODUCTION

The main contributions of this paper are: (1) a presentation of our trilateration approach to multi-robot localization (i.e., each robot locates its neighbors), and (2) a set of experimental results obtained with our trilateration approach under outdoor conditions. These experimental results highlight the advantages of our approach and clarify its limitations. The outdoor experiments are conducted in an environment with varying terrain (e.g., grass, dirt, and concrete), rocks, protruding tree roots, leaves, pine cones and other ground protrusions. Also, there was a considerable amount of dust and wind (over 9 meters per second). Despite this, the robots are able to maintain high quality formations.

The organization of this paper is as follows. Section 2 introduces our trilateration approach to localization, which is fully distributed and assumes that each robot has its own local coordinate frame (i.e., no global information is required). Each robot determines its neighbors' range and bearing with respect to its own egocentric, local coordinate system. After such localization, sensor values and other data can be exchanged between robots in a straightforward manner. Next, sections 3, 4 and 5 describe our trilateration implementation and current robot platforms. Sections 6 and 7 present results from our experiments. Section 8 summarizes and concludes the paper.

2 LOCALIZATION VIA TRILATERATION

The purpose of our trilateration technology is to create a plug-in hardware module to accurately localize neighboring robots, without global information and/or the use of vision systems. Our localization technology does not preclude the use of other technologies. Beacons, landmarks, vision systems, GPS (Borenstein et. al, 1996), and pheromones are not necessary, but they can be added if desired. It is important to note that our trilateration approach is not restricted to one particular class of control algorithms – it is useful for behavior-based approaches (Balch and Hybinette, 2000), control-theoretic approaches (Fax and Murray, 2004; Fierro et. al, 2002), motor schema algorithms (Brogan and Hodgins, 1997), and physicomimetics (Spears et. al, 2004; Zarzhitsky et. al, 2005; Hettiarachchi, 2007).

In 2D *trilateration*, the locations of three base points are known as well as the distances from each of these three base points to the object to be localized. Looked at visually, 2D trilateration involves finding the location where three circles intersect. Thus, to locate a robot using 2D trilateration the sensing robot must know the locations of three points in its own coordinate system and be able to measure distances from these three points to the sensed robot.

2.1 Measuring Distance

Our distance measurement method exploits the fact that sound travels significantly more slowly than light, thereby enabling us to employ a Difference in Time of Arrival technique. To tie this to 2D trilateration, assume that each robot has one radio frequency (RF) transceiver and three ultrasonic acoustic transducers. The ultrasonic transducers are the "base points." Suppose robot 2 simultaneously emits an RF pulse and an ultrasonic acoustic pulse. When robot 1 receives the RF pulse (almost instantaneously), a clock on robot 1 starts. When the acoustic pulse is received by each of the three ultrasonic transducers on robot 1, the elapsed times are computed. These three times are converted to distances, according to the speed of sound. Because the locations of the acoustic transducers are known, robot 1 is now able to use trilateration to compute the location of robot 2 (precisely, the location of the emitting acoustic transducer on robot 2). Of the three acoustic transducers, all three must be capable of receiving, but only one must be capable of transmitting.

Measuring the elapsed times is not difficult. Since the speed of sound is roughly 340.2 meters per second at standard temperature and pressure, it takes approximately 2.9 ms for sound to travel one meter. Times of this magnitude are easily measured using inexpensive electronic hardware.

2.2 Channeling Acoustic Energy into a Plane

Ultrasonic acoustic transducers produce a cone of energy along a line perpendicular to the surface of the transducer. The width of this main lobe (for the inexpensive 40 kHz transducers used in our implementation) is roughly 30°. To produce acoustic energy in a 2D plane would require 12 acoustic transducers in a ring. To get three base points would hence require 36 transducers. This is expensive and is a large power drain. We adopted an alternative approach. Each base point is comprised of one acoustic transducer pointing downward. A parabolic cone (Heil, 2004) is positioned under the transducer, with its tip pointing up toward the transducer (see Figure 2 later in this paper). The parabolic cone acts like a lens. When the transducer is placed at the virtual "focal point" the cone "collects" acoustic energy in the horizontal plane, and focuses this energy to the receiving acoustic transducer. Similarly, a cone also functions in the reverse, reflecting transmitted acoustic energy into the horizontal plane. This works extremely well – the acoustic energy is detectable to a distance of 3.5 m. which is adequate for our needs. Greater range can be obtained with more power (the scaling appears to be quite manageable).

2.3 Related Work

Trilateration is a well-known technique for robot localization. Most approaches (including ours) are algebraic, although recently a geometric method was proposed (Thomas and Ros, 2005). Many localization techniques, including those involving trilateration, use global coordinates (Peasgood et. al, 2005); however ours relies on local coordinates only.

MacArthur (MacArthur, 2003) presents two different trilateration systems. The first uses three acoustic transducers, but without RF. Localization is based on the differences between distances rather than the distances themselves. The three acoustic transducers are arranged in a line. The second uses two acoustic transducers and RF in a method similar to our own. Unfortunately, both systems can only localize points "in front" of the line.

Cricket (Nissanka, 2005) is another system that makes use of RF and ultrasound for localization. It was developed to be used indoors. Compared to our system, which does not require fixed beacons, the Cricket requires beacons attached to fixed locations in order to function. This is not practical for mobile robot localization in outdoor environments.

Our particular approach was inspired by the CMU *Millibot* project. They also use RF and acoustic transducers for trilateration. However, due to size limitations, each Millibot has only one acoustic transducer (coupled with a right-angle cone, rather than the parabolic cone we use). Hence trilateration is a collaborative endeavor that involves several robots. To perform trilateration, a minimum of three Millibots must be stationary and serve as beacons at any moment in time. The set of three stationary robots changes as the robot team moves. The minimum team size is four robots (and is preferably five). Initialization generally involves having some robots make L-shaped maneuvers, in order to disambiguate the localization (Navarro-Serment, 1999). Our approach operates with as few as two robots (but is scalable to an arbitrary number), due to the presence of three acoustic transducers on each robot (see below).

In terms of functionality, an alternative localization method in robotics is to use line-of-sight infrared (IR) transceivers. When IR is received, signal strength provides an estimate of distance. The IR signal can also be modulated to provide communication. Multiple IR sensors can be used to provide the bearing to the transmitting robot (e.g., see (Rothermich

et. al, 2004; Payton et. al, 2004)). We view this method as complementary to our own; however, our method is more appropriate for tasks where greater localization accuracy is required. This is especially important in outdoor situations where water vapor or dust could change the IR opacity of air. Similar issues arise with the use of cameras and omni-directional mirrors/lenses, which require far more computational power and a light source.

3 OUR TRILATERATION APPROACH

Our trilateration approach to localization is illustrated in Figure 1. Assume two robots, shown as circles. An RF transceiver is in the center of each robot. Each robot has three acoustic transducers (also called *base points*), labeled **A**, **B**, and **C**. Note that the robot's local XY coordinate system is aligned with the L-shaped configuration of the three acoustic transducers, as shown in the figure. *Note, Y points to the front of the robot.*

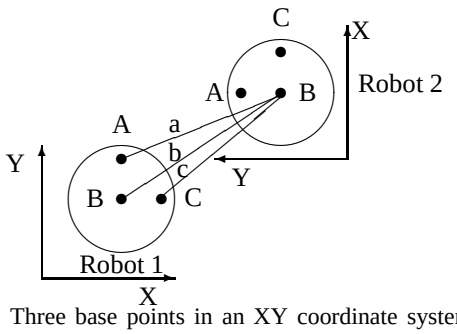


Figure 1: Three base points in an XY coordinate system pattern.

In Figure 1, robot 2 simultaneously emits an RF pulse and an acoustic pulse from its transducer **B**. Robot 1 then measures the distances **a**, **b**, and **c**. Without loss of generality, assume that transceiver **B** of robot 1 is located at $(x_{1B}, y_{1B}) = (0,0)$ (Heil, 2004).[1] In other words, let **A** be at $(0,d)$, **B** be at $(0,0)$, and **C** be at $(d,0)$, where d is the distance between **A** and **B**, and between **B** and **C** (see Figure 1).

For robot 1 to determine the position of **B** on robot 2 within its own coordinate system, it needs to find the simultaneous solution of three nonlinear equations, the intersecting circles with centers located at **A**, **B** and **C** on robot 1 and respective radii of **a**, **b**, and **c**:

$$(x_{2B} - x_{1A})^2 + (y_{2B} - y_{1A})^2 = a^2 \quad (1)$$
$$(x_{2B} - x_{1B})^2 + (y_{2B} - y_{1B})^2 = b^2 \quad (2)$$
$$(x_{2B} - x_{1C})^2 + (y_{2B} - y_{1C})^2 = c^2 \quad (3)$$

Given the transducer configuration shown above, we get (Heil, 2004):

$$x_{2B} = \frac{b^2 - c^2 + d^2}{2d} \qquad y_{2B} = \frac{b^2 - a^2 + d^2}{2d}$$

An interesting benefit of these equations is that they can be simplified even further, if one wants to trilaterate purely in hardware (Spears et. al, 2006).

By allowing robots to share coordinate systems, robots can communicate their information arbitrarily far throughout a robotic network. For example, suppose robot 2 can localize robot 3. Robot 1 can localize only robot 2. If robot 2 can also localize robot 1 (a fair assumption), then by passing this information to robot 1, robot 1 can now determine the position of robot 3. Furthermore, robot orientations can also be determined. Naturally, localization errors can compound as the path through the network increases in length, but multiple paths can be used to alleviate this problem to some degree. Heil (Heil, 2004) provides details on these issues.

In addition to localization, our trilateration system can also be used for data exchange. Instead of emitting an RF pulse that contains no information but only performs synchronization, we can also append data to the RF pulse. Simple coordinate transformations allow robot 1 to convert the data from robot 2 (which is in the coordinate frame of robot 2) to its own coordinate frame.

4 TRILATERATION IMPLEMENTATION

Figure 2 illustrate how our trilateration framework is currently implemented in hardware. Figure 2 (left) shows three acoustic transducers pointing down, with reflective parabolic cones. The acoustic transducers transmit and receive 40 kHz acoustic signals.

Figure 2 (middle) shows our in-house acoustic sensor boards (denoted as "XSRF" boards, for *Experimental Sonic Range Finder*). There is one XSRF board for each acoustic transducer. The XSRF board calculates the time difference between receiving the RF signal and the acoustic pulse. Each XSRF contains 7 integrated circuit chips. A MAX362 chip controls whether the board is in transmit or receive mode.

[1] Subscripts denote the robot number and the acoustic transducer. The transducer **A** on robot 1 is located at (x_{1A}, y_{1A}).

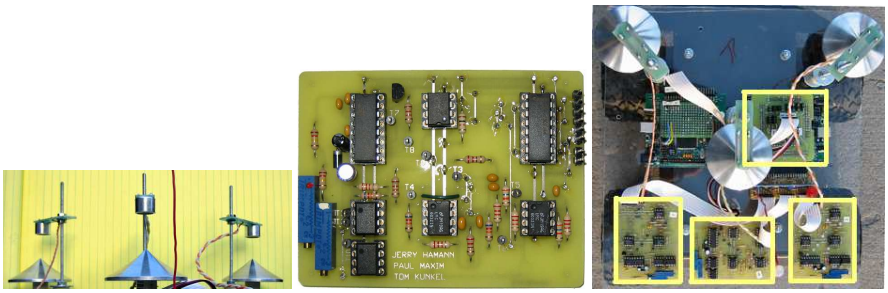


Figure 2: The acoustic transducers and parabolic cones (left). The XSRF acoustic sensor printed circuit board (middle), and the completed trilateration module (top-down view, right).

When transmitting, a Microchip PIC microprocessor generates a 40 kHz signal. This signal is sent to an amplifier, which then interfaces with the acoustic transducer. This generates the acoustic signal.

In receive mode, a trigger indicates that an RF signal has been heard and that an acoustic signal is arriving. When the RF is received, the XSRF board starts counting. To enhance the sensitivity of the XSRF board, three stages of amplification occur. Each of the three stages is accomplished with a LMC6032 operational amplifier, providing a gain of roughly 15 at each stage. Between the second and third stage there is a 40 kHz bandpass filter to eliminate out-of-bound noise that can lead to saturation. The signal is then passed to two comparators, set at thresholds of $\pm$ 2 VDC. When the acoustic energy exceeds either threshold, the XSRF board finishes counting, indicating the arrival of the acoustic signal.

The timing counts provided by each of the XSRF boards is sent to a MiniDRAGON[2] powered by a Freescale 68HCS12 microprocessor that performs the trilateration calculations. Figure 2 (right) shows the completed trilateration module from above. The MiniDRAGON is outlined near the center and the three XSRF boards are outlined at the bottom.

4.1 Synchronization Protocol

Trilateration involves at least two robots. One transmits the acoustic-RF pulse combination, while the others use these pulses to compute (trilaterate) the coordinates of the transmitting robot. Hence, trilateration is a one-to-many protocol, allowing multiple robots to simultaneously trilaterate and determine the position of the transmitting robot. Our "token passing" scheme to allow robots to take turns transmitting is not used for the experiments in this paper, to simplify the experimental design. Only leader/follower experiments are presented herein.

[2]Produced by Wytec (http://www.evbplus.com/)

5 Maxelbot PLATFORMS

Our University of Wyoming "Maxelbot" (named after the two graduate students who designed and built the robot) is modular. The platform is an MMP5, made by The Machine Lab [3]. Figure 3 (left) shows four Maxelbots. A primary MiniDRAGON is used for control. It communicates via an I^2C bus to all other peripherals, allowing us to plug in new peripherals as needed. Figure 3 (right) shows the architecture. The primary MiniDRAGON is the board that drives the motors. It also monitors proximity sensors and shaft encoders. The trilateration module is shown at the top of the diagram. This module controls the RF and acoustic components of trilateration. Additional modules have been built for digital compasses, thermometers, and chemical plume tracing (Spears et. al, 2006). The PIC processors provide communication with the I^2C bus.

6 TRILATERATION ACCURACY AS A FUNCTION OF VELOCITY AND DISTANCE

In (Spears et. al, 2006) we presented the accuracy of our trilateration technique on stationary robots, to a distance of one meter. In this paper we present results for a moving Maxelbot on a treadmill from 0.5 to 3.5 meters behind a stationary Maxelbot placed ahead of the treadmill, at two speeds: 0.32 m/s and 0.64 m/s. Hence we are measuring the accuracy of the performance of the whole system, including the trilateration module and our physicomimetics control algorithm. The results shown are the mean $\pm$ range of the $error \equiv ideal\_distance - measured\_distance$ (in cm), as measured physically with a ruler (see Tables 2

[3]See http://www.themachinelab.com/MMP-5.html

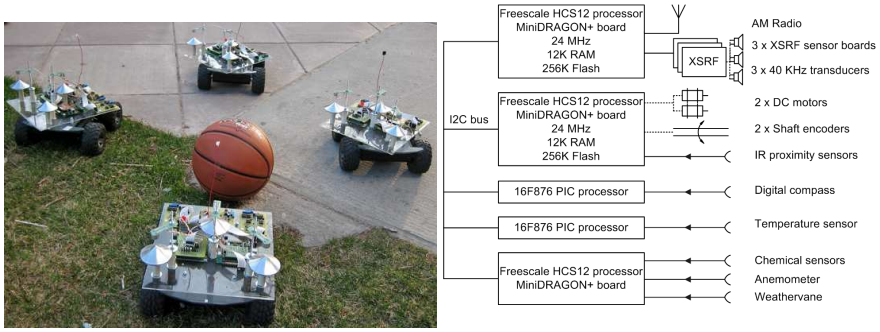


Figure 3: Maxelbots and the architecture.

Table 1: Mean and range of error of the followers' X position at different distances and velocities.

Velocity	Ideal Distance (cm)						
	50	100	150	200	250	300	350
0.32 m/s	0.0 ± 0.2	0.3 ± 0.3	-0.3 ± 0.3	-0.6 ± 0.4	0.6 ± 0.5	1.9 ± 0.6	0.6 ± 0.6
0.64 m/s	0.0 ± 0.6	0.3 ± 0.3	-1.3 ± 0.9	-1.0 ± 1.0	1.0 ± 1.0	1.9 ± 1.1	0.0 ± 1.3

and 1). Above 3.5 m the acoustic signal is lost, because the acoustic energy falls below the threshold of $\pm$ 2 VDC and hence is not detected.

The mean error in X is very small ($< 1\%$), which means that the side-to-side position of the Maxelbot is very close to the desired position. The mean error in Y is larger, and at higher distances the Maxelbot lags more behind the desired position (but the mean error is $< 5\%$). However, note that the range in error is less at the higher velocity. The increased momentum of the robot helps filter sensor noise.

7 OUTDOOR EXPERIMENTS

This section presents three experiments that test the trilateration system outside. In particular, the Maxelbots are run in a region in the center of the University of Wyoming campus. This region consists mostly of grass, of average height 5 cm, interspersed with concrete sidewalks, trees, rocks, leaves, and other debris. The grass hits the bottom of the Maxelbot. Although generally flat, the ground slope can change rapidly (within 0.6 m), by up to 20°, at boundaries. Results presented below are averaged over five independent runs, each taken over a 20 minute interval. The speed of the robots is approximately 0.55 m/s. For these experiments we are forced to use the trilateration readings themselves as an estimate to the quality of the formation. Given the accuracy of the results in the prior section, this is a reasonable and practical approach.

7.1 Accuracy of a Linear Formation

The first experiment has three Maxelbots in a linear formation. The purpose of this experiment was to determine the effect of having the middle robot occlude the acoustic signal between the first and third Maxelbots. The first follower (middle robot) tries to keep the leader at (0 cm, 63.5 cm) with respect to its local coordinate system. The second follower tries to keep the leader at (0 cm, 145 cm) with respect to its local coordinate system.

Table 3 summarizes the quality of the results. This first follower maintains position quite well. The second follower does exhibit some difficulties due to occlusion, since it does lag a bit behind the ideal distance. However, even in this case the distance is approximately only 10% off from the ideal distance. Also, the standard deviation is acceptably low.

7.2 Accuracy of Non-Linear Formations

The second and third experiments examine the effect of position with respect to the quality of the results (trilateration accuracy can be affected by the difference in the bearing of one robot with respect to another (Heil, 2004)). We try two different configurations of the robots. In the first, there are three robots. The right follower tries to keep the leader at (-48 cm, 91 cm) with respect to its local coordinate system. The left follower tries to keep the leader at (53 cm, 91 cm) with respect to its local coordinate system. In the second configuration we use four Maxelbots in a diamond formation. In this latter experiment the wind

Table 2: Mean and range of error of the followers' Y position at different distances and velocities.

Velocity	Ideal Distance (cm)						
	50	100	150	200	250	300	350
0.32 m/s	-1.0 $\pm$ 1.6	0.0 $\pm$ 1.3	-1.3 $\pm$ 2.5	-1.9 $\pm$ 3.2	-1.3 $\pm$ 5.1	-5.0 $\pm$ 10.2	-5.0 $\pm$ 7.6
0.64 m/s	-5.7 $\pm$ 0.6	-0.6 $\pm$ 1.9	-3.8 $\pm$ 2.5	-5.1 $\pm$ 3.8	-3.2 $\pm$ 4.4	-14.0 $\pm$ 3.8	-16.5 $\pm$ 3.8

Table 3: Accuracy of the two followers' X and Y positions in a linear formation (results in cm).

	Ideal	Mean	Std. dev.
Follower1-X	0	-1.5	0.8
Follower1-Y	63.5	67.1	1.0
Follower2-X	0	0.3	4.8
Follower2-Y	145	159.8	4.1

speed near the ground ranged from 4 to 9 m/s.

Table 4 shows the XY-coordinates derived from the trilateration readings, for both configurations. From this table, it can be seen that the means are very close to the ideal. The standard deviation is somewhat higher, reflecting the more difficult environmental conditions. However, very good formations are maintained by the trilateration system despite ground disturbances, wind, dust, and relatively high speed (Y has at most an error of roughly 11%). Results are averaged over five independent runs. Thus far no position-dependent effects have been noticed (other than distance, as is expected).

7.3 Trilateration Reliability Results

A detailed data analysis has been performed on the reliability of the trilateration system during the outdoor experiments. The RF failure rate is 0.2%. The rate at which the RF pulse is received but acoustic pings are not received (at all three receivers) is only 1%. Almost every acoustic failure was isolated, and not consecutive. Consider the interpretation of these results. Given that acoustic pings are sent at a rate of approximately four per second (4.17 Hz), this implies that 1% of the time, the Maxelbots ran for only 0.25 seconds on old data. Only once were two consecutive pings in a row not received, yielding one 0.5 second gap in readings.

Of all of our tests, the factor most important to success was the temperature. Below roughly 6°C the electronics failed. Given that our components are not ruggedized, this is not surprising.

8 SUMMARY

This paper describes a 2D trilateration framework for the fast, accurate localization of neighboring robots. The framework uses three acoustic transducers and one RF transceiver. Our framework is designed to be modular, so that it can be used on different robotic platforms, and is not restricted to any particular class of control algorithms. Although we do not rely on GPS, stationary beacons, or environmental landmarks, their use is not precluded. Our framework is fully distributed, inexpensive, and scalable.

To illustrate the general utility of our framework, we demonstrated the application of our new robots in outdoor situations. The results from these experiments highlight the accuracy of our trilateration framework, as well as its current limitations (range and environmental temperature). For all of the Maxelbots, their X and Y positions are within roughly 11% of the desired values.

Open Source Project URL

The open source URL http://www.cs.uwyo.edu/∼wspears/maxelbot provides schematic details and videos of this project. We thank the Joint Ground Robotics Enterprise for funding portions of this work.

REFERENCES

Balch, T., Hybinette, M.: Social potentials for scalable multirobot formations. In: IEEE Transactions on Robotics and Automation. Volume 1. (2000) 73–80

Fax, J., Murray, R.: Information flow and cooperative control of vehicle formations. IEEE Transactions on Automatic Control 49 (2004) 1465–1476

Fierro, R., Song, P., Das, A., Kumar, V.: Cooperative control of robot formations. In Murphey, R., Pardalos, P., eds.: Cooperative Control and Optimization. Volume 66., Hingham, MA, Kluwer Academic Press (2002) 73–93

Brogan, D., Hodgins, J.: Group behaviors for systems with significant dynamics. Autonomous Robots 4 (1997) 137–153

Table 4: Accuracy of the followers' X and Y positions in a both formations (results in cm).

	Triangle			Diamond		
	Ideal	Mean	Std. dev.	Ideal	Mean	Std. dev
Follower1-X	-48	-53.0	3.3	61	62.2	1.8
Follower1-Y	91	97.8	2.8	61	54.9	2.8
Follower2-X	53	57.2	4.8	-61	-64.3	4.3
Follower2-Y	91	96.0	3.6	61	54.7	3.6
Follower3-X	-	-	-	0	1.0	5.1
Follower3-Y	-	-	-	122	111.8	4.6

Spears, W., Spears, D., Hamann, J., Heil, R.: Distributed, physics-based control of swarms of vehicles. Autonomous Robots 17(2-3) (2004)

Borenstein, J., Everett, H., Feng, L.: Where am I? Sensors and Methods for Mobile Robot Positioning. University of Michigan (1996)

Thomas, F., Ros, L.: Revisiting trilateration for robot localization. IEEE Transactions on Robotics 21(1) (2005) 93–101

Peasgood, M., Clark, C., McPhee, J. Localization of multiple robots with simple sensors. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05). (2005) 671–676

MacArthur, D.: Design and implementation of an ultrasonic position system for multiple vehicle control. Master's thesis, University of Florida (2003)

Nissanka, B., P.: The cricket indoor location system. Doctoral thesis, Massachusetts Institute of Technology, Cambridge, MA (2005)

Navarro-Serment, L., Paredis, C., Khosla, P.: A beacon system for the localization of distributed robotic teams. In: International Conference on Field and Service Robots, Pittsburgh, PA (1999) 232–237

Rothermich, J., Ecemis, I., Gaudiano, P.: Distributed localization and mapping with a robotic swarm. In Şahin, E., Spears, W., eds.: Swarm Robotics, Springer-Ve rlag (2004) 59–71

Payton, D., Estkowski, R., Howard, M.: Pheromone robotics and the logic of virtual pheromones. In Şahin, E., Spears, W., eds.: Swarm Robotics, Springer-Verlag (2004) 46–58

Spears, W., Hamann, J., Maxim, P., Kunkel, T., Heil, R., Zarzhitsky, D., Spears, D., Karlsson, C. Where are you? In Şahin, E., Spears, W., eds.: Swarm Robotics, Springer-Verlag (2006)

Heil, R.: A trilaterative localization system for small mobile robots in swarms. Master's thesis, University of Wyoming, Laramie, WY (2004)

Zarzhitsky, D., Spears, D., Spears, W.: Distributed robotics approach to chemical plume tracing. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05). (2005) 4034–4039

Hettiarachchi, S.: Distributed evolution for swarm robotics. Ph.D. thesis, University of Wyoming, Laramie, WY (2007)

POTENTIAL FIELD BASED INTEGRATED EXPLORATION FOR MULTI-ROBOT TEAMS

Miguel Juliá, Arturo Gil, Luis Payá and Óscar Reinoso
Miguel Hernández University, System Engineering Department
Avda. Universidad s/n. Edif. Torreblanca, 03202 Elche-Alicante, Spain
{*mjulia, arturo.gil, lpaya, o.reinoso*}*@umh.es*

Keywords: Integrated Exploration, SPLAM, Potential Fields, Cooperative Mobile Robotics.

Abstract: In this paper we present an approach for multi-robot cooperative exploration based on the potential field generated by several basic behaviours. When an unknown environment is explored the uncertainty in the localization normally grows, this fact may cause the failure of the Simultaneous Localization and Mapping (SLAM) algorithm, and thus constructing an useless and inaccurate map. The exploration algorithm described here considers the current knowledge of the environment, the location of the robots and the uncertainty in their positions in order to return to previously explored areas when it is needed. These actions definitely help the SLAM algorithm to build a precise map. Several simulations are presented that demonstrate the validity of the approach.

1 INTRODUCTION

In the last years, a large number of applications have emerged that require the utilization of cooperative mobile robots. Most of these applications require the robot team to be able to explore unknown environments autonomously. Employing multiple robots instead of a single robot in exploration is an advantage because the exploration time could be reduced (Cao et al., 1995).

Simultaneous Localization and Mapping (SLAM) techniques are generally used to explore an unknown environment. They allow to build a map that describes the environment while simultaneously using that map to localize the robots. However, the result obtained by the SLAM algorithm strongly depends on the trajectories performed by the robots (Stachniss et al., 2005). When the robots travel through unknown environments, the uncertainty over their position increases and the construction of the map becomes difficult. Returning to previously explored areas or closing loops reduces the uncertainty over the pose of the robots and improves the SLAM process.

Typical exploration algorithms do not take localization uncertainty into account and direct the exploration in order to minimize the distance traveled while maximizing the information gained. However, the solution presented here explores the environment efficiently and also considers the requisites of the SLAM algorithm. Our algorithm considers returning to previously explored places when the uncertainty becomes too large. This idea has been previously exploited by other authors and is commonly denoted as *Integrated Exploration* or SPLAM (Simultaneous Planning Localization And Mapping). A solution to the SPLAM problem enables a mobile robot to acquire data from sensors by autonomously moving through its environment while at the same time building a map. The main contribution of this paper is a new technique for *Integrated Exploration* for multi-robot teams.

The remainder of the paper is structured as follows. Section 2 discusses related work and Section 3 presents the behaviour based exploration algorithm. In Section 4 we explain the active localization state. Next, Section 5 presents simulation results to test the functionality of the method proposed. Finally, the main conclusions and future work are presented.

2 RELATED WORK

Exploration techniques work basically using the frontier concept introduced by (Yamauchi, 1997). He divided the map into a regular grid of cells where to represent the occupation probability. At the beginning of the exploration all the cells are unknown, so they are

initialized with an occupation probability of 0.5. This value is updated with the information of the sensors of the robots during the exploration. Relying on the occupation probability for each cell, the cells are labeled as free, occupied or unknown. Frontier cells are free cells that lie next to an unknown cell.

A group of exploration methods employ path planning techniques in order to direct the robots to the frontier cells (Simmons et al., 2000; Burgard et al., 2005; Zlot et al., 2002). They differ in the coordination strategies used to assign a frontier to each robot: the robots can go to the nearest frontier (Yamauchi, 1997) or they can follow a cost-utility model to make their assignments. Normally, the cost is the length of the path to a frontier cell, whereas utility could be understood in different ways: (Simmons et al., 2000) consider the utility as the expected visible area behind the frontier. (Burgard et al., 2005) consider in the utility function the proximity of frontiers assigned to other robots. (Zlot et al., 2002) suggest using a market economy where the robots negotiate their assignments.

Another group of exploration techniques makes use of potential field methods (Arkin and Diaz, 2002). Potential field based exploring methods take into account a set of behaviours to generate a resultant potential field. The most common behaviours in exploration are attraction to frontiers and repulsion from obstacles and other robots. This leads to the avoidance of other robots and collisions and also improves the exploration by dispersing the robots. As stated by many authors, the main drawback of this technique is the occurrence of local minima in the potential field, which may trap the robot and block the exploration process. A common solution to this problem is to plan a path to a frontier cell in order to get the robot out from the local minimum (Lau, 2003).

A few authors used integrated exploration in the last years (Feder et al., 1999; Bourgoult et al., 2002; Makarenko et al., 2002; Sim et al., 2004; Stachniss et al., 2005). (Feder et al., 1999) decide the next movement for robots by optimizing the information gain of the environment and minimizing the uncertainty in the localization of the robot. (Bourgoult et al., 2002) and (Makarenko et al., 2002) use a similar idea including the uncertainty in the localization as part of the utility function in the assignment of destinations to robots. These 3 techniques are based on the estimation of landmarks and they try to prevent that the uncertainty in the pose of the robots grows, by means of keeping always well estimated landmarks in the field of view. (Sim et al., 2004) recover the certainty over the pose of the robots during the exploration using a parametric curve trajectory and including returning to explored zones when the uncertainty in the pose of the robot is too high. (Stachniss et al., 2005) reduce the uncertainty by actively closing loops with previously explored areas. They create a topological map of the environment and look for opportunities for closing loops in it. As we can see, there are two main approaches to the problem of localization during the exploration: to take the uncertainty in the pose of the robots into account when choosing the movements for the robots or to explore and return later to previously explored zones when the uncertainty is large.

In this paper, a potential field based SPLAM technique is described. It is based on the potential field generated by several basic behaviours designed to rapidly explore the environment. It also considers returning to previously explored zones when needed.

3 BEHAVIOUR-BASED EXPLORATION ALGORITHM

In typical environments we can find a set of highly distinctive elements that can be easily extracted with the sensors of a robot. These elements are typically called landmarks. In our application, we assume that the robots are able to detect a set of distinctive 3D visual landmarks and they are able to obtain relative measurements to them using stereo cameras. These landmarks can be extracted as interest points found in the images of the environment (Mozos et al., 2007). The robot team is able to build a map with a vision-based technique consisting on a particle filter approach to the SLAM problem, known as FastSLAM (Gil et al., 2007).

Landmark based maps do not represent the free or occupied areas in the environment. This is the reason why we make use of a grid map to represent free and occupied cells detected using the information of the sonar. In addition, all the cells have a numerical value associated that indicates their degree of exploration, which is increased each time it falls into the field of view of the robot, until it reaches a limit value when the cell is considered to be fully explored. A cell with an exploration degree of zero is considered unexplored. We define the frontier cells as explored cells that lie next to an unexplored cell that do not belong to an obstacle.

Our approach to the problem of multi-robot exploration consists of five basic behaviours whose composition results in the trajectory of each robot in the environment:

Go to unexplored Areas: Each cell attracts each robot with a force that depends on the degree of ex-

Table 1: Forces defined for each behavior.

Go to unexplored areas:

$$\vec{F}_k^1 = \frac{1}{M}\sum_{i=1}^{M}\frac{\nu - e_i}{\nu}\frac{\vec{s}_i - \vec{p}_k}{r_{i,k}^3}$$

Go to frontier:

$$\vec{F}_k^2 = \frac{1}{M_F}\sum_{i=1}^{M_F}\frac{\vec{s}_i - \vec{p}_k}{r_{i,k}^3}$$

Avoid other robots:

$$\vec{F}_k^3 = \frac{1}{X}\sum_{j=1}^{X} -\frac{\vec{p}_j - \vec{p}_k}{r_{j,k}^3}$$

Avoid obstacle:

$$\vec{F}_k^4 = \frac{1}{M_O}\sum_{i=1}^{M_O} -\frac{\vec{s}_i - \vec{p}_k}{r_{i,k}^3}$$

Improve imprecise landmarks:

$$\vec{F}_k^5 = \frac{1}{n_t}\sum_{l=1}^{n_t}\sigma^l\frac{\vec{q}_l - \vec{p}_k}{r_{l,k}^3}$$

M:	Number of cells in the map.
M_F:	Number of frontier cells.
M_O:	Number of obstacle cells in the range.
X:	Number of robots.
n_t:	Current number of imprecise landmarks.
e_i:	Exploration level of cell i.
ν:	Maximum exploration level.
σ^l:	Landmark position measure uncertainty.
$\vec{s}_i$:	Position vector of the i-th cell.
$\vec{p}_j$:	Position vector of the j-th robot.
$\vec{p}_k$:	Position vector of the k-th robot.
$\vec{q}_l$:	Position vector of the l-th landmark
$r_{i,k}$:	Distance from i-th cell to robot k.
$r_{j,k}$:	Distance from robot j-th to robot k.
$r_{l,k}$:	Distance from l-th landmark to robot k.

ploration of the cell.

Go to Frontier: This behaviour attracts the robots to frontier cells since these are the cells that give way to areas of interest.

Avoid other Robots: This behaviour results in a repulsive force between robots that normally allows to spread the robots around the environment.

Avoid Obstacle: Each cell within a specific range that is identified as belonging to an obstacle, applies a repulsive effort over every robot. This range allows to easily adjust the system.

Improve imprecise Landmarks: This behaviour tries to improve the quality of the exploration of those areas where some landmarks have been extracted but whose accuracy is not high enough.

Table 1 shows how the forces are calculated for each behaviour. This way, the resulting force of the combination of those five behaviours on each robot constitutes a vector that indicates the trajectory of the robot to optimize the exploration process as follows:

$$\vec{F}_k^A = k_1\vec{F}_k^1 + k_2\vec{F}_k^2 + k_3\vec{F}_k^3 + k_4\vec{F}_k^4 + k_5\vec{F}_k^5 \,. \qquad (1)$$

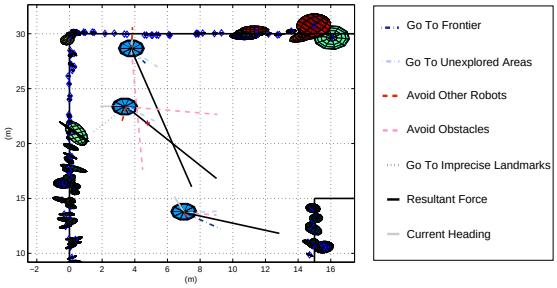


Figure 1: Weighted outputs of the behaviours and resultant force in an exploring situation. Also, the landmarks that have been detected until that moment are shown.

The composition of the behaviours is carried out taking into account a set of weights k_i whose value is deduced experimentally. Fig. 1 shows the bird's eye view of an exploring situation with three robots.

Potential field methods have a main disadvantage: when exploring complex environments, a robot may be trapped at local minima in the potential field and may not move, thus stopping the exploration process. To solve this problem, we assume that we are able to detect the situation in which the robot is trapped at a local minimum. In this case, a new state is triggered that enables the robot to escape from the local minimum by planning a path to the nearest frontier cell.

4 INTEGRATED EXPLORATION

As an unknown environment is explored the uncertainty in the localization of the robot grows. When the uncertainty over the pose of the robots is high, it is difficult to generate a correct map, and thus the exploration process is inefficient. If the error in the localization is very high, some frontiers and obstacles could be added to the grid map erroneously and some zones could remain unexplored. The perceptions of the robots in a given moment can be in conflict with past perceptions or with perceptions of other robots because of a deficient localization.

Figure 2 shows an example of an extremely deficient exploration caused by a large error in the localization. It can be observed how the wrong location of some obstacles obstructs the corridor and part of the environment remain unexplored. The error in the map of landmarks created is considerably high. All the landmarks in the map should appear over the walls but they are situated erroneously. These are the reasons we introduce new techniques to improve the localization.

The SLAM method we use in our experiments is commonly known as FastSLAM (Gil et al., 2007). It

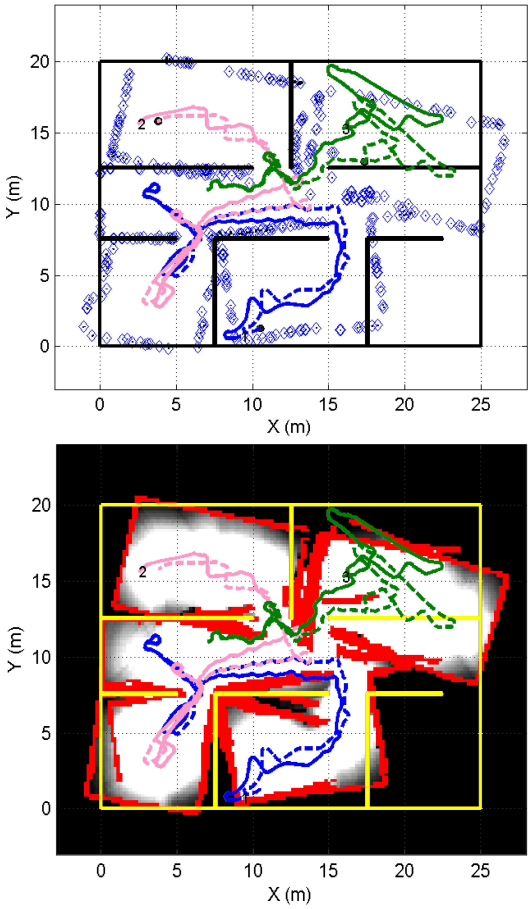


Figure 2: The upper figure shows the visual landmark map created in an exploration with deficient localization. The trajectories performed by the the robots are indicated in continuous lines and the estimated trajectories in discontinuous lines. The landmarks detected are marked in their estimated position. The bottom figure shows the grip map generated for this situation. The grade of exploration is indicated in gray levels and the obstacles detected in red color. Real obstacle positions are marked in yellow.

consists of a particle filter, each particle having an estimation for the path of each robot and an estimation of a set of landmarks conditioned to the path. We can measure the uncertainty on the localization of a robot by considering the dispersion for all the particles in the position of the robot. When travelling through unknown terrain, the dispersion of the particles usually increases since each is well localized in his own local map. Since we are using a finite number of particles to represent the pose of the robots, this representation gets worse when the uncertainty is too high. In this case we consider returning to previously explored areas to reduce this uncertainty. This idea has been employed by many authors (Feder et al., 1999; Bourgoult et al., 2002; Makarenko et al., 2002; Sim et al., 2004; Stachniss et al., 2005). Avoiding large periods of time with a high dispersion is a good technique to avoid the accumulation of error in the global localization and an accurate map can be obtained. Thus, our strategy is to return to positions with low dispersion when the dispersion in the pose of the robot grows. This solution produces a better estimation of the map and the robot's path.

We denote the model explained in Section 3 as the *Exploration State* (*StateA*). Besides, we introduce an *Active Localization State* (*StateB*). The *Exploration State* allows exploring new areas of the map meanwhile the robots are well localized. The *Active Localization State* intends to lead the robots to previously explored areas when they have a relatively high uncertainty associated, thus improving their localization. The transition between both states is made according to a hysteretic model with two transition thresholds that are compared with the dispersion in the pose of the robot.

In the *Active Localization State*, the control action of the robot is the composition of *Avoid Obstacle*, already presented, and a new behaviour *Go to Accurate Landmarks*. This new behavior aims at localizing the robot returning to previously explored landmarks.

Go to Accurate Landmarks: This behaviour tries to improve the estimation of the position of the robot, driving it to landmarks whose position has a robust estimation. Given a landmark, its position is calculated for each particle and a measure of its dispersion ε^l is calculated using the correspondent landmarks for the different particles. The correspondence is done considering an unique visual descriptor for each landmark. Each accurate landmark attracts the robot with a force inversely proportional to the distance:

$$\vec{F}_k^6 = \frac{1}{n}\sum_{l=1}^{n}\frac{1}{\varepsilon^l}\frac{\vec{q}_l - \vec{p}_k}{r_{l,k}^2}. \qquad (2)$$

being n the current number of landmarks in the map, $\vec{q}_l$ is the position of the l-th landmark, $\vec{p}_k$ is the position of the robot k and $r_{l,k}$ is the euclidean distance between both positions . Then the trajectory to follow is pointed by the vector:

$$\vec{F}_k^B = k_4\vec{F}_k^4 + k_6\vec{F}_k^6\ , \qquad (3)$$

where the weights are deduced experimentally.

As stated before, the local minima in the potential field can block the exploration process. In these cases, we plan a path to the nearest frontier cell. This solution directs the robots to unknown areas and thus is only a good solution in the *State A*. In *StateB*, local minima are also likely to appear. In this case, we plan a path to the last past position in the trajectory of the robot where the dispersion is low.

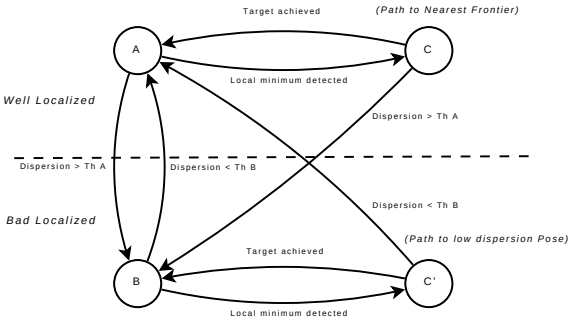


Figure 3: State transition diagram.

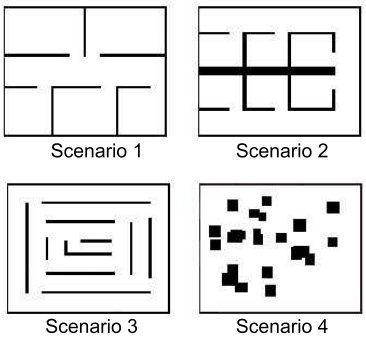


Figure 4: Scenarios.

Figure 3 shows the state diagram for a robot. We can distinguish two zones of operation: when the robot is well localized and when it is not. When it is well localized, it explores the environment by following the *State A* combination of behaviors. If it finds a local minimum during the exploration it plans a path (*State C*) to the nearest frontier cell. When it arrives to this cell it returns to the *State A*. If the dispersion on the robot position in the particle filter is over a given threshold the robot is considered to be bad localized and it switches to the *Active Localization State* (B). If it finds a local minimum being in this state, it plans a path (C') to a past position in the trajectory with low dispersion. When the dispersion decreases below a threshold the robot returns to the *Exploration State* (A).

5 EXPERIMENTS AND RESULTS

In this section, we analyze simulation results of the method proposed in this paper. The proposal is tested in presence of uncertainty in the robots localization to show the improvement in the quality of the maps generated and in the estimated path using the techniques proposed to return to previously explored areas.

The scenarios chosen to test the method are shown in Figure 4. Scenarios that represent hypothetical real places like *Scenario 1* or *Scenario 2* were chosen at the same time that other artificial scenarios as for example *Scenario 3* or a completely random scene as *Scenario 4*.

The method proposed is tested with and without considering the uncertainty in the position of the robots. Besides, it is compared with a pure path planning approach where the robots always plan a path to the nearest frontier cell. The mean error per robot in the estimated trajectories, the exploration time, as well as the error in the map of landmarks are analyzed.

The results of the simulation are shown in Figure 5. On *Scenario 1* and *Scenario 4*, the error on the estimated path and in the map is smaller with the proposed integrated exploration approach than when not returning to previously explored zones. This two scenarios have large free spaces. When the robots travel in these zones the measurement of the landmarks is difficult as they are far away. This makes the difference between including the *Go To Accurate Landmarks* behaviour or not. This large periods of time with bad localization increases the global localization error when not considering the dispersion to try to relocalize the robot.

For *Scenario 2* and *Scenario 3* we do not observe an improvement. Note that the global localization error depends on the form of trajectories and the exploration time that depend on the structure of the environments which is unknown. If the measurements are good enough this other factors affects in a random way and in average no difference is observed between the methods.

The exploration time always increases because this method does not always guide the robots to the direction of the maximal information gain as it looks also for the localization. As a conclusion, we think that a method that only tries to minimize the exploration time produces normally inaccurate maps useless for navigation. We consider that taking into account the requisites of SLAM while exploring the environment allows to obtain more precise maps.

6 CONCLUSIONS AND FUTURE WORK

In this paper a method for multi-robot cooperative exploration has been presented. The method is based on the computation of a set of behaviours designed so that we simultaneously consider the necessity of rapidly exploring the whole environment and the requisite to build an accurate map. In this sense, the

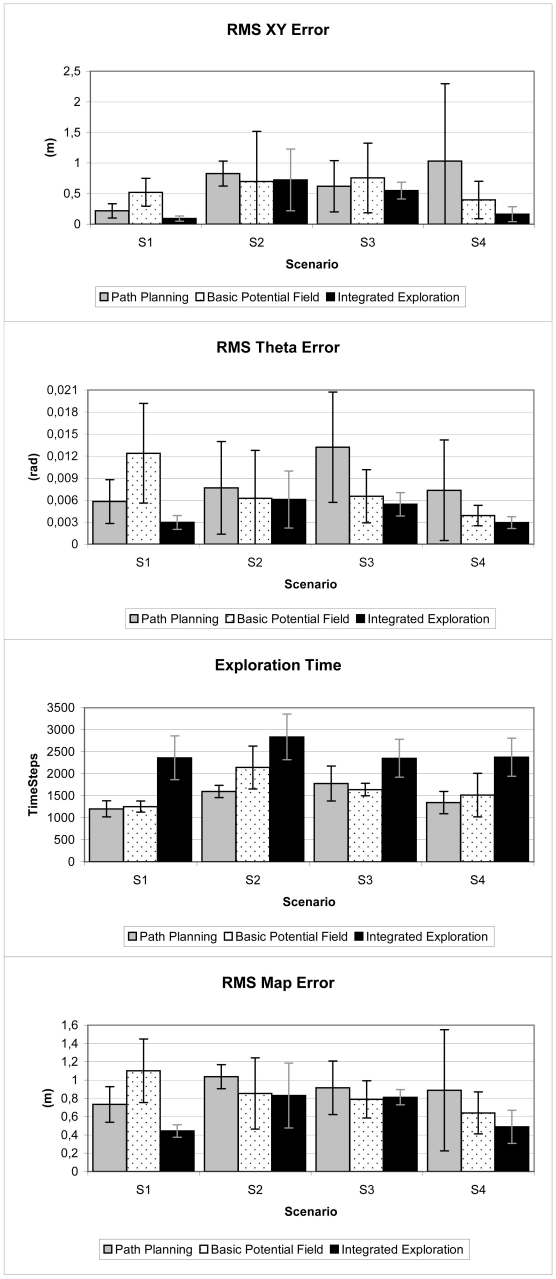


Figure 5: Results with uncertainty in localization: Path error on x,y and orientation (1st and 2nd graphs), exploration time (3rd graph) and map error (4th graph).

method directs the robots to return to previously explored places when the uncertainty on the location becomes significant, and this fact improves the quality of the resulting map. Several simulation results demonstrate the validity of the approach. On scenarios with large free space where there is a lack of good measures of the landmarks the accuracy of the map improves considerably.

As future works, we consider the extension of the approach in dynamic environments, adding techniques to learn automatically the multiple settings of the system. New behaviors that avoid the dispersion in the localization by trying to keep accurate landmarks in the field of view will be added. Furthermore, behaviors of attraction between robots will be incorporated in order to improve the localization since the observation of one robot by other member in the team may improve its localization. Semi-operated models that integrate the commands expressed by a human operator in the exploration task will also be studied, where these commands would be taken as an advice.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Government (Ministerio de Educación y Ciencia). Project: 'Sistemas de percepción visual móvil y cooperativo como soporte para la realización de tareas con redes de robots'. Ref.: DPI2007-61197.

REFERENCES

Arkin, R. and Diaz, J. (2002). Line-of-sight constrained exploration for reactive multiagent robotic teams. *7th International Workshop on Advanced Motion Control, AMC'02 , Maribor, Slovenia*.

Bourgoult, F., Makarenko, A., Williams, S., Grocholsky, B., and Durrant-Whyte, F. (2002). Information based adaptive robotic exploration. *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Lausanne, Switzerland.*

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics, Vol. 21 No3 pp 376-386, June*.

Cao, Y., Fukunaga, A. S., Kahng, A. B., and Meng, F. (1995). Cooperative mobile robotics: Antecedents and directions. *In IEEE/TSJ International Conference on Intelligent Robots and Systems, Yokohama, Japan*.

Feder, H., Leonard, J., and Smith, C. (1999). Adaptive mobile robot navigation and mapping. *Int. Journal of Robotics Research, 18(7)*.

Gil, A., Reinoso, O., Payá, L., and Ballesta, M. (2007). Influencia de los parámetros de un filtro de partículas en la solución al problema de slam. *Accepted for publication in the IEEE Latin America*.

Lau, H. (2003). Behavioural approach for multi-robot exploration. *Australasian Conference on Robotics and Automation (ACRA 2003), Brisbane, December*.

Makarenko, A., Williams, S., Bourgoult, F., and Durrant-Whyte, F. (2002). An experiment in integrated exploration. *In Proc. of the IEEE/RSJ Int. Conf. on Intelli-*

gent Robots and Systems (IROS), Lausanne, Switzerland.

Mozos, O. M., Gil, A., Ballesta, M., and Reinoso, O. (2007). Interest point detectors for visual slam. *Proc. of the Conference of the Spanish Association for Artificial Intelligence (CAEPIA), Salamanca, Spain, November*.

Sim, R., Dudek, G., and Roy, N. (2004). Online control policy optimization for minimizing map uncertainty during exploration. *In Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), New Orleans, LA, USA.*

Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., and Younes, H. (2000). Coordination for multi-robot exploration and mapping. *In Proceedings of the AAAI National Conference on Artificial Intelligence, Austin, TX*.

Stachniss, C., Haehnel, D., Burgard, W., and Grisetti, G. (2005). Actively closing loops in grid-based fastslam. information. *Advanced Robotics - The Int. Journal of the Robotics Society of Japan (RSJ), Volume 19, number 10, pages 1059-1080*.

Yamauchi, B. (1997). A frontier based approach for autonomous exploration. *IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, July 10-11*.

Zlot, R., Stentz, A., Dias, M. B., and Thayer, S. (2002). Multi-robot exploration controlled by a market economy. *Proceedings of the IEEE International Conference on Robotics and Automation.*

A GENERIC ARCHITECTURE FOR A COMPANION ROBOT*

Bas R. Steunebrink, Nieske L. Vergunst, Christian P. Mol, Frank P. M. Dignum
Mehdi Dastani and John-Jules Ch. Meyer
Intelligent Systems Group, Institute of Information and Computing Sciences
Utrecht University, The Netherlands
{*bass, nieske, christian, dignum, mehdi, jj*}*@cs.uu.nl*

Keywords: Social robot, architecture.

Abstract: Despite much research on companion robots and affective virtual characters, a comprehensive discussion on a generic architecture is lacking. We compile a list of possible requirements of a companion robot and propose a generic architecture based on this list. We explain this architecture to uncover issues that merit discussion. The architecture can be used as a framework for programming companion robots.

1 INTRODUCTION

Recently, research in companion robots and affective virtual characters has been increasing steadily. Companion robots are supposed to exhibit sociable behavior and perform several different kinds of tasks in cooperation with a human user. Typically, they should proactively assist users in everyday tasks and engage in intuitive, expressive, and affective interaction. Moreover, they usually have multiple sensors and actuators that allow for rich communication with the user. Of course, the task of designing and building a companion robot is highly complex.

Companion robots and affective virtual characters have already been built up to quite advanced stages. However, teams wishing to research companion robots often have to start from scratch on the software design part, because it is hard to distill a firm framework from the literature to use as a basis. Of course many figures representing architectures have been published, but it remains difficult to find out how existing companion robots really work internally. This may be due to most publications focusing on test results of the overall behaviors rather than on explaining their architectures in the level of detail required for replication.

The lack of emphasis on architectures may be caused by much of the research on companion robots being driven by the teams' research goals, resulting in their architectures mostly being designed to support just the desired behaviors instead of being generic for companion robots. If there were a good generic architecture for companion robots, a (simple) default implementation could be made, providing (new) researchers with a framework that they could use as a starting point. Depending on the application domain and research goals, some default implementations of modules constituting the architecture may be replaced to achieve the desired custom behavior, while other modules can just be readily used to complete the software of the companion robot.

Of course, anyone wishing to build the software of a companion robot can just start up his/her favorite programming environment and try to deal with problems when they occur, but obviously this is not a very good strategy to follow. Instead, designing and discussing an architecture beforehand raises interesting issues and allows questions to be asked that otherwise remain hidden. Indeed, there are many non-trivial choices that have to be made, pertaining to e.g. distribution and assignment of control among processes, synchronization of concurrent processes, which process is to convert what data into what form, where to store data in what form, which process has access to which stored data, which process/data influences which other process and how, the types of action abstractions that can be distinguished (e.g. strategic planning actions, dialogue actions, locomotion actions), the level of action abstraction used for reasoning, who converts abstract actions into control signals, how are conflicts in control signals resolved, what are the properties of a behavior emerging from

*This work supported by SenterNovem, Dutch Companion project grant nr: IS053013.

a chosen wiring of modules, what defines the character/personality of a companion robot (is it stored somewhere, can its parameters be tweaked, or does it emerge from the interactions between the modules?). Answers to these and many other questions may not be obvious when presented with a figure representing an architecture, but these issues can be made explicit by proposing and discussing one.

In this paper we introduce an architecture which is generic for companion robots and explain it in as much detail as possible in this limited space. This architecture contains the components necessary to produce reasonably social behavior given the multimodality of a companion robot's inputs and outputs. We do not claim that the proposed architecture represents the ultimate companion robot architecture. Rather, the aim of this paper is to provoke a discussion on the issues and choices involved in designing the software of a companion robot. Ultimately, this work could lead to the implementation of a generic framework that could be used as a basis for the software of new companion robots.

This paper is outlined as follows. In Section 2, we gather a list of possible requirements for a companion robot and introduce our architecture satisfying these requirements. In Section 3, we treat the *functional components* in the architecture in more detail. In Section 4, we connect the functional components by explaining the *interfaces* between them. We discuss some related work in Section 5 and finish with conclusions and plans for future research in Section 6.

2 POSSIBLE REQUIREMENTS FOR A COMPANION ROBOT

In order to come up with a generic architecture suitable for companion robots, we must first investigate the possible requirements for a companion robot. These requirements are optional, meaning that only the 'ultimate' companion robot would satisfy them all. In practice however, a companion robot does not have to. The actual requirements depend on various factors, such as the application area of the robot and its hardware configuration. However, below we compile a list, as exhaustive as possible, of possible requirements which a generic architecture must take into account.

First of all, a companion robot should be able to perceive the world around it, including auditory, visual, and tactile information. The multimodality of the input creates the need for synchronization (e.g., visual input and simultaneously occurring auditory input are very likely to be related), and any input inconsistent over different modalities should be resolved. Moreover, input processors can be driven by *expectations* from a reasoning system to focus the robot's attention to certain signals. Of course, any incoming data must be checked for relevancy and categorized if it is to be stored (e.g., to keep separate models of the environment, its users, and domain knowledge).

A companion robot should be able to communicate with the user in a reasonably social manner. This means not only producing sensible utterances, but also taking into account basic rules of communication (such as topic consistency). In order to maintain a robust interaction, a companion robot must always be able to keep the conversation going (except of course when the user indicates that he is done with the conversation). This also involves real-time aspects; e.g., to avoid confusing or boring the user, long silences should not occur in a conversation.

Additionally, a companion robot is likely to be designed for certain specific tasks, besides communicating with its users. Depending on e.g. the domain for which the companion robot is designed and the type of robot and the types of tasks involved, this may call for capabilities involving planning, physical actions such as moving around and manipulating objects, or electronic actions (e.g., performing a search on the internet or programming a DVD recorder). Proactiveness on part of the robot is often desirable in tasks involving cooperation.

A companion robot should also exhibit some low-level reactive behaviors that do not (have to) enter the reasoning loop, such as blinking and following the user's face, and fast reactive behaviors such as startling when subjected to a sudden loud noise. To make the interactions more natural and intuitive, a companion robot should also be able to form and exhibit emotions. These emotions can be caused by cognitive-level events, such as plans failing (disappointment), goal achievement (joy), and perceived emotions from the user (if negative: pity). Reactive emotions like startle or disgust can also influence a robot's emotional state. Moreover, emotions can manifest themselves in many different ways; e.g., facial expressions, speech prosody, selecting or abandoning certain plans, etc.

Finally, a companion robot should of course produce coherent and sensible output over all available modalities. Because different processes may produce output concurrently and because a companion robot typically has multiple output modalities, there should be a mechanism to synchronize, prioritize, and/or merge these output signals; e.g., speech should co-

incide with appropriate lip movements, which should overrule the current facial animation, but only the part that concerns the mouth of the robot (provided it has a mouth with lips).

In Figure 1, we present a generic architecture for companion robots which accounts for the requirements described above. Note that we abstract from specific robot details, making the architecture useful for different types of companion robots. We emphasize again that this is an architecture for an 'ultimate' companion robot; in practice, some modules can be left out or implemented empty.

3 FUNCTIONAL COMPONENTS CONSTITUTING THE ARCHITECTURE

In this section we describe the 'blocks' that constitute the proposed architecture. The interfaces ('arrows') between the components are explained in Section 4.

To begin with, the architecture is divided into eight *functional components* (i.e. the larger boxes encompassing the smaller blocks). Each functional component contains several *modules* that are functionally related. Modules drawn as straight boxes represent data storages, the rounded boxes represent processes, and the ovals represent sensors and actuators. Each process is allowed to run in a separate thread, or even on a different, dedicated machine.

No synchronization is forced between these processes by the architecture; they can simply send information to each other (see Section 4), delegating the task of making links between data coming in from different sources to the processes themselves. Below, each of the eight functional components is described, together with the modules they encompass.

Input Modalities. A companion robot typically has a rich arsenal of input modalities or sensors. These are grouped in the lower left corner of Figure 1, but only partially filled in. Of course, different kinds of companion robots can have different input modalities, of which a camera and a microphone are probably the most widely occurring. Other sensors may include touch, (infrared) proximity, accelerometer, etc.

Input Preprocessing. It is impractical for a reasoning engine to work directly with most raw input data, especially raw visual and auditory data. Therefore, several input preprocessing modules must exist in order to extract salient features from these raw inputs and convert these to a suitable data format. Some input modalities may even require multiple preprocessing modules; for example, one audio processing module may extract only speech from an audio signal and produce text, while another audio processing module may extract other kinds of sounds to create a level of 'sound awareness' for the companion robot.

Note that some of these input preprocessing modules may be readily available as off-the-shelf software (most notably, speech recognizers), so a generic architecture must provide a place for them to be plugged in.

Furthermore, there may be need for an input synchronizer that can make links between processed data from different modalities, in order to pass it as a single event to another module. The input synchronizer may initially be implemented empty; that is, it simply passes all processed data unchanged to connected modules. The input synchronizer can also be used to dispatch expectations that are formed by the action selection engines to the input preprocessing modules, which can use these expectations to facilitate feature recognition.

Low-level Behaviors. Low-level behaviors are autonomous processes that compete for control of actuators in an emergent way. Some behaviors may also influence each other and other modules. Examples of low-level behaviors include face tracking and gaze directing, blinking, breathing, and other 'idle' animations, homeostasis such as the need for interaction, sleep, and 'hunger' (low battery power), and reactive emotions such as startle and disgust.

Action Selection Engines. The 'heart' of the architecture is formed by the action selection engines. These are cognitive-level processes that select actions based on collections of data, goals, plans, events, rules, and heuristics. The outputs that they produce can generally not be directly executed by the actuators, but will have to be preprocessed first to appropriate control signals. Note that the interpreters of the action selection engines are depicted as layered to indicate that they can be multi-threaded.

The reasoning engine may be based on the BDI theory of beliefs, desires, and intentions (Bratman, 2002), deciding which actions to take based on percepts and its internal state. It should be noted that in terms of the BDI theory, the databases component plus the working memories of the action selection engines constitute the robot's beliefs.

An action selected by the reasoning engine may be sent to an output preprocessing module, but it can also consist of a request to initiate a dialogue. Because dialogues are generally complex and spread over a longer period of time, a dedicated action selection engine may be needed to successfully have a conversation.

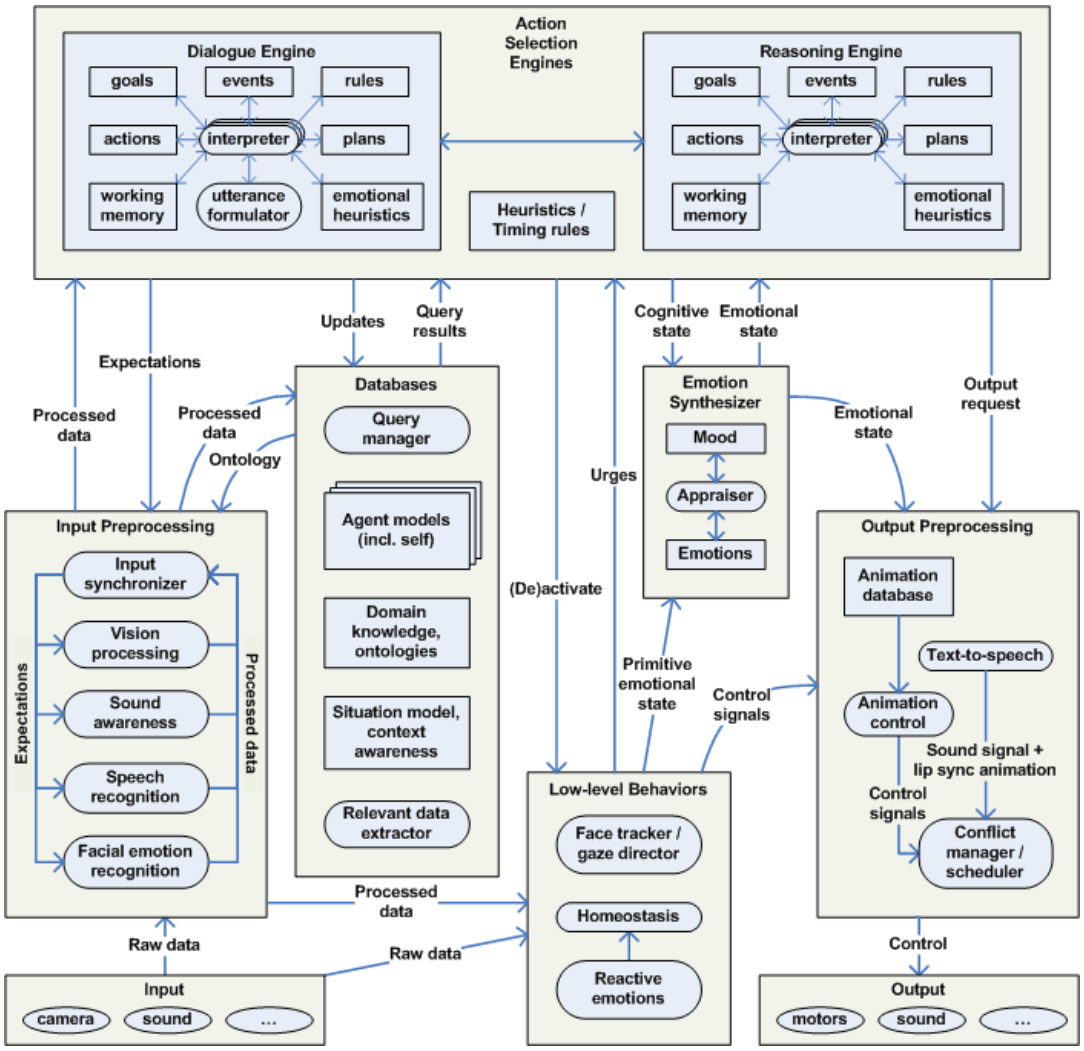


Figure 1: A generic architecture for a companion robot. The architecture takes into account the possible (or rather, probable) existence of multiple *input modalities*, multiple *input preprocessing* modules for each input modality, *databases* for filtering, storing, and querying relevant information, *action selection engines* for complex, goal-directed, long-term processes such as conversing, planning, and locomotion, an *emotion synthesizer* producing emotions that influence action selection and animations, multiple (reactive) *low-level behaviors* that can compete for output control, multiple *output preprocessing* modules including a conflict manager, and finally, multiple *output modalities*. Straight boxes stand for data storages, rounded boxes for processes, and ovals for sensors/actuators. The interfaces (arrows) between different modules indicate flow of data or control; the connections and contents are made more precise in the text. Note that only the 'ultimate' companion robot would fully implement all depicted modules; a typical companion robot implementation will probably leave out some modules or implement them empty, awaiting future work.

This dialogue engine contains an extra process called an utterance formulator; the task of this module is to convert an illocutionary act to fully annotated text, i.e. the exact text to utter together with information about speed, emphasis, tone, etc. This text can then be converted to audio output by the text-to-speech module (in the output preprocessing component).

A similar discussion about separating dialogues and (strategic) planning can be held for locomotion. In our research we have worked with stationary companion robots that focus on dialogues and facial animations. But there can of course be companion robots with advanced limbs and motions. For such robots there may be need for a third action selection engine, dedicated to motion planning. In the proposed architecture, there is room for additional dedicated engines in the functional component of action selection engines.

Finally, the architecture provides for a module called heuristics / timing rules. This is a collection

of heuristics for balancing control between the different action selection engines, as they are assumed to be autonomous processes. The different engines will get priorities in different cases; e.g., the plans of the dialogue engine will get top priority if a misunderstanding needs to be repaired. On the other hand, if the dialogue engine does not have any urgent issues, the reasoning engine will get control over the interaction in order to address its goals. Furthermore, it can verify whether the goals of the different action selection engines adhere to certain norms that apply to the companion robot in question, as well as provide new goals based on timing rules; e.g., to avoid long silences, the robot should always say something within a few seconds, even if the reasoning engine is still busy.

Databases. We have created a distinct functional component in the architecture where data is stored in different forms. This data includes domain knowledge, ontologies, situation models, and profiles of the robot itself and of other agents. The ontologies and domain knowledge are (possibly static) databases that are used by the input preprocessing modules to find data representations suitable to the action selection engines and databases. The agent profiles store information about other agents, such as the robot's interaction histories with these modeled agents, the common grounds between the robot and each modeled agent, and the presumed beliefs, goals, plans, and emotions of each modeled agent. These agent models also include one of the robot itself, which enables it to reason about its own emotions, goals, etc.

In order to provide a consistent interface to these different databases, a query manager must be in place to handle queries, originating from the action selection engines. A special situation arises when the robot queries its own agent model, for there already exist modules containing the goals, plans, and emotions of the robot itself. So the query manager should ensure that queries concerning these types of data can get their results directly from these modules.

Finally, a relevant data extractor takes care of interpreting incoming data in order to determine whether it can be stored in a more suitable format; e.g., if visual and auditory data from the input preprocessing component provides new (updated) information about the environment of the robot, it is interpreted by the relevant data extractor and stored in the situation model. Moreover, simple spatial-temporal reasoning may be performed by the relevant data extractor. If advanced spatial-temporal reasoning is needed for some companion robot, it may be better to delegate this task to a separate input preprocessing module.

Emotion Synthesizer. Typically, companion robots must show some level of affective behavior. This means responding appropriately to emotions of a (human) user, but also includes experiencing emotions itself in response to the current situation and its internal state. The emotions that concern this functional component are those of the companion robot itself and are at a cognitive level, i.e., at the level of the action selection engines. Examples of emotions are joy when a goal is achieved, disappointment when a plan fails, resentment when another agent (e.g. a human user) gains something at the expense of the robot, etc. More reactive emotions (e.g., startle) can be handled by a low-level behavior.

The emotion component consists of three parts. The appraiser is a process that triggers the creation of emotions based on the state of the action selection engines. The intensity of triggered emotions is influenced by the robot's mood (the representation of which may be as simple as a single number) and a database of previously triggered emotions. This database of emotions then influences the action selection engines (by way of their emotional heuristics module) and the animations of the robot, e.g., by showing a happy or sad face.

Output Preprocessing. Different processes may try to control the robot's actuators at the same time; obviously, this calls for conflict management and scheduling of control signals. Moreover, some modules may produce actions that cannot be directly executed, but instead these abstract actions need some preprocessing to convert them to the low-level control signals expected by the robot's actuators. E.g., the dialogue engine may want some sentence to be uttered by the robot, but this must first be converted from text to a sound signal before it can be sent to the loudspeaker. This functionality is provided by the text-to-speech module, which is also assumed to produce corresponding lip sync animations.

For companion robots with a relatively simple motor system, it suffices to have a single module for animation control which converts abstract animation commands to low-level control signals. This can be done with the help of an animation database containing sequences of animations that can be invoked by name and then readily played out. For companion robots with a complex motor system, the animation control module may be replaced by a motion engine (which is placed among the other action selection engines), as discussed above. In this case, an animation database may still fulfill an important role as a storage of small, commonly used sequences of motor commands.

Finally, actuator control requests may occur con-

currently and be in conflict with each other. It is the task of the conflict manager to provide the actuators with consistent control signals. This can be done by choosing between conflicting requests, scheduling concurrent requests, or merging them. These choices are made on a domain-dependent basis.

Output Modalities. All output modalities or actuators are grouped in the lower right corner of Figure 1. Similarly with the input modalities, these will be different for different kinds of companion robots, but a typical companion robot will probably have at least some motors (for e.g. facial expressions and locomotion) and a loudspeaker. Other actuators may include lights, radio, control of other electronic devices, etc.

4 INTERFACES BETWEEN FUNCTIONAL COMPONENTS

In this section, we explain the meaning of the interfaces between the functional components. For cosmetic reasons, the 'arrows' in Figure 1 appear to lead from one functional component to another, while they actually connect one or more specific modules *inside* a functional component to other modules inside another functional component. References to arrows in Figure 1 are marked in boldface.

Raw Data that is obtained by the input sensors is sent to the input preprocessing component for processing. Needless to say, data from each sensor is sent to the appropriate processing module; e.g., input from the camera is sent to the vision processing and facial emotion recognition modules, while input from the microphone is sent to the sound awareness and speech recognition modules. Any module inside the low-level behaviors component is also allowed to access all raw input data if it wants to perform its own feature extraction. In addition to raw data, low-level behaviors also have access to the **Processed data** from the modules inside the input preprocessing component. After the processed data is synchronized (or not) by the input synchronizer, it is sent to the action selection engines, where it is placed in the *events* modules inside the engines. The processed data is also sent to the databases, where the relevant data extractor will process and dispatch relevant data to each of the databases; e.g., context-relevant features are added to the situation model, while emotions, intentions and attention of a user that are recognized by the various input preprocessing modules are put in the appropriate agent model. Furthermore, the action selection engines can form **Expectations** about future events. These expectations are sent from the action selection engines back to the input synchronizer, which subsequently splits up the expectations and sends them to the appropriate input processing modules. They can then use these expectations to facilitate processing of input.

All processing modules in the input preprocessing component have access to **Ontology** information, which they might need to process raw data properly; e.g., the vision processing module might need ontological information about a perceived object in order to classify it as a particular item. This also ensures the use of consistent data formats. The processing modules can obtain this ontological information via the query manager in the databases component, which takes care of all queries to the databases. **Updates** to the databases can be performed by the action selection engines. The updates are processed by the relevant data extractor, which places the data in a suitable format in the appropriate database, in the same way as the processed data from the input preprocessing component. **Query results** can be requested by the action selection engines from the databases. The query manager processes the query and searches the proper database(s), guaranteeing a coherent interface to all databases.

(De)activate signals can be sent from the action selection engines to the low-level behaviors component. These signals allow the action selection engines some cognitive control over the robot's reactive behavior; e.g., if needed, the face tracker can be activated or deactivated, or in some special cases the reactive emotions can be turned off. **Urges** arising from the low-level behaviors can be made into goals for the action selection engines. For example, if the homeostasis module detects a low energy level, a goal to go to the nearest electricity socket can be added to the goals of the motion engine.

The action selection engines provide their **Cognitive state** to the emotion synthesizer. The cognitive state can be used by the appraiser to synthesize appropriate emotions. In addition to the cognitive state, a **Primitive emotional state** is also sent to the appraiser, where it can influence the intensity of cognitive-level emotions and the robot's mood. The current **Emotional state**, which is a compilation of the collection of triggered emotions, is sent to the action selection engines. The emotional heuristics inside the action selection engines then determine how the interpreter is influenced by these emotions. The animation control module inside the output preprocessing component also receives the emotional state of the agent, so that it can select a suitable facial expression from the animation database representing the current emotional state.

Output Requests are sent from the interpreters inside action selection engines to the output preprocessing component. Of course, the different kinds of output requests are sent to different modules inside the output preprocessing; e.g., (annotated) utterances from the dialogue engine's utterance formulator are sent to the text-to-speech module, while any actions from the action selection engines that involve motors are sent to the animation control module. The synchronization of all output signals is taken care of by the conflict manager, as explained in the previous section. Finally, **Control** signals are gathered and synchronized by the conflict manager inside the output preprocessing component and sent to the appropriate output modality.

5 DISCUSSION AND RELATED WORK

We do not claim that the presented architecture is perfect, and although we claim that it is generic for companion robots, it is probably not unique. Another team setting out to make a generic companion robot architecture will probably come up with a different figure. However, we expect the level of complexity of alternative architectures to resemble that of the one presented here, as it takes many components and processes to achieve reasonably social behavior. It should be noted that the intelligence of the system may not lie within the modules, but rather in the wiring (the "arrows"). The presentation of an architecture should therefore include a discussion on the particular choice of interfaces between modules. We draw confidence in our architecture from the fact that mappings can be found between this one and the architectures of existing companion robots and affective virtual characters, several of which we discuss next.

Breazeal (Breazeal, 2002) uses competing behaviors for the robot Kismet in order to achieve an emerging overall behavior that is sociable as a small child. Kismet has a number of different response types with activation levels that change according to Kismet's interaction with a user. In Kismet's architecture, the Behavior System and Motivation System can be mapped on our low-level behaviors; however, it lacks cognitive reasoning (obviously this was not necessary for its application), which is provided by our action selection engines. Other components in Kismet's architecture pertain to input and output processing, which map to corresponding preprocessing modules in our architecture.

Max, the "Multimodal Assembly eXpert" developed at the University of Bielefeld (Kopp, 2003), can also be mapped to our architecture. For example, it uses a reasoning engine that provides feedback to the input module to focus attention to certain input signals, which is similar to our *expectations*. It also has a lower-level Reactive Behavior layer that produces direct output without having to enter the reasoning process, and a Mediator that performs the same task as our conflict manager (i.e. synchronizing output). However, Max only has one (BDI) reasoning engine, where we have provided for two or more action selection engines.

6 CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have presented a generic architecture for a companion robot. We do not claim that it should be the foundation of the 'ultimate' companion robot; rather, we have presented this architecture in order to make many of the issues encountered when programming a companion robot explicit, so that these issues can be appropriately discussed.

The implementation of our companion robot architecture is yet to be finished, although it should be noted that it does not have to be fully implemented in all cases. Some of the functional components can be left out, simplified, or even extended (depending on the application) or programmed empty (awaiting future work), while for some modules, off-the-shelf or built-in software can be used. We are using the Philips iCat (Van Breemen, 2005) as a platform for developing a proof of concept of the proposed architecture. Ultimately, it can be used as a framework on top of which the software of new companion robots can be developed.

REFERENCES

Bratman, M., *Intention, Plans, and Practical Reasoning*, Harvard U. Press, 1987.

Breazeal, C.L., *Designing Sociable Robots*, MIT Press, 2002.

Van Breemen, A.J.N. *iCat: Experimenting with Animabotics*, AISB 2005 Creative Robotics Symposium, England, 2005.

Kopp, S., Jung, B., Lessmann, N., Wachsmuth, I. *Max – A Multimodal Assistant in Virtual Reality Construction*. In KI-Künstliche Intelligenz 4/03, p. 11-17, 2003.

CONCEPTS FOR AUTONOMOUS COMMAND AND CONTROL

Fernando Escobar and John McDonnell
Innovation center, Spawar sys ctr – San Diego, San Diego, CA 92152, U.S.A.
Fernando.escobar@navy.mil, John.mcdonnell@navy.mil

Keyword: Autonomous, Unmanned Systems, Self Organizing, MANET, Operational Orders, Multi-objective Optimization.

Abstract: The new Department of Defense (DOD) transformational doctrines for future battlefield operations emphasizes the need to more aggressively pursue program developments with unmanned systems technologies. Currently, there are ongoing Battle Experiments testing and assessing the operational performance of these technologies. These experiments in turn are uncovering current and future capability gaps that need to be fulfilled with aggressive research, engineering, test and evaluation. The Innovation Center at SPAWAR Systems Center, San Diego, has established a research and development process to better address Future Naval Capability gaps in the areas of both, Intelligent Autonomy and Autonomous Command and Control for Unmanned Systems. In this paper we report our research on two important components concepts for AC2: 1) Autonomous Resource Allocation, 2) Autonomy and Commanders Intent, and 3) A discussion on Self organizing C2.

1 INTRODUCTION

Sea Power 21 is a Naval vision that seeks to transform defense processes and modernize technologies for the battlefields of the future. The greatest challenges to transforming Naval doctrines from the industrial age to the information age has been the development of a clear notion of the value that distributed command architectures bring to modern combat Fig1. Distributed command architectures bring increased update speed of situational awareness. Each modernization step in C4ISR technology that enables faster horizontal integration is one step closer to a fully distributed command structure allowing for near real-time transmission of intent from the Commander on downwards resulting in better Situational Awareness of the Battlefield. Intention awareness is therefore an integral part of distributed command architecture and must be properly established in the information environment where faster and optimum execution of mission objectives is needed.

The fundamental infrastructure enabling command and control (C2) is undergoing a revolutionary change. The assumptions embedded in traditional C2 such as a centralized decision authority and well-defined hierarchy are being reassessed, especially in light of mission areas that involve coalition operations and the emergence (and dependence) on a ubiquitous IT capability (Alberts, 2007). While moving away from traditional C2 to a net-centric environment represents unique challenges, the prevalence of unmanned systems must also be considered within the context of emerging architectures and concepts. If properly architected, unmanned C2 systems should meld seamlessly into the operational environment augmenting and working in concert with C2 for manned units. Most investment in autonomy is being made at the platform level. This work focuses on the next level of autonomy- that is, the autonomous interaction of autonomous platforms to achieve pre-specified objectives.

The DoD Definition for *C2* is given (Joint Publication, 2002) as *the exercise of authority and direction by a properly designated commander over assigned and attached forces in the accomplishment of the mission.*

Autonomous is defined as *not controlled by others or by outside forces; independent and independent in mind or judgment; self-directed.*

Considering these definitions, Autonomous Command and Control (AC2) can be defined as *the independent, self-governed exercise of authority and direction over the assigned forces in the accomplishment of the mission*.

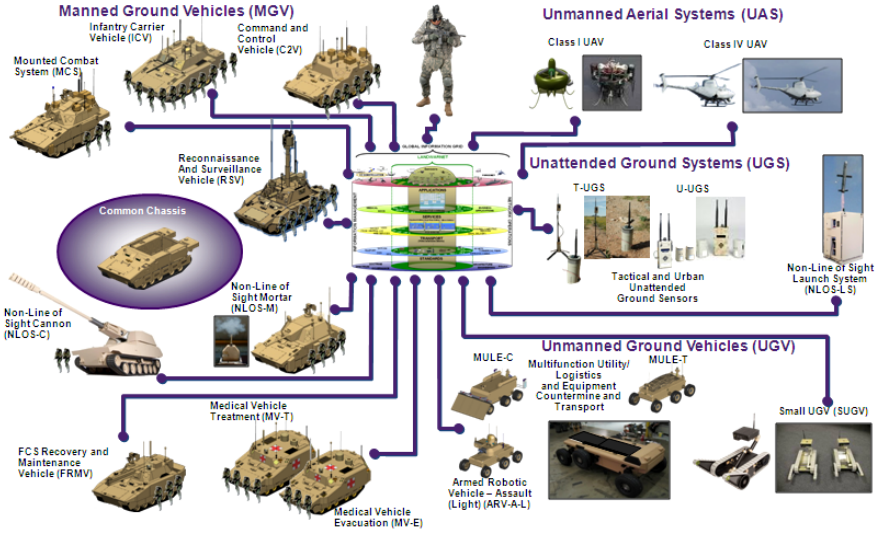


Figure 1: Unmanned Systems in DODs Transformational Information Architecture. (http://www.army.mil/fcs/).

Figure 2: Autonomous Command & Control (AC2) for Self Organizing Unmanned Systems.

The prevalence of unmanned systems has increased dramatically across the DoD services in recent engagements. In addition, user acceptance has become well established over this time ensuring that unmanned platforms will remain pervasive in future conflicts. Recently released Master Plans for both USVs (UUV, 2007) and UUVs (UUV, 2004) allude to the need for *autonomous group/cooperative behavior* to achieve the desired mission objectives for these types of systems. Fig. 2 illustrates the self organizing concepts of a disparate set of unmanned platforms.

The capabilities required to achieve AC2 include:

- Self-Organizing C2
- Translate Commander Intent to Executable Missions
- Autonomous Allocation and Management of Resources
- Machine Learning from Training/Experience
- Near Real-Time Analysis for predicting future C2 actions
- Seamless Interoperability of C4ISR Systems

- Sufficient BW and communications
- Autonomous Platforms and Sensing
- Level 3, 4 Fusion

The first three bullets are elaborated on in the following sections. While critical to achieving AC2, the remaining topics are advancing under a myriad of other efforts. For example, the seamless interoperability of C4ISR systems is being addressed under next-generation C2 efforts which are focused on providing a Service Oriented Architecture (SOA) to the warfighter. In addition to architecture, mobile add-hoc networks (MANET) are being studied to determine the best methodologies to achieve self-forming/self-healing networks and provide desired QoS levels. Bandwidth utilization will continue to improve with spectrum management, compressed sensing, along with novel routing and radio capabilities. Higher levels of sensor fusion are being rigorously investigated in order to ascertain enemy course-of-action analysis, turn data into understanding and wisdom, and autonomously improve sensor fusion capability. Autonomous sensing is also in the critical path as that dovetails with the allocation and management algorithms that are incumbent in AC2. Finally, significant investment continues in imbuing individual platforms with autonomy and analyzing the benefits of shared information/awareness.

2 SELF ORGANIZING C2

The key attributes of next-generation C2 include agility, focus, and convergence (Alberts, 2007). Agility is the ability of distributed platforms to self-synchronize and organize into an appropriate C2 topology in a dynamic manner. Self-synchronization will determine the decision rights across the platforms, and, in effect, serve as part of the cost function in the formation of the C2 topology. It is imperative that any self-organizing C2 topology yield deterministic behavior(s). The salient features that should be used to automatically determine an appropriate C2 topology remain to be discovered. Intuitively, the decision space could include the number of assets, the information capacity of the assets, the connectivity bandwidth between assets, and mission and environmental complexities. For purposes of discussion, C2 topologies are characterized in (Figure 3) as centralized, localized, and distributed. If, for example, a key component for determining C2 topology is the number of assets in the area of interest, then thresholds could be configured to trigger the formation a different topologies as exemplified in Figure 4. In addition to determining the salient factors, there is significant challenge is in determining the threshold functions.

A more effective approach may consider decomposing the problem such that these lower-level categories are mapped into the higher levels characterizations of information distribution, interaction patterns, and allocation of decision rights such as discussed by Alberts (Alberts, 2007). This hierarchical decomposition may serve to simplify the complexities involved in determining effecting AC2 topologies.

3 COMMANDER'S INTENT

The understanding of Commanders Intent (CI) clearly demonstrates that although the concept of intent has been in our doctrine for quite a while, confusion still exists and there is little empirical investigation into the process of communicating intent.

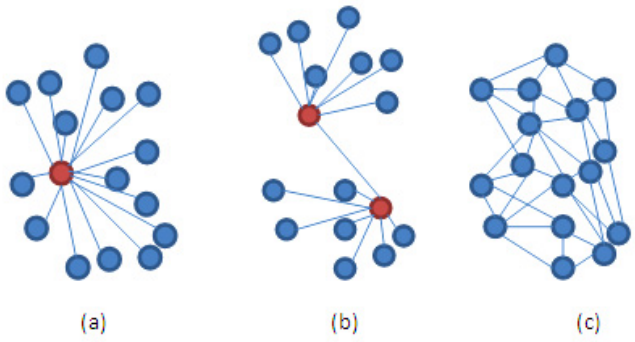


Figure 3: C2 Topologies: (a) Centralized, (b) Localized, and (c) Distributed.

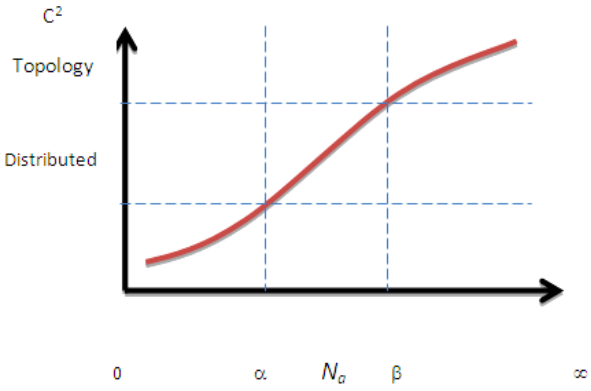


Figure 4: A notional mapping between the number of assets in the region of interest and the most effect C2 Topology.

CI has long been used to guide the actions of subordinates, but has only recently been formally included in doctrine. CI first appeared in US Army Field Manual in 1982 (GPO, 1982). During the 1970s, when the military tended to centralize decision making; however, failed hostage missions and similar events signaled the need to empower subordinate players on the scene. A model of today concept of CI can be traced to Army doctrine writers that used the German army's Aftragstaktik (Silva, 1989) first introduced in the early 19th century. The word means "mission-oriented" reflecting the developments in response to the French Revolution. This mission oriented methodology was the realization that battle is marked by confusion and ambiguity and that trust between superior and subordinate is the cornerstone of mission-oriented combat. Today, CI consists of a brief directive, usually in written format with a purpose, a method, and an endstate for any given operation. It is also the single unifying focus for all subordinates elements or groups of a command structure which are dedicated to different activities (communication, Intelligence, surveillance…) but which cooperate/collaborate to achieve mission effectiveness and success.

3.1 Concepts for Automating Commanders Intent (CI)

Automating Commanders Intent (CI) and military courses of action are very complex and difficult activities. These activities should take into consideration environmental information, predictions, the end state targeted and resource constraints. Automating Commanders Intent involves solving simultaneously planning and scheduling problems. In this section we provide 1) an approach to transforming CI objectives into an algebraic form, 2) a discussion on task scheduling, optimization, and resource allocation.

3.1.1 Algebraic Representations of CI

An approach to transforming CI into an algebraic form can best be described by the flow diagram Figure 5. As mentioned above, a CI consists of a brief directive containing objective statements. The first transformation (formalization) of these statements is done by utilizing a formal specification language such as the one provided by Berzin & Luqi (Berzins and Luqi, 1991). Formal statements of objectives and constraints are then stored permanently on a database. A Natural Language Processing (parsing) function aided by a Naval Lexicon provides formal unambiguous objective statements for encoding; the encoder creates an algebraic representation of these objectives creating what we call elementary actions. The elementary actions together with proper task scheduling algorithms, multi-objective optimization functions, and resource allocation methods provide a framework for automating Commanders Intent.

3.1.2 Task Scheduling & Optimization

We suggest a task (course of action) approach to automating Commanders Intent based on evolutionary algorithms that use multi-objective optimization methods and support resource constrained CI development with both cardinal and ordinal objectives.
During the development step, the commander analyses the relative combat power of friendly and enemy forces, and generates the CI.

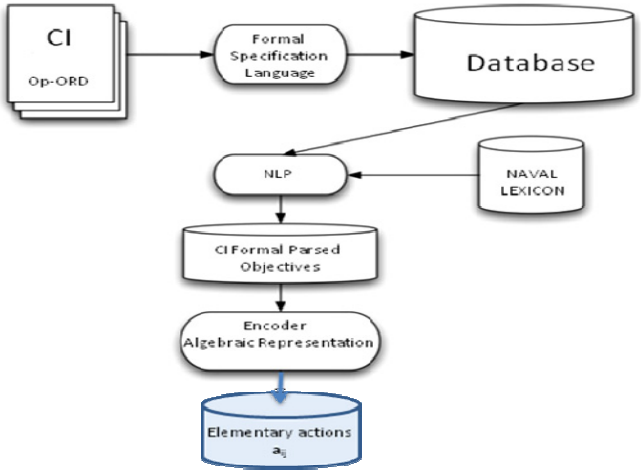


Figure 5: An approach for transforming CI into a formal algebraic representation.

During the mission analysis, the objectives are identified, assigned, and tasks (courses of action) are implemented to perform the mission. These tasks can be decomposed into sub-tasks. Tasks and sub-tasks can be represented by means of a hierarchical structure –a Graph. Synchronization analyses leads to identifying temporal and spatial relationships between elementary tasks. The automating algorithm must consider all available resources and capabilities and assign them to tasks. Synchronizing tasks then requires scheduling of all tasks according to resource availability, deployment constraints, and task relationships. We provide a task (courses of action) planning model as a multiple mode resource-constrained scheduling problem (MRCPS) since, from a methodological point of view planning and scheduling are not much different. Our model consists of representing generic activities (tasks with specific combinations of resources) into elementary (or primitive) actions interrelated to accomplish the mission objectives. This process implies the identification of the tasks (when and where), precedence relationships, the pool of available resources with their localization, and the objectives of the mission. An objective is then represented as an oriented time-space graph of tasks. Figure 6.

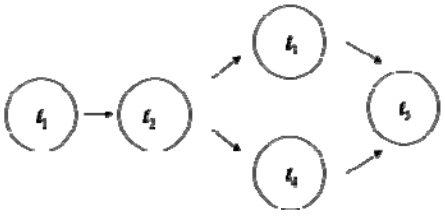


Figure 6: An Objective represented as a task.

Depending on the combination of resources allocated and the actions in the scheduler, different courses of action networks could be obtained, such as the one above. They constitute variants (or alternatives) of a mission with different evaluations on objectives. Solving CI and courses of action planning problems is NP-Hard. But a feasible process for automating CI with respect to multiple objectives for resource allocation may include evolutionary algorithms (EA) with meta-heuristic approaches or a method that addresses the multi-objective aspect of resource-constrained scheduling problems in which all objectives are combined into one single scalar value by using weighted aggregating functions. The search is then performed several times to find a compromise solution that reflects these preferences. Another approach is to generate the set of compromise solutions in a single execution of the optimization such as done by multiple-objective Evolutionary Algorithms. In this section, we provide a construct for the tasking and resource allocation associated with a CI that can be implemented using multiple-objective EAs. Evolutionary Algorithms are able to deal simultaneously with multiple solutions for solving multi-objective optimization problems allowing a set of potential Pareto optimal solutions to be found in the same iteration.

Here is our construct: Multi-objective CI can be characterized by a set of tasks, a set of resources, precedence relationships, resources, constraints and global performance functions F_z shown in Figure 7.

Once a CI has been decomposed into its requisite tasks, the question of which autonomous unmanned system should be responsible for executing each particular task still remains. Many techniques for

Optimize: $F_z, z = 1,...,Z$

s.t. $t \in D$

s.t. $R \in D$

Use vector of tasks $t = \{t_1, t_{2,...,} t_n\}$ having the following attributes for each task t_i :

Define starting and ending time [$\tau_b(i)$, $\tau_f(i)$]

Define earliest and latest starting and ending time [$\tau_s(i)$, $\tau_e(i)$]

Define type and quality of resources required, represented by a set $\boldsymbol{R}$ composed of renewable and nonrenewable resources available in limited quantities, i.e:

$R_k(t_i) = \{r_{1i}, r_{2i}, ..., r_{mi}\}$ is the k^{th} set of resources required to accomplish the task t_i .

Consider set of predecessors {PR} characterized by the tasks that temporally and /or spatially precedes t_i

Use resources $\boldsymbol{R}$ having the following attributes:

Define starting and ending time of availability $[t_{rs}(k), t_{re}(k)]$

Define localization of resources (*x,y,z*)

Define types of resources.

Define other specific characteristics such as “mean speed of (for mobile resources)”, reliability, etc.

Figure 7: A construct for multi-objective task optimization for low size problem (~10actions).

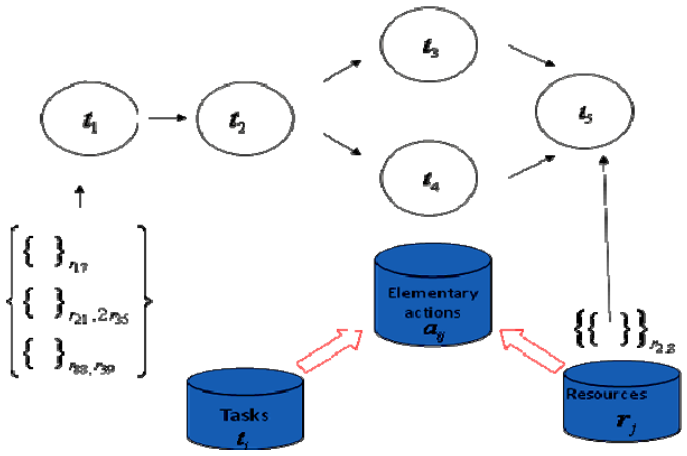


Figure 8: Summary of the main concepts needed for Commanders Intent Automation.

multi-robot task allocation are included in the works of Parker (Parker, 1998), LePape (Le Pape, 1990), and others (Botelho and Alami, 1999). Mataric (Gerkey and Mararic) provides a thorough review of several Multi-Robots Task Allocation Frameworks.

4 AUTONOMOUS RESOURCE ALLOCATION

Another key attribute of next-generation C2 is convergence (Alberts, 2007). Convergence is the ability for independent actors to achieve operational coherence in a deterministic manner. The emergence of platforms with multiple modalities (eg. sensing, SAR, strike, etc….) in the manned and unmanned arenas allows for additional flexibility in the allocation of resources at the added cost of an increasing complexity in the search space. The resource allocation problem for AC2 must be able to consider any platform for any task based upon the platform's capabilities. Optimizing across any modality (COMMS, strike, sensing, etc…) is an NP-hard problem. The AC2 resource allocation must consider all modalities simultaneously in assigning assets to objectives.

As stated above, the AC2 resource allocation problem is a combinatorial optimization problem that must consider the dynamic environment; a nonlinear, multi-modal objective function; nonlinear constraints; and binary decision variables. Algorithms which address resource allocation problems of this nature tend to be based on heuristic methods. The extreme team methods (Scerri et al., 2005) are effective in the presence of communications limitations where global decision support is not a viable option. Extreme teams have the following characteristic:

- Near real-time assignments
- Platforms may perform more than one task
- Inter-task constraints may be present

Extreme teams are largely based on distributed constraint optimization problems (DCOP) methods. These types of algorithms can be applied to either end of the C2 topology spectrum or can be used in a complementary fashion for a localized topology shown in Table 1.

Table 1: Recommended Resource Allocation Algorithms for C2 Topologies.

Distributed	Localized	Centralized
DCOP	DCOP+Heuristic	Heuristic

The AC2 resource allocation performance must be considered in light of scalability, satisficing behavior (GPO, 1982), robustness, and generality. It is important that the resource algorithm scale for large numbers of assets and mission objectives. If the solutions are near-optimal and generated in a reasonable timeframe, the performance can be considered to meet the satisficing criteria. In addition, the algorithm must be stable, converge rapidly, and insensitive to initial conditions. Finally, the algorithm must be able to accommodate the general nature of the objective described above.

The objective function under consideration by the optimization engine should consider the following components;

- Mission Effectiveness
- Mission Risk
- Mission Persistence
- Information Utility

The Mission Effectiveness considers all aspects sensing communications and weapons required to meet mission goals. The risk component considers items such as METOC enemy defenses, deconfliction and energy consumption. The Persistence parameter may be required to minimize global change in the solution set. For example, if a global optimizer is used, then the results could be dramatically varied at every solution step. Persistence will reduce this variability. Finally, the Information component is must be incorporated as a metric to ensure that the right data gets to the right place and platforms. For Autonomous C2 the ramifications of *automated* subtask generation should also be considered. Mission planners generate many subtasks to satisfy the overall mission objectives to achieve the desired effect(s). AC2 must also be able generate sub-goals in a parsimonious manner so that objectives can be accomplished and new constraints generated by these sub-goals are readily satisfied. The process of introducing sub-goals and their associated constraints introduces a complexity versus performance issue that should be bounded within the AC2 construct. This notion is analogous to Akaike's Information Criterion (AIC)

Figure 9: Notional depiction of an AC2 Turing test in a mixed manned/unmanned systems environment.

where the number of parameters and the log-likelihood of the error in the function being fitted are balanced.

5 CONCLUSIONS

Command and control *in the ether* represents a shift away from traditional C2 constructs. AC2 represents the ubiquitous nature of C2 in the distributed realm where emergent behaviors are manifested by large groups of platforms that are more complex than those emulating ants and birds in colony and flocking models, respectively. The potential collaborative behaviors that would emerge under different information management strategies should be addressed as part of an integrated investigation incorporating the C2 topology and resource allocation ideas described here.

While C2 of UxVs will be a driver in developing AC2, the evolutionary step of mixed manned and unmanned missions can be considered as an AC2 Turing Test. This notion is exemplified in Figure. 8 where the manned platforms under direction of the AC2 system do not know whether they are under direction of manned or unmanned systems.

REFERENCES

Alberts, D. S.: Agility, Focus, and Convergence: The Future of Command and Control. The International C2 Journal: Special Issue on The Future of C2, Vol. 1, No 1, (2007) 1-30.

Joint Publication 1-02: Department of Defense Dictionary of Military and Associated Terms, (Sep,2002).http://www.dtic.mil/doctrine/jel/doddict.

The Navy Unmanned Surface Vehicle (UUV) Master Plan. 23 July, 2007.

The Navy Unmanned Undersea Vehicle (UUV) Master Plan. Nov. 9, 2004.

US Army Field Manual (FM) 100-5, Operations (Washington, DC: Government Printing Office, (GPO) 1982.

J.L. Silva, "Aufragstaktik", Infantry, September-October 1989, 6-7.

Valdis Berzins., Lucia Luqi., Software enginnering with Abstractions, Addison-Wesley, Longman Publishing Co., Inc., Boston, MA, 1991

P. Scerri, et al. "Coordinating very large groups of wide area search munitions," Theory and Algorithms of Cooperative Systems, 2005.

L. Parker, "ALLIANCE: An Architecture for fault-tolerant multi-robot cooperation," IEEE Transactions of Robotics and Automation, vol. 14, no. 2, pp. 220-240, 1998.

C. Le Pape, "A combination of centralized and distributed methods for multi-agent planning and scheduling," in Proceedigns of ICRA, 1990, pp. 448-493.

S. Botelho and R. Alami, "M+: A Scheme for multi-robot cooperation through negotiated task allocation and achievement," in Proceedings of ICRA, 1999, pp. 1234-1239.

B. Gerkey and M. Mararic, "A formal analysis and taxonomy of task allocation in multi-robot systems," International Journal of Robotics Research, vol.23, no. 9, pp. 939

AUTHOR INDEX

AUTHOR INDEX (CONT.)

Proceedings of ICINCO 2008
Fifth International Conference on Informatics in Control, Automation and Robotics
ISBN: 978-989-8111-31-9
http://www.icinco.org