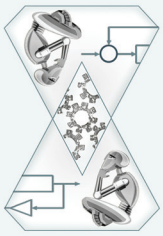


ICINCO 2008

FIFTH INTERNATIONAL CONFERENCE ON INFORMATICS IN CONTROL, AUTOMATION AND ROBOTICS

Proceedings

Intelligent Control Systems and Optimization

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 ICSO

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: 978-989-8111-30-2

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

SHORT PAPERS

POSTERS

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:

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

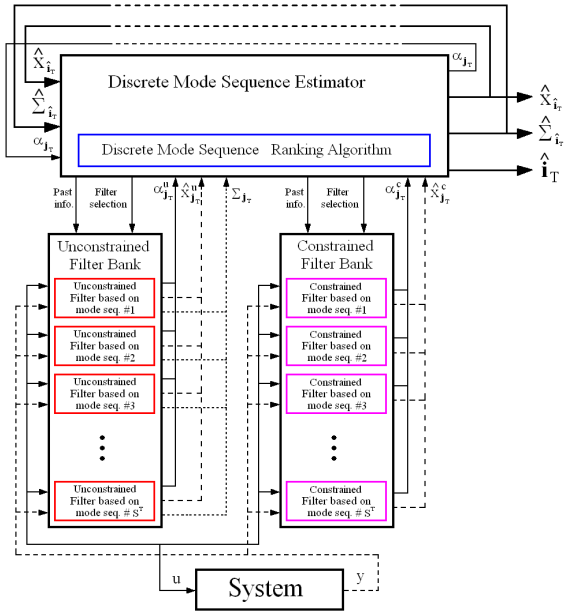


Figure 1: Interacting Multiple-Model Estimation Algorithm.

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.

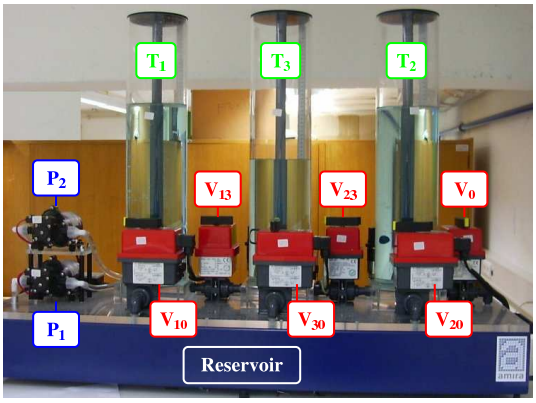


Figure 2: The three-tanks setup.

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.

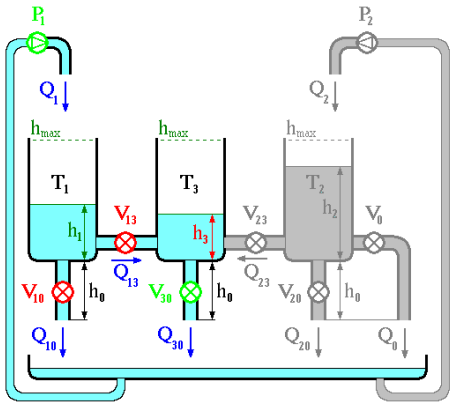


Figure 3: Final model of the three-tanks system.

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.

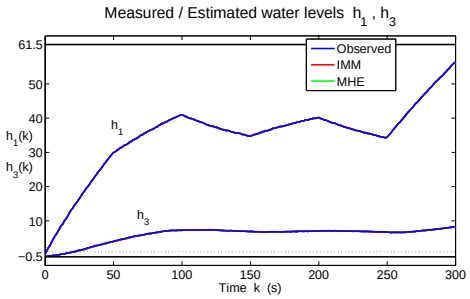


Figure 4: Water levels estimation using the IMM and MHE estimation algorithms.

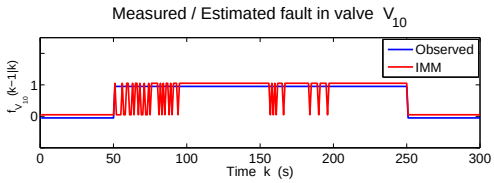


Figure 5: Estimation of the discrete mode sequence relative to the fault in valve $\mathbf{V_{10}}$.

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.

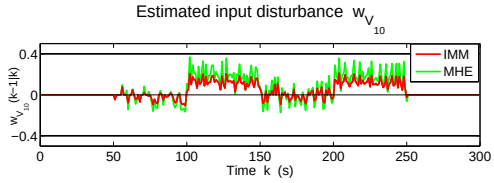


Figure 6: Estimation of the input disturbance $w_{V_{10}}(k-1|k)$ corresponding to the fault in valve $\mathbf{V_{10}}$.

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.

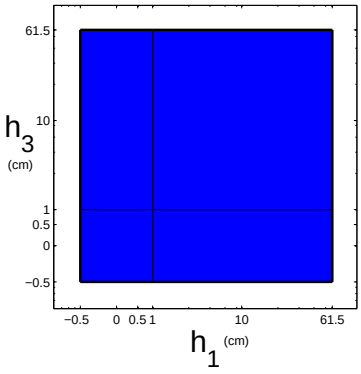


Figure 7: Map of probability of correct mode estimation, with 1s delay, when valve $\mathbf{V_{10}}$ is fully closed. (Red - probability of correct mode estimation 0, Blue - probability of correct mode estimation 1)

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

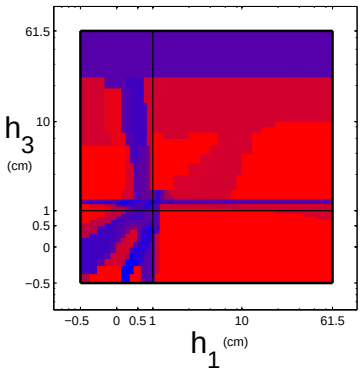


Figure 8: Map of probability of correct mode estimation, with 1s delay, when fault $f_{V_{10}}$ is active. (Red - probability of correct mode estimation 0, Blue - probability of correct mode estimation 1)

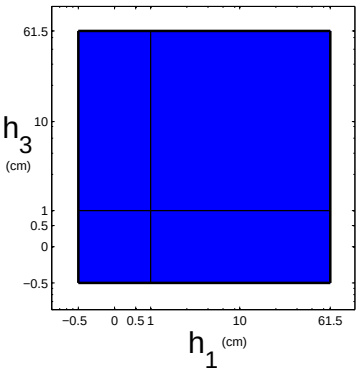


Figure 9: Map of probability of correct mode estimation, with 1s delay, considering that fault $f_{V_{10}}$ has maximum intensity, $w_{V_{10}} = 0.4$. (Red - probability of correct mode estimation 0, Blue - probability of correct mode estimation 1)

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.

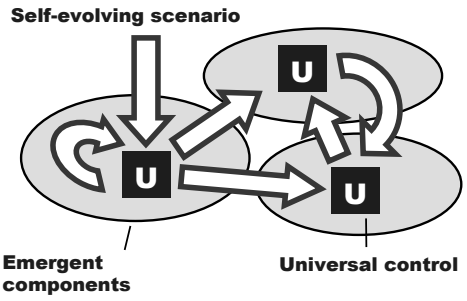


Figure 1: Runtime coverage of a distributed system.

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).

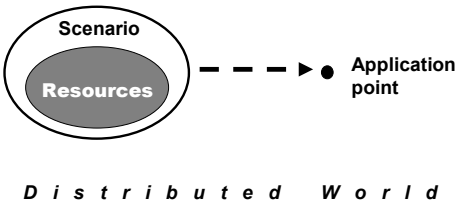


Figure 2: The initial state.

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.

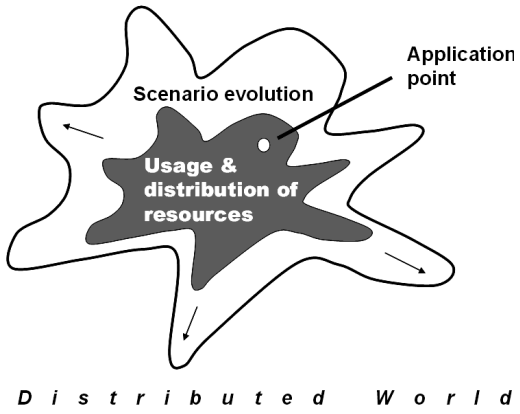


Figure3: Spreading operations and resources.

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:

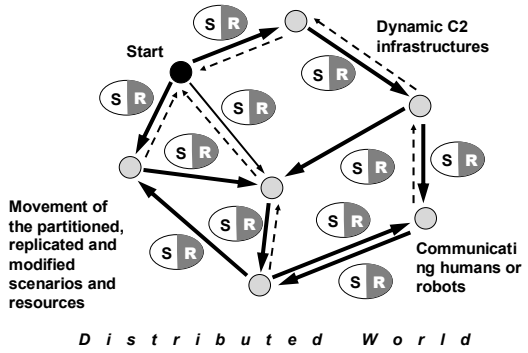


Figure 4: Resultant interactions between system parts.

- 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).

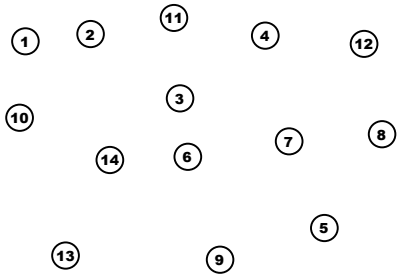


Figure 5: Distributed system nodes.

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.

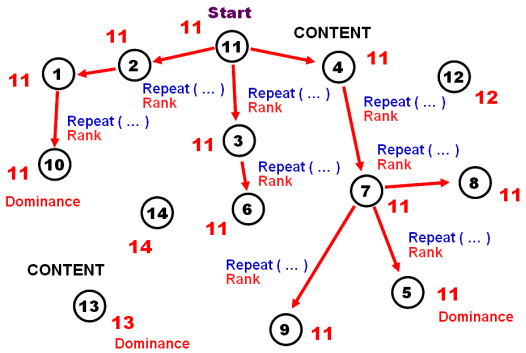


Figure 6: Resulting in partial dominance.

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.

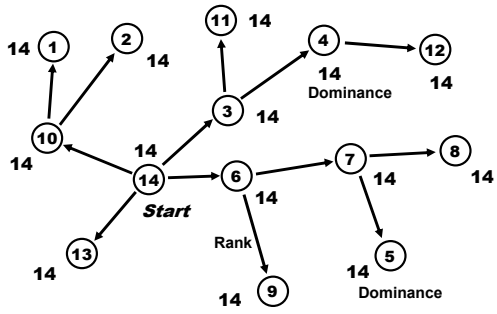


Figure 7: Setting absolute dominance of node 14.

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.

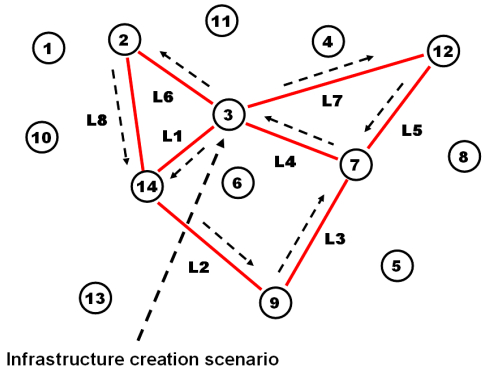


Figure 8: Creating a distributed infrastructure.

```
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.

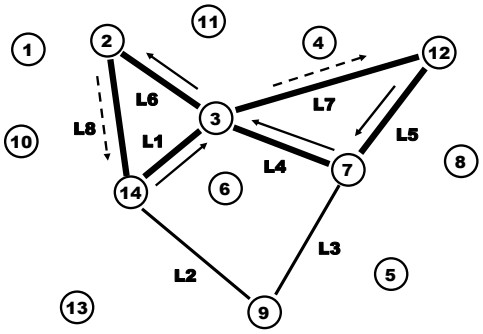


Figure 9: Finding all triangles in the infrastructure.

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").

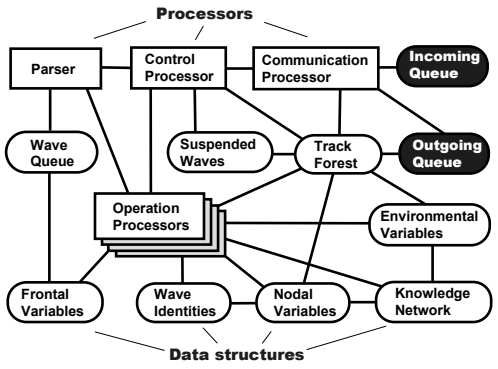


Figure 10: The WPL interpreter architecture.

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.

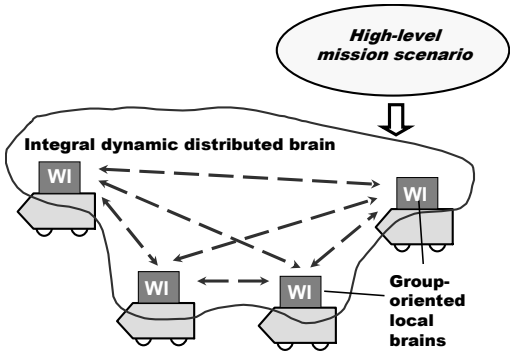


Figure 11: WPL interpreters (WI) forming distributed robotic brain.

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.

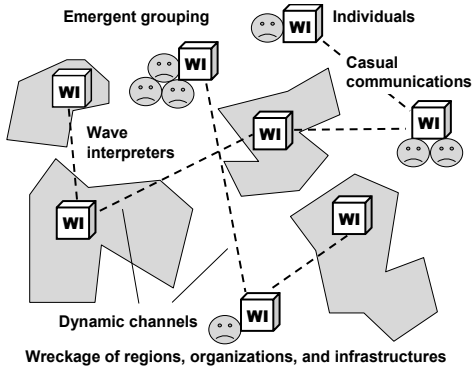


Figure 12: A disaster area with WPL interpreters embedded.

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)))))
```

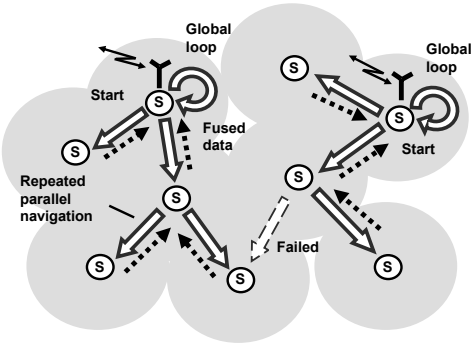


Figure 13: Collecting data by a sensor network.

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)
```

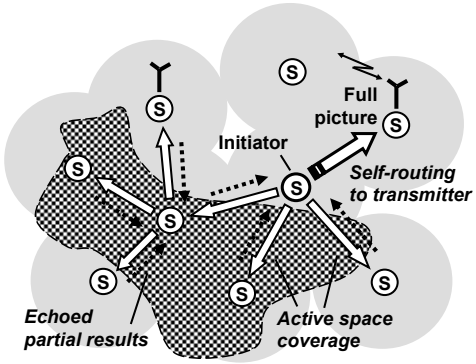


Figure 14: Space coverage with hierarchical assembling of a distributed phenomenon.

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.

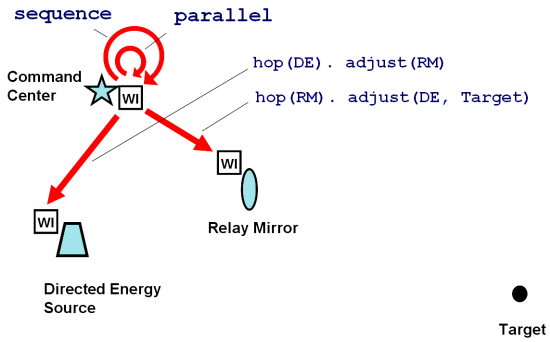


Figure 15: DE system operation, Snapshot 1.

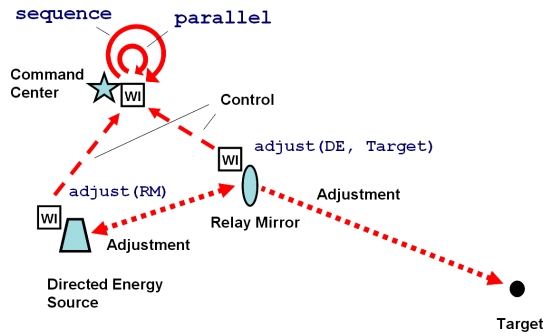


Figure 16: DE system operation, Snapshot 2.

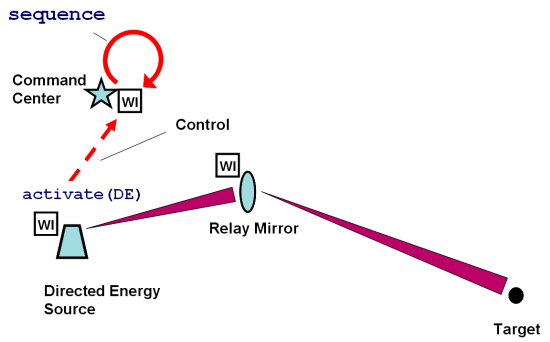


Figure 17: DE system operation, Snapshot 3.

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)
```

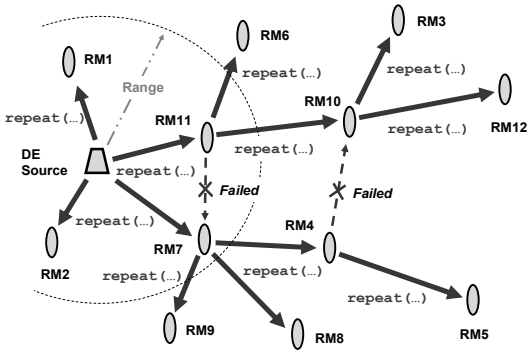


Figure 18: Dynamic shortest path tree through all RMs.

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)
```

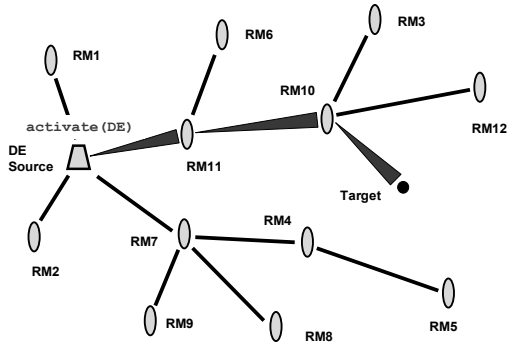


Figure 19: Energy delivery via the path found.

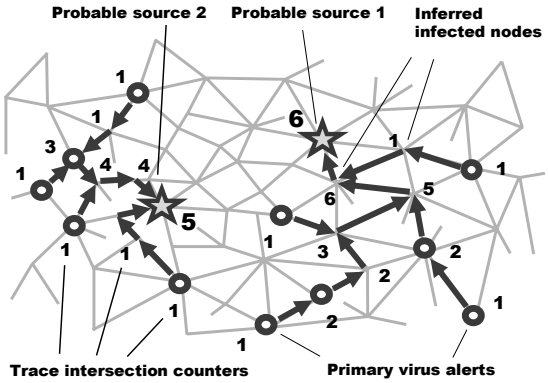


Figure 20: Finding virus sources in parallel.

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.

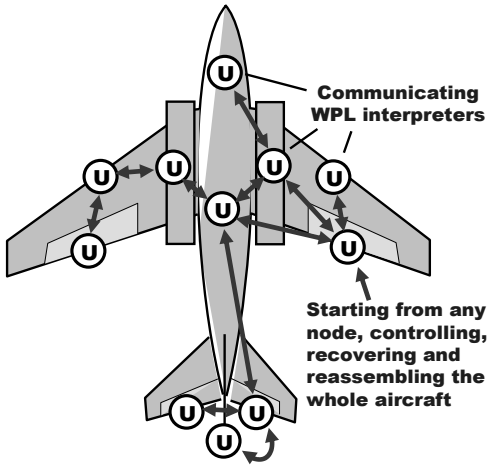


Figure 21: Aircraft self-analysis by the WPL network.

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).

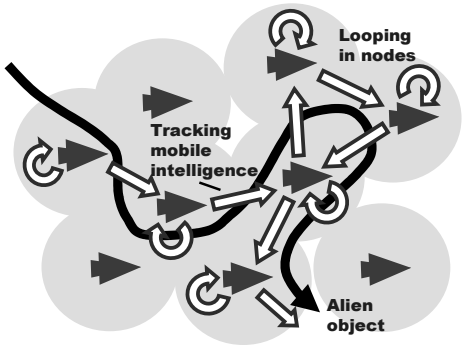


Figure 22: Collective tracking of a mobile object.

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.

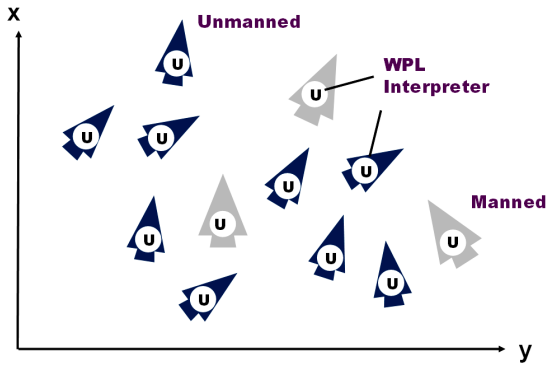


Figure 23: Initial distribution of units.

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.

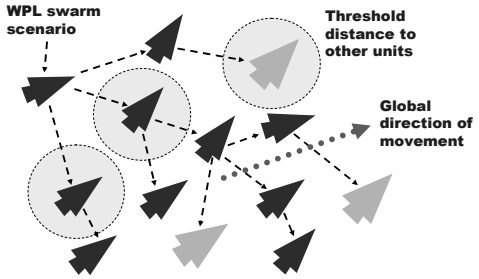


Figure 24: A swarm movement snapshot.

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)
```

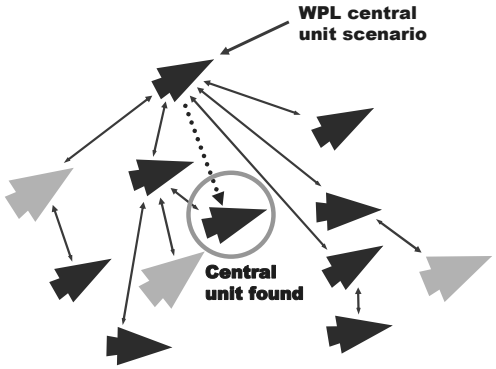


Figure 25: Finding topologically central unit.

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)))
```

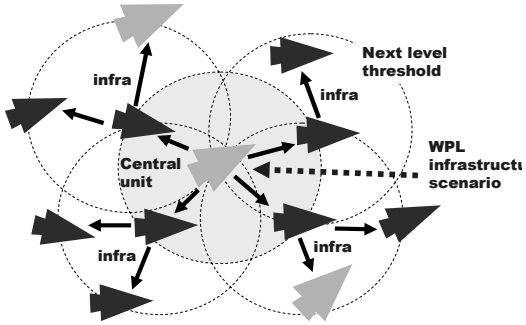


Figure 26: Hierarchical infrastructure built.

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))
```

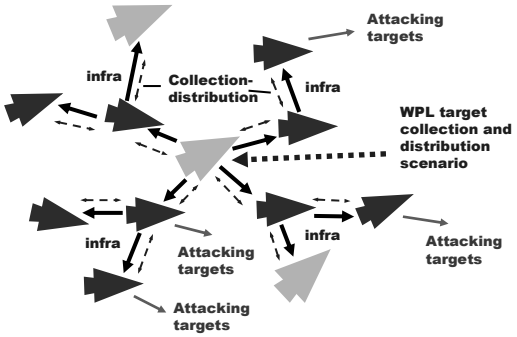


Figure 27: Hierarchical fusion and distribution of targets.

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.

INTELLIGENT CONTROL SYSTEMS AND OPTIMIZATION

FULL PAPERS

SYNCHRONIZATION OF ARM AND HAND ASSISTIVE ROBOTIC DEVICES TO IMPART ACTIVITIES OF DAILY LIVING TASKS

Duygun Erol[1] and Nilanjan Sarkar[2]
[1]*Department of Electrical & Electronics Engineering, Yeditepe University, Kayisdagi, Istanbul, Turkey*
[2]*Department of Mechanical Engineering, Vanderbilt University, Nashville, TN, U.S.A.*
duygunerol@yeditepe.edu.tr, nilanjan.sarkar@vanderbilt.edu

Keywords: Robot-assisted rehabilitation, robot-assisted rehabilitation for activities of daily living tasks, coordination of arm and hand assistive devices, hybrid system model.

Abstract: Recent research in rehabilitation indicates that tasks that focus on activities of daily living (ADL) is likely to show significant increase in motor recovery after stroke. Most ADL tasks require patients to coordinate their arm and hand movements to complete ADL tasks. This paper presents a new control approach for robot assisted rehabilitation of stroke patients that enables them to perform ADL tasks by providing controlled and coordinated assistance to both arm and hand movements. The control architecture uses hybrid system modelling technique which consists of a high-level controller for decision-making and two low-level assistive controllers (arm and hand controllers) for arm and hand motion assistance. The presented controller is implemented on a test-bed and the results of this implementation are presented to demonstrate the feasibility of the proposed control architecture.

1 INTRODUCTION

Stroke is leading cause a disability that results in high costs to the individual and society (Matchar, 1994). Literature supports the idea of using intense and task oriented stroke rehabilitation (Cauraugh, 2005) and creating highly functional and task-oriented practice environments (Wood, 2003) that increase task engagement to promote motor learning, cerebral reorganization and recovery after stroke. The task-oriented approaches assume that control of movement is organized around goal-directed functional tasks and demonstrated promising results in producing a large transfer of increased limb use to the activities of daily living (ADL) (Ada, 1994). The availability of such training techniques, however, is limited by the amount of costly therapist's time they involve and the ability of the therapist to provide controlled, quantifiable and repeatable assistance to movement. Consequently, robot-assisted rehabilitation that can quantitatively monitor and adapt to patient progress, and ensure consistency during rehabilitation has become an active research area to provide a solution to these problems. MIT-Manus (Krebs, 2004), MIME (Lum, 2006), ARM-Guide (Kahn, 2006) and the GENTLE/s (Loureiro, 2003) are the devices developed for arm rehabilitation, whereas Rutgers Master II-ND (Jack, 2001), the CyberGrasp (Immersion Corporation), a pneumatically controlled glove (Kline, 2005) and HWARD (Takahashi, 2005) are used for hand rehabilitation.

Even though existing arm and hand rehabilitation systems have shown promise of clinical utility, they are limited by their inability to simultaneously assist both arm and hand movements. This limitation is critical because the stroke therapy literature supports the idea that the ADL-focused tasks (emphasis on task-oriented training), which engage patients to perform the tasks in enriched environments have shown significant increase in the motor recovery after stroke. Robots that cannot simultaneously assist both arm and hand movements are of limited value in the ADL-focused task-oriented therapy approach. It is possible to integrate an arm assistive device and a hand assistive device to provide the necessary motion for ADL-focused task-oriented therapy. However, none of the existing controllers used for robot-assisted rehabilitation can be directly used for this purpose because they are not suited for controlling multiple systems in a coordinated manner. In this work, we address the controller design issue of a robot-assisted rehabilitation system that can simultaneously coordinate both arm and hand

motion to perform ADL tasks using an intelligent control architecture. The proposed control architecture uses hybrid system modeling technique that consists of a high-level controller and two low-level device controllers (e.g., arm and hand controllers). The versatility of the proposed control architecture is demonstrated on a test-bed consisting of an upper arm assistive device and a hand assistive device. Note that the presented control architecture is not specific to a given arm and hand assistive device but can be integrated with other previously proposed assistive systems.

In this paper, we first present the intelligent control architecture in Section 2, and then the rehabilitation robotic system and design details of the high-level controller are presented in Section 3. Later, results of the experiments to demonstrate the efficacy of the proposed controller architecture are presented in Section 4. Finally, the contributions of the work are presented in Section 5.

2 INTELLIGENT CONTROL ARCHITECTURE

Let us first present the proposed intelligent control architecture in the context of generic ADL tasks that require coordination of both arm and hand movement (e.g., eating, drinking, etc.). Stroke patients may not be able to complete the ADL tasks by themselves because of motor impairments. Thus, low-level arm assistive controller and low-level hand assistive controller may be used to provide assistance to the subject's arm and hand movement, respectively. The nature of assistance given to the patients and coordination of the assistive devices, however, could be impacted by various events during the ADL task (e.g., completion of a subtask, safety related events etc.). A high-level controller (HLC) may be used to allocate task responsibility between the low-level assistive controllers (LLACs) based on the task requirements and specific events that may arise during the task performance. HLC plays the role of a human supervisor (therapist) who would otherwise monitor the task, assess whether the task needs to be updated and determine the activation of the assistive devices. However, in general, the HLC and the LLACs may not communicate directly because each may operate in different domains. While the LLACs may operate in a continuous way, the HLC may need to make intermittent decisions in a discrete manner. Hybrid system theory provides mathematical tools that can accommodate both continuous and discrete systems in a unified manner. Thus, we take advantage of using a hybrid system model to design the proposed intelligent control architecture (Koutsoukos, 2000). In this architecture, the "Plant" represents both the assistive devices and their low-level assistive controllers and the Interface functions as analog-to-digital/digital-to-analog (AD/DA) adaptor.

The proposed control architecture for robot-assisted rehabilitation to be used to perform ADL tasks is presented in Fig. 1. In this architecture, the sensory information from the arm assistive device, the hand assistive device and the feedback from the human are monitored by the process-monitoring module through the interface. The sensory information (plant event) is converted to a plant symbol so that the HLC can recognize the event. Based on a plant symbol, the decision making module of the HLC sends its decision to the LLACs through the interface using the control symbols. Interface converts the control symbols to the plant inputs which are used to activate/deactivate the LLACs to complete the ADL task. The proposed control architecture is extendible in the sense that new events can be included by simply monitoring the new sensory information from the human and the assistive devices, and accommodated by introducing new decision rules.

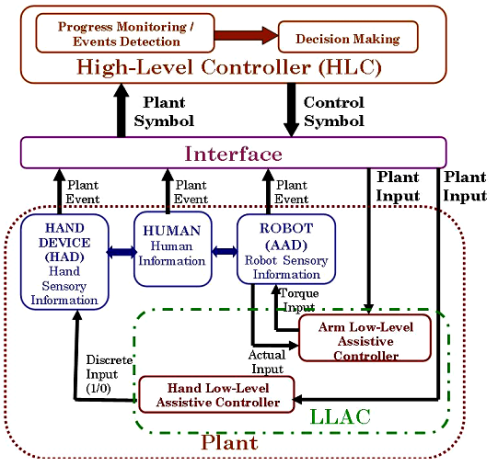


Figure 1: Control Architecture.

3 METHODOLOGY

The primary focus of this paper is to demonstrate how the assistive devices can be coordinated using the proposed intelligent control framework for a given ADL task. The intelligent control framework consists of a HLC and two low-level arm and hand assistive controllers. The focus of the paper is to design of HLC that can coordinate a number of given LLACs using the presented intelligent control

framework. First we briefly present the rehabilitation system used as the test-bed to implement the intelligent control framework. We then present a detailed discussion on the design and implementation of the HLC and its workings within the presented control framework using an ADL task.

3.1 Rehabilitation Robotic System - a Test Bed

The rehabilitation robotic system used in this work consists of an arm assistive device, hand assistive device and two sensory systems (contact detection and proximity detection systems) (Fig. 2A). We have modified a Power Grip Assisted Grasp Wrist-Hand Orthosis (Broadened Horizons) as a hand assistive device (Fig. 2B). A computer control capability in a Matlab/Realtime Windows Target environment is added in order to integrate the hand device in the proposed control architecture. The subject is asked to follow the opening/closing speed of the hand device and if the subject cannot follow the hand device movement, then the hand device provides assistance to complement subject's effort to open/close his/her hand. The PUMA 560 robotic manipulator is augmented with a force-torque sensor and a hand attachment device (Fig.2A) to provide assistance to the upper arm movement (Erol, 2007). A proportional-integral-derivative (PID) position control is used as a low-level arm assistive controller for providing robotic assistance to a subject. The subject is asked to pay attention to tracking the desired position trajectory (visually monitoring his/her actual and desired position trajectories on a computer screen) as accurately as possible. If the subject deviates from the desired motion, then the low-level arm assistive controller provides robotic assistance to complement the subject's effort to complete the task as required. We have designed a contact detection system to provide sensory information about grasping activity that may be a part of an ADL task of interest. The force-sensitive resistors (FSR) (Interlink Electronics, Inc.) are placed on the fingertip to estimate the forces applied on the object during the grasping task (Fig. 2C). When the subject starts grasping an object, then the voltage across the FSR changes as a function of the applied force. Additionally, a proximity detection system (PDS) is designed in order to detect the closeness of the subject's hand relative to the object to be grasped. The PDS contains a phototransistor (sensitive to infrared light) and an infrared emitter, which are mounted onto two slender posts close to the object facing each other. When the subject approaches to the object by moving his/her hand between these posts, the continuity of the receiving signal (infrared beam) is broken and the corresponding voltage change is used to generate an event to inform the HLC that the subject is close to the object.

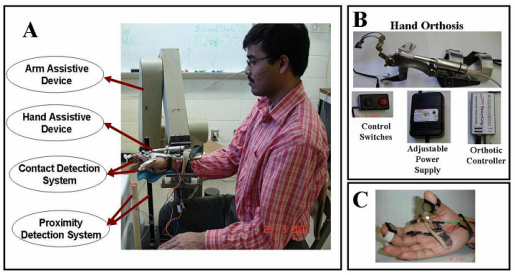


Figure 2A: Rehabilitation Robotic System, Figure 2B: Hand Assistive Device and Figure 2C: Force Sensor Resistors Placement on the Fingers.

3.2 Task Description

The main focus of this work is to present how a HLC is designed and how it functions within the proposed intelligent control architecture. The ADL tasks consist of several primitive movements such as reaching towards an object, grasping the object, lifting the object from the table, using the object for eating/drinking, and placing the object back on the table (Murphy, 2006) and they all require coordination between arm and hand movements. In here, we choose one of the ADL tasks, called drinking from a bottle (DFB) task to explain the HLC. We decompose the DFB task into the following primitive movements: i) reach towards the bottle while opening the hand, ii) reach the bottle, iii) close the hand to grasp the bottle, iv) move the bottle towards the mouth, v) drink from a bottle using a straw, vi) place the bottle back on the table, vii) open the hand to leave the bottle back on the table and viii) go back to the starting position. Note that similar task decomposition could be defined for other ADL tasks.

3.3 Design Details of the High-Level Controller

3.3.1 Theory of Hybrid Control Systems

The hybrid control systems consist of a plant which is generally a continuous system to be controlled by a discrete event controller (DES) connected to the plant via an interface in a feedback configuration (Koutsoukos, 2000). If the plant is taken together with the interface, then it is called a DES plant

model. The DES plant model is is a nondeterministic automaton which is represented by $G = (\tilde{P}, \tilde{X}, \tilde{R}, \psi, \lambda)$. Here $\tilde{P}$ is the set of discrete states; $\tilde{X}$ is the set of plant symbols generated based on the events; and $\tilde{R}$ is the set of control symbols. $\psi : \tilde{P} \times \tilde{R} \rightarrow 2^{\tilde{P}}$ is the state transition function. For a given DES plant state and a given control symbol, state transition function is defined as the mapping from $\tilde{P} \times \tilde{R}$ to the power set $2^{\tilde{P}}$, since for a given plant state and a control symbol the next state is not uniquely defined. The output function, $\lambda : \tilde{P} \times \tilde{P} \rightarrow 2^{\tilde{X}}$, maps the previous and current plant states to a set of plant symbols. The DES controller, which is called the HLC in this work, controls the DES plant. The HLC is responsible to coordinate the assistive devices based on both task and the safety requirements. The HLC is modeled as a discrete-event system (DES) deterministic finite automaton, which is specified by $D = (\tilde{S}, \tilde{X}, \tilde{R}, \delta, \phi)$. Here $\tilde{S}$ is the set of control states. Each event is converted to a plant symbol, where $\tilde{X}$ is the set of such symbols, for all discrete states. The next discrete state is activated based on the current discrete state and the associated plant symbol using the following transition function: $\delta : \tilde{S} \times \tilde{X} \rightarrow \tilde{S}$. In order to notify the LLACs the next course of action in the new discrete state, the HLC generates a set of symbols, called control symbols, denoted by $\tilde{R}$, using an output function: $\phi : \tilde{S} \rightarrow \tilde{R}$.

3.3.2 Modelling of an ADL Task using Hybrid Control System

Now we discuss how the above theory could be used to model and control an ADL task (e.g., the DFB task) for rehabilitation therapy. The first step is to design the DES plant and define the hypersurfaces that separates different discrete states. The hypersurfaces are used to detect the events and are decided considering the capabilities of the rehabilitation robotic systems and the requirements of the task. The following hypersurfaces are defined: $h_1 = vir > 0, h_2 = |x| \geq |x_t - \in|$, $h_3 = (vfsr < vth) \wedge (hcb == 0)$, $h_4 = (|x| \leq |x_t - \in|) \wedge (hob == 0)$, $h_5 = (t == t_{hand})$, $h_6 = \theta_l < \theta < \theta_u$, $h_7 = \tau_r \geq \tau_{rth}$, $h_8 = \tau_h \geq \tau_{hth}$, $h_9 = (eb == 1)$, $h_{10} = (pb == 1)$ $h_{11} = (pb == 0) \wedge (eb == 0)$, where vir is the voltage in the PDS system. x and x_t are the hand actual position and the object's position, respectively. $\in$ is a value used to determine if the subject is close enough to the object's position. $vfsr$ and vth are the voltage across the FSRs and the threshold voltage, respectively. The values of hob and hcb are binary values, which could be *1* when it is pressed and *0* when it is released. t and t_{hand} are the current time and the final time to complete hand opening, respectively. θ_l and θ_u represent the set of lower and upper limits of the joint angles, respectively and θ is the set of the actual joint angles. τ_r and τ_{rth} are the torque applied to the motor of the arm assistive device and the torque threshold value, respectively. The torque applied to the motor of the hand assistive device and its threshold value is defined as τ_h and τ_{hth}, respectively. The values of eb and pb are binary values, which could be *1* when it is pressed and *0* when it is released. The above hypersurfaces can be classified into two groups: i) the hypersurfaces that are defined considering the requirements of the tasks (i.e., h_1 - h_5), and ii) the hypersurfaces that are defined considering the capabilities of the rehabilitation robotic system (i.e., h_6 - h_{11}). The hypersurfaces provide information to the HLC in order to make decisions for execution of the task in a safe manner. The set of DES plant states $\tilde{P}$ is based upon the set of hypersurfaces realized in the interface. Each region in the state space of the plant, bounded by the hypersurfaces, is associated with a state of the DES plant. During the execution of the task, the state evolves over time and the state trajectory enters a different region of the state space by crossing the hypersurfaces and a plant event, occurs when a hypersurface is crossed. A plant event generates a plant symbol to be used by the HLC. The plant symbols $\tilde{X}$ in the DES plant model $G = (\tilde{P}, \tilde{X}, \tilde{R}, \psi, \lambda)$ are defined as follows:

$$\tilde{x}[n] = \lambda(\tilde{p}[n-1], \tilde{p}[n]) \quad (1)$$

where $\tilde{x} \in \tilde{X}$, $\tilde{p} \in \tilde{P}$, λ is the output function and n is the time index that specifies the order of the symbols in the sequence. In (1) the plant symbol, $\tilde{x}$, is generated as an output function of the current and the previous plant state. We define the following plant symbols considering the hypersurfaces discussed before: i) $\tilde{x}_1$, the arm approaches to the bottle with the desired grip aperture, which is generated when h_1 is crossed, ii) $\tilde{x}_2$, the arm reaches to the bottle, which is generated when h_2 is crossed, iii) $\tilde{x}_3$, the hand reaches desired grip closure to

grasp the bottle, which is generated when h_3 is crossed, iv) $\tilde{x}_4$, the arm leaves the bottle on the table, which is generated when h_4 is crossed, v) $\tilde{x}_5$, the hand reaches desired grip aperture, which is generated when h_5 is crossed, vi) $\tilde{x}_6$, safety related issues happened such as the robot joint angles are out of limits (when h_6 is crossed), or the robot applied torque is above its threshold (when h_7 is crossed), or hand device applied torque is above its threshold (when h_8 is crossed) or emergency button is pressed (when h_9 is crossed), vii) $\tilde{x}_7$, the subject presses the pause button, which is generated when h_{10} is crossed, and viii) $\tilde{x}_8$, the subject releases the pause button which is generated when h_{11} is crossed. $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4, \tilde{x}_5, \tilde{x}_6, \tilde{x}_7, \tilde{x}_8\}$ is the set of plant symbols.

In this work, the purpose of the DES controller (HLC) is to activate/deactivate the assistive devices in a coordinated manner to complete the DFB task. In order to perform this coordination, the following control states are defined: $\tilde{s}_1$: both the hand device and arm device are active, ii) $\tilde{s}_2$: the arm device alone is active, iii) $\tilde{s}_3$: the hand device alone is active to close the hand, iv) $\tilde{s}_4$: the hand device alone is active to open the hand, v) $\tilde{s}_5$: both the arm and hand devices are idle. Additionally, a memory control state ($\tilde{s}_6$) is defined to detect the previous control actions when the subject wants to continue with the task after he/she presses pause button. $\tilde{S} = \{\tilde{s}_1, \tilde{s}_2, \tilde{s}_3, \tilde{s}_4, \tilde{s}_5, \tilde{s}_6\}$ is the set of control states in this application. When new control actions are required for an ADL task, new control states can be included in the set of the states, $\tilde{S}$. The transition function $\delta : \tilde{S} \times \tilde{X} \rightarrow \tilde{S}$ uses the current control state and the plant symbol to determine the next control action that is required to update the ADL task, where $\tilde{s} \in \tilde{S}$, $\tilde{x} \in \tilde{X}$, $\tilde{r} \in \tilde{R}$, and n is the time index that specifies the order of the symbols in the sequence.

$$\tilde{s}[n] = \delta(\tilde{s}[n-1], \tilde{x}[n]) \quad (2)$$

The HLC generates a control symbol $\tilde{r}$, which is unique for each state, $\tilde{s}$ as given in (3). In here, the following control symbols are defined: i) $\tilde{r}_1$: drive arm device towards the object while driving hand device to open the hand, ii) $\tilde{r}_2$: drive arm device to perform various primitive arm motion such as move the bottle towards the mouth etc., iii) $\tilde{r}_3$: drive hand device to close the hand to grasp the bottle, iv) $\tilde{r}_4$: drive hand device to open the hand to leave the bottle, and v) $\tilde{r}_5$: make arm and hand devices idle. The set of control symbols are defined as $\tilde{R} = \{\tilde{r}_1, \tilde{r}_2, \tilde{r}_3, \tilde{r}_4, \tilde{r}_5\}$.

$$\tilde{r}[n] = \phi(\tilde{s}[n]) \quad (3)$$

The LLACs cannot interpret the control symbols directly. Thus the interface converts the control symbols into continuous outputs, which are called plant inputs. The plant inputs are then sent to the LLACs to modify the ADL task. We define the following plant inputs: i) if $\tilde{r} = \tilde{r}_1$ then provide 1 to activate both arm and hand devices, ii) if $\tilde{r} = \tilde{r}_2$, then provide 2 to activate only the arm device, iii) if $\tilde{r} = \tilde{r}_3$, then provide 3 to activate only the hand device to close hand, iv) if $\tilde{r} = \tilde{r}_4$, then provide 4 to activate only the hand device to open hand, v) if $\tilde{r} = \tilde{r}_5$, then provide 0 to keep both arm and hand devices idle. Note that the design of the elements of the DES plant and the DES controller is not unique and is dependent on the task and the sensory information available from the robotic system.

4 RESULTS

In this section we present two experiments that were conducted to demonstrate the feasibility and usefulness of the proposed control architecture. Since we experiment with an unimpaired subject who could ideally do the DFB task by himself (unlike a real stroke patient) we instructed him to be passive so that we can demonstrate that the proposed control architecture was solely responsible for the coordinated arm and hand movements (which is the main objective of this work) as needed to complete the DFB task. Such an experimental condition is not only helpful to demonstrate the efficacy of the proposed control architecture but also could occur when a low functioning stroke survivor participates in a task-oriented therapy who will initially need continuous robotic assistance to perform an ADL.

The subject was asked to wear the hand device and then place his forearm on the hand attachment (Fig. 2A). In the first experiment (E1), we asked the subject to perform the DFB task, where the task proceeded as planned (i.e., there was no event

occurred during the task that would require dynamic modification of the execution of the task; however, it still needed the necessary coordination between hand and arm motion). We designed a DFB task trajectory in consultation with a physical therapist as shown in Fig. 3. Fig. 3 shows how the DFB task was supposed to proceed: the subject was required to reach the bottle (A-B), grasp the bottle by applying a certain amount of force (B-C), bring the bottle to the mouth (C-D), drink water (D-E), bring back the bottle from where he picked it up (E-F), open hand to release the bottle (F-G), and then go back to the starting position (G-H). Furthermore, the desired trajectory from A-B into A-A' and A'-B trajectories have been decomposed because in naturalistic movement it has been shown that a subject reached his/her maximum aperture approximately two-third of the way through the duration of the reaching movement (Jeannerod, 1981).

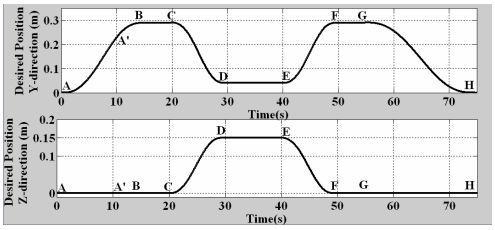


Figure 3: Desired Motion Trajectories for a DFB task.

The overall mechanism for the high-level control, which is used to activate/deactivate the LLACs, is shown in Fig. 4. When a device is active, we mean it tracks a non-zero trajectory and when the device is idle, we mean the device remains in its previous position set points.

Let us now explain how the HLC accomplished the DFB task using the control mechanism given in Fig. 4. When the DFB task started, $\tilde{s}_1$ became active where both arm and hand devices remained active till point A' to help the subject to open his hand while moving towards the bottle. Then at point A', the *vir* crosses a predefined threshold value, which confirmed that the subject reached close to the bottle ($\tilde{x}_1$ was generated) and $\tilde{s}_2$ state became active. The arm device remained active to help the subject to reach the bottle and the hand device was idle from A' to B. After that at point B, when the subject's position, x, was close to the bottle position, x_t, and then $\tilde{x}_2$ was generated and $\tilde{s}_3$ state became active. If $\tilde{s}_3$ was active, then the hand device remained active to assist the subject to grasp the bottle. Then, at point C, the *vfsr* value was dropped below the threshold ($\tilde{x}_3$ was generated) and $\tilde{s}_2$ state became active again. The arm device remained active to assist the subject to move the bottle to his mouth, to drink water using a straw and at the end to leave the bottle on the table. When the subject brought the bottle back on the table at point F, $\tilde{x}_4$ was generated and $\tilde{s}_4$ state became active and the hand device remained active to help the subject to open his hand till G. Then the subject reached the desired grip aperture $\left(t == t_{hand}\right)$, $\tilde{x}_5$ was generated and $\tilde{s}_2$ state became active. The arm device remained active to help the subject to go back to the starting position. The actual trajectory of the subject was exactly same as the desired trajectory given in Fig. 3. The subject's hand configuration diagram was given in Fig. 5. It could be seen from the figures that the subject was able to track the desired trajectories while opening/closing his hand at desired times.

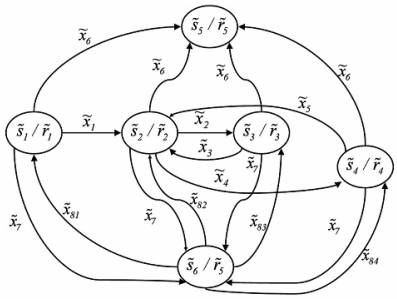


Figure 4: Control Mechanism for the HLC.

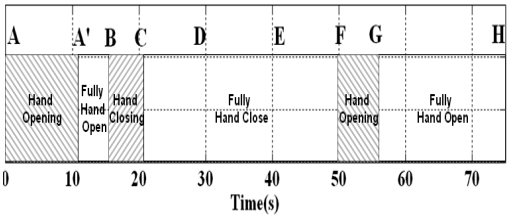


Figure 5: Hand Configuration Diagram for E1.

In the second experiment (E2), we demonstrated that if an event takes place at some point of time during the task execution that requires modification of the desired task trajectory such as a stroke patient wants to pause for a while due to some discomfort, then the HLC has the ability to dynamically modify the desired trajectory using the control mechanism given in Fig. 4. In this case, the subject started the execution of the task with the same desired trajectory as shown in Fig. 3 (which is the dotted trajectory in Fig. 6). During the execution of the task

at time *t'*, the subject pressed the pause button at time *t'* (when $\tilde{s}_1$ was active), the plant symbol $\tilde{x}_7$ was generated and $\tilde{s}_6$ state became active and both the arm and hand devices became idle. $\tilde{s}_6$ state stored the previous active state. When the subject released the pause button at time *tt'* to continue the task execution, $\tilde{x}_{81}$ was generated and $\tilde{s}_1$ became active again to activate both the arm and hand devices to resume the task execution. The rest of the desired trajectory had been generated in the same way as it was described in E1 (Fig. 6-solid lines). It could be noticed from Fig. 6-solid lines that at time of *t'*, the assistive devices remained in their previous set points. Additionally, the subject's position at time of the *tt'* was automatically detected and taken as an initial position to continue the task where it was resumed with zero initial velocity (Fig. 6-solid lines). If the HLC did not modify the desired trajectories to register the intention of the subject to pause the task, then i) the desired motion trajectories would start at point *tt'* with a different starting position and a non-zero velocity (Fig. 6 -dotted lines), which could create unsafe operating conditions, and ii) the subject would close/open his hand at undesirable times. We had also noticed that the subject's actual trajectory was same as given in Fig. 6-solid line. Fig. 7 demonstrated the subject's hand configuration diagram for E2. It can be noticed that the subject was able to track the modified desired trajectory and he was able to coordinate his arm and hand motions in a safe and desired manner.

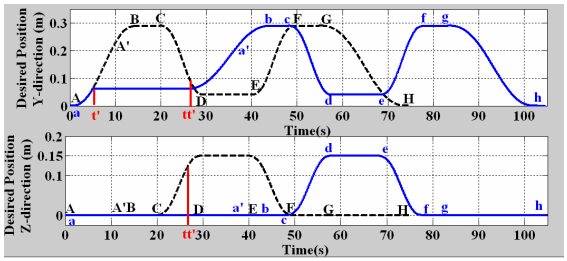


Figure 6: Desired Trajectory for the DFB task when an Unplanned Event Happened.

It is conceivable that one could pre-program all types of desired trajectories beforehand such that they could address all types of unplanned events, and retrieve them as needed. However, for non-trivial tasks, designing such a mechanism might be too difficult to manage and extend as needed.

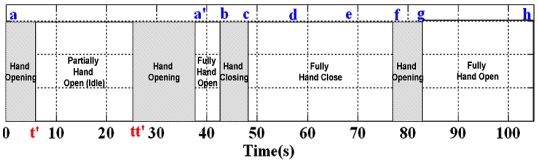


Figure 7: Hand Configuration Diagram for E2.

5 DISCUSSION/ CONCLUSIONS

The purpose of this work is to design a versatile control mechanism to enable robot-assisted rehabilitation in a task-oriented therapy that involves tasks requiring coordination of motion between arm and hand. In order to achieve this goal we design a new intelligent control architecture that is capable of coordinating both the arm and the hand assistive devices in a systematic manner. This control architecture exploits hybrid system modeling technique to provide the robotic assistance to enable a subject to perform ADL tasks that may be needed in a task-oriented therapy and has not been explored before for rehabilitation purpose. Hybrid system modeling technique offers systematic control design tools that provide design flexibility and extendibility of the controller, which gives ability to integrate multiple assistive devices and to add/modify various ADL task requirements in the intelligent control architecture. The control architecture combines a high-level controller and low-level assistive controllers (arm and hand). In here, the high-level controller is designed to coordinate with the low-level assistive controllers to improve the robotic assistance with the following objectives: 1) to supervise the assistive devices to produce necessary coordinated motion to complete a given ADL task, and 2) to monitor the progress and the safety of the ADL task such that necessary dynamic modifications of the task execution can be made to complete the given task in a safe manner.

Although the focus of the current work is to present a new high-level control methodology for rehabilitation that is independent of the low-level controllers, we want to mention the limitations of the hand and arm assistive devices used in the presented work. The hand device used in this paper does not allow independent control of fingers in performing various hand rehabilitation tasks. As discussed earlier, the focus of the paper is to present how arm and hand motion can be dynamically coordinated to accomplish ADL tasks. In that respect, the current hand device allowed us to perform an ADL task that showed the efficacy of the presented high-level controller. A more functional hand device would

allow the patients to perform more complex ADL tasks. Note that this current hand device is being used with C5 quadriplegic patients to complete their ADL tasks such as picking up bottle etc. (Broadened Horizons). We are also aware that a PUMA 560 robotic manipulator might not be ideal for rehabilitation applications. However the use of safety mechanisms, both in hardware (e.g., emergency button, quick arm release mechanism etc.) and in software (e.g., within the design of the high-level controller) will minimize the scope of injuries. Note that the proposed control architecture is not specific to the presented assistive devices but can also be integrated with other assistive devices.

We believe that such a robot-assisted rehabilitation system with capabilities of coordination of both arm and hand movement is likely to combine the advantages of robot-assisted rehabilitation systems with the task-oriented therapy. In this paper, the efficacy of the proposed intelligent controller is demonstrated with healthy human subject. We are aware that a stroke patient with a spastic arm is much more different from a healthy subject following the robotic moves. In that respect, more functional assistive devices and their corresponding low-level controllers can be integrated inside the proposed intelligent controller to allow stroke patients to take part in task-oriented therapy. As a future work, it is possible to use intelligent robot-assisted rehabilitation systems in clinical trials to understand on how impairment changes carryover of gained functional abilities to real living environments and how robot-assisted environments influence these changes.

ACKNOWLEDGEMENTS

We gratefully acknowledge the help of Dr. Thomas E. Groomes and Sheila Davy of Vanderbilt University's Stallworth Rehabilitation Hospital for their feedback about task design and Mark Felling who is C5 quadriplegic patient for his feedback about the hand assistive device.

REFERENCES

Ada, L., Canning, C. G., Carr, J. H., Kilbreath, S. L. & Shepherd, R. B. (1994). Task-specific training of reaching and manipulation. *In Insights into Reach to Grasp movement*, *105*, 239-265.

Broadened Horizons, http://www.broadenedhorizons.com/

Cauraugh J. H., Summers, J.J. (2005). Neural plasticity and bilateral movements: A rehabilitation approach for chronic stroke. *Prog Neurobiol.*, *75*, 309-320.

Erol, D. & Sarkar, N. (2007), Design and Implementation of an Assistive Controller for Rehabilitation Robotic Systems. *Inter. J. of Adv. Rob. Sys*, *4*(*3*), 271-278.

Immersion Corporation, http://www.immersion.com/.

Jack, D., Boian, R., Merians, A. S., Tremaine, M., Burdea, G. C., Adamovich, S. V., Recce, M. & Poizner, H. (2001). Virtual reality-enhanced stroke rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *9*, 308–318.

Jeannerod, M., (1981). '*Intersegmental coordination during reaching at natural visual objects'*. In Attention and performance IX, J. L. a. A. Baddeley, Ed. New Jersey, (pp. 153-168).

Kahn, L. E., Zygman, M. L., Rymer, W. Z. & Reinkensmeyer, D. J. (2006). Robot-assisted reaching exercise promotes arm movement recovery in chronic hemiparetic stroke: a randomized controlled pilot study. *J. Neuroengineering Rehabil*, *3*,1-13.

Kline, T., Kamper, D. & Schmit, B. (2005). Control system for pneumatically controlled glove to assist in grasp activities. *IEEE 9th International Conference on Rehabilitation Robotics*, pp. 78 - 81.

Krebs, H. I., Ferraro, M., Buerger, S. P., Newbery, M. J., Makiyama, A., Sandmann, M., Lynch, D., Volpe, B. T. & Hogan, N. (2004). Rehabilitation robotics: pilot trial of a spatial extension for MIT-Manus. *J Neuroengineering Rehabil*, *1*, 1-15.

Koutsoukos, X. D., Antsaklis, P. J., Stiver, J. A. & Lemmon, M.D. (2000). Supervisory control of hybrid systems. *IEEE on Special Issue on Hybrid Systems: Theory and Applications*, *88*, 1026-1049.

Matchar D. B., Duncan, P. W. (1994). Cost of Stroke, *Stroke Clin. Updates*. *5*, 9-12.

Murphy, M. A., Sunnerhagen, K. S., Johnels, B. & Willen, C. (2006). Three-dimensional kinematic motion analysis of a daily activity drinking from a glass: a pilot study. *J. Neuroengineering Rehabil.*, *16*, 3-18.

Loureiro, R., Amirabdollahian, F., Topping, M., Driessen, B. & Harwin, W. (2003). 'Upper limb mediated stroke therapy - GENTLE/s approach. *Autonomous Robots*, *15*, 35-51.

Lum, P. S., Burgar, C. G., Van der Loos, H. F. M., Shor, P. C., Majmundar, M. & Yap, R. (2006). MIME robotic device for upper-limb neurorehabilitation in subacute stroke subjects: A follow-up study. *J. of Rehab. Res. & Dev.*, *43*, 631-642.

Takahashi, C. D., Der-Yeghiaian, L., Le, V. H. & Cramer, S. C. (2005). A robotic device for hand motor therapy after stroke. *IEEE 9th International Conference on Rehabilitation Robotics*, pp. 17 – 20.

Wood, S. R., Murillo, N., Bach-y-Rita, P., Leder, R. S., Marks, J. T. & Page, S. J. (2003).Motivating, game-based stroke rehabilitation: a brief report. *Top Stroke Rehabil*, *10*, 134-140.

LOSS MINIMIZATION OF INDUCTION GENERATORS WITH ADAPTIVE FUZZY CONTROLLER

Durval de Almeida Souza*, José Antonio Dominguez Navarro**
** Federal Center for Technology Education of Bahia – CEFET-BA/ UE de Vitória da Conquista*
3150 Zabelê, 45030-220, Vitória da Conquista, Bahia, Brazil
durval@cefetba.br

Jesús Sallán Arasanz
*** University of Zaragoza, Department of Electrical Engineering, Calle Maria de Luna s/n*
50018 Campus del Actur, Zaragoza, Spain
jadona@unizar.es, jsallan@.unizar.es

Keywords: Inverter-fed induction generator, loss minimization, vector control, fuzzy controller, adaptive control.

Abstract: In this paper a new technique for efficiency optimization of induction generator working at variable speed and load is introduced. The technique combines two distinct control methods, namely, on-line search of the optimal operating point, with a model based efficiency control. For a given operating condition, characterized by a given turbine speed (ω_T) and electric torque (T_e), the search control is implemented via the "Rosenbrock" method, which determines the flux level that results in the maximum output power. Once the optimal flux level has been found, this information is used to update the rule base of a fuzzy controller, which plays the role of an implicit mathematical model of the system. Initially, for any load condition the rule base yields the rated flux value. As the optimum points associated with the different operating conditions are identified, the rule base is progressively updated, so that the fuzzy controller learns to model the optimal operating conditions for the entire torque-speed plane. After every rule base update, the Rosenbrock controller output is reset, but it is kept active to track possible minor deviations of the optimum point.

1 INTRODUCTION

In the last decade, wind power is integrated into the electrical grid and accounts for a noticeable share of the total power generation (Kumar, 2007). In this context the inverter-fed induction generator has been identified as a possible source of energy to be used in modern micro and high power applications (Leidhold, 2002). The presence of a converter in a drive system enables an extra degree of freedom, namely, flux adjustment. In fact, efficiency optimization in adjustable speed drives is usually obtained by machine flux control. This is due to the fact that in electric machines, maximum efficiency is achieved when the copper losses become equal to the core losses. Typically, under partial load operation, rated flux condition results in relatively large core losses, small copper losses, and poor efficiency. By decreasing the flux, core losses are reduced, whereas an increase in copper losses takes place. The total losses, however, are reduced, and the efficiency is improved (Sousa, 1995). In this work, a new efficiency optimization technique is introduced. It is applicable to any adjustable speed drive, but it is illustrated here for an induction generator under field-oriented control. The technique combines two distinct control strategies, namely, on-line search and model base control. For a given operating condition, characterized by a given turbine speed (ω_T) and electric torque (T_e), the search control is implemented via the Rosenbrock method, which determines the flux level that results in maximum output power. Once the optimal flux level has been found, this information is used to update the rule base of a fuzzy controller, which plays the role of an implicit mathematical model of the system. Figure 1 shows the scheme of the Inverter-Fed Induction Generator used in this research.

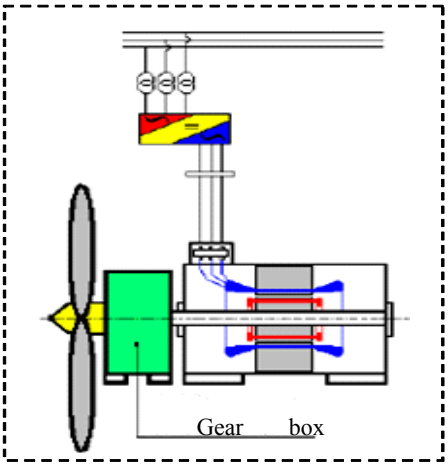


Figure 1: Inverter-fed induction generator connected to grid. Figure adapted by (Rüncos).

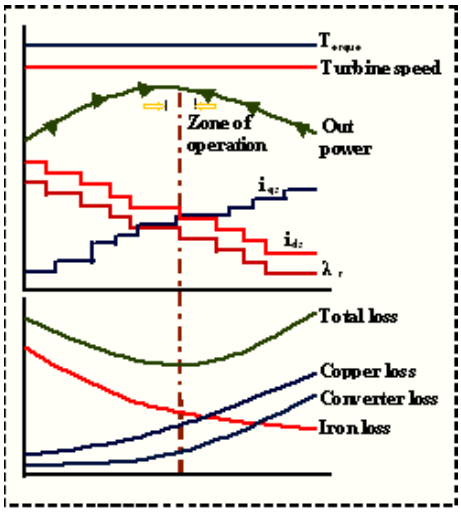


Figure 2: Philosophy for search method of efficiency optimization.

2 EFFICIENCY OPTIMIZATION

2.1 Search Control

In steady state with constant torque and speed, the flux component of the chain is decreased whereas the torque component is increased. Initially a reduction in total stator current occurs ($I_s = i_{qs} - j\, i_{ds}$), and consequently a reduction in stator copper losses, but rotor copper losses are increased. As the reduction in both stator and iron losses are higher than the increase in rotor losses, the total losses are reduced. If i_{ds} is continuously reduced, a reduction in the total losses will occur until the moment when the increase in copper losses becomes higher that the reduction in core losses, that is, the point minimum losses will be exceeded. The determination of this point, of minimum losses, that corresponds to optimum efficiency, can be performed using different procedures. This philosophy is illustrated in figure 2.

A turbine is submitted to a load proportional to the square of the angular speed of the wind. Thus, in weak wind conditions, below 7 m/s, the generator works typically with light rotor load. In such conditions the intensity of the rotor flux of the induction generator (IG), commanded in vector control for i_{ds}, can be reduced to values below nominal flux, reducing reactive circulation, diminishing iron losses and consequently increasing global efficiency of both inverter and machine (Sousa,1995), something essential at low wind speed to improve power extraction capacity, (Simões, 1999).

As mentioned before, rated flux results in excessive core losses and poor efficiency under light load conditions. Another aspect worth mentioning is the need to prevent machine torque disturbances during the efficiency optimization control. Under vector control, the developed torque can be expressed as:

$$T_e = k_t \lambda_r i_{qs} \quad (1)$$

where: λ_r is the rotor flux and i_{qs} is the torque component of the stator current, and k_t is a constant. If the flux is reduced to improve efficiency, i_{qs} must be increased accordingly, such that their product remains constant at any given time.

2.1.1 The Rosenbrock Method

This is a very simple method, and guaranteed to converge. The reference for the flux component of the stator current (i_{ds}^*) is modified in small steps in a given direction, while the system approaches the optimum efficiency point, i.e., the measured change in output power in the n-th step is positive ($\Delta P(n) > 0$). When the method recognizes that an "overshoot" has occurred ($\Delta P(n) < 0$), it reverses the search direction, with a reduced step size, (J. Moreno-Eguilaz, 1997,). The search process can be mathematically expressed as in (2):

$$i_{ds}^*(n+1) = i_{ds}^*(n) + k\Delta i_{ds}^*(n); \quad \begin{cases} k = 1; & \text{if } \Delta P(n) > 0 \\ k = -\frac{1}{2}; & \text{if } \Delta P(n) < 0 \end{cases} \quad (2)$$

where: $\Delta P(n) = P(n) - P(n-1)$ and $\Delta i_{ds}^*(n) = i_{ds}^*(n) - i_{ds}^*(n-1)$.

3 THE PROPOSED SYSTEM

The indirect method of vector control is applied to the IG. It derives the reference for the torque component of the stator current (i_{qs}^*) from the speed error, utilizing a conventional proportional-Integral (PI) controller. As the system operates with variable flux, a compensation block is introduced at the output of the speed PI controller. Essentially, this block multiplies the original PI controller output by the ratio rated flux / actual flux (estimate).

The reference for the flux component of stator flux (i_{ds}^*) is not kept constant here, as in the majority of high performance IM drive systems. It is defined as the sum of two block outputs: $i_{ds}^*(k) = i_{ds}'^*(k) + \Sigma\Delta i_{ds}^{*'}$. The first term ($i_{ds}'^*$) is obtained from a fuzzy controller, that from two inputs (speed (ω_r) and estimate load torque(T_L), derives a preliminary reference ($i_{ds}'^*$) through fuzzy inference. The second one ($\Sigma\Delta i_{ds}'^*$) is the actual output of a search controller, based on the Rosenbrock method. Its value represents the accumulated control actions taken by the controller during the search process up to the current iteration (n), as can be seen in Fig. 3.

When the system is turned on for the first time, the rule base of the fuzzy controller contains rated d-axis current reference ($i_{ds}'^*$) for all rules, i.e. for any speed and load torque point. When a steady state condition is detected, the search controller becomes active. After a few steps, it reaches the optimum efficiency point by imposing the $\Sigma\Delta i_{ds}'^*$ change to the original reference ($i_{ds}'^*$) from the fuzzy controller. Once the controller recognizes this optimum condition, the rule base can be updated to reflect the knowledge of the optimum flux level for this particular operating point (load torque and speed). At the same time, the search controller output must be reset, to prevent erroneous operation. When the optimum point is found, the rule base is updated, and the output of the search controller reset, such that, effectively, $i_{ds\,opt}^* = i_{ds}'^*$.

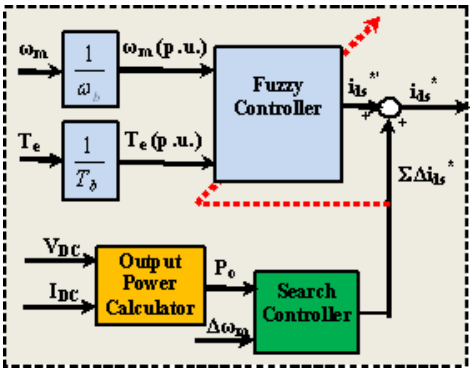


Figure 3: Hybrid efficiency controller.

As the optimum efficiency points related to the several operating conditions are identified, the rule base is progressively updated, such that the fuzzy controller "learns" the optimum flux level for the entire torque-speed plane. Once completed the learning process, the output of the fuzzy controller already reflects the optimum flux level, and the fuzzy controller is capable of driving the system at optimum efficiency without delays. To prevent sub-optimal operation, the search controller remains active to track possible deviations of the optimum point. Under transient conditions, the search process is cancelled, and the flux reference is solely derived from the fuzzy controller. It is worth noticing that no switching of strategies is required, since higher torques demands are normally met by imposing higher flux levels, i.e., the optimum level of flux for higher torques is close to the rated flux value. This methodology is summarized in Figure 4.

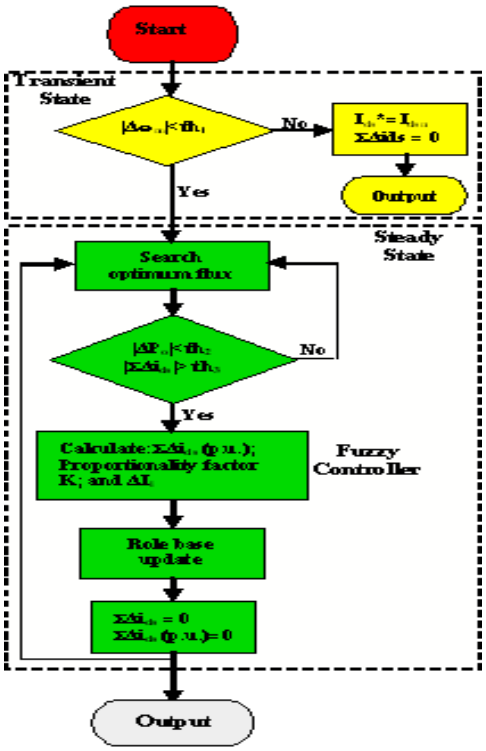
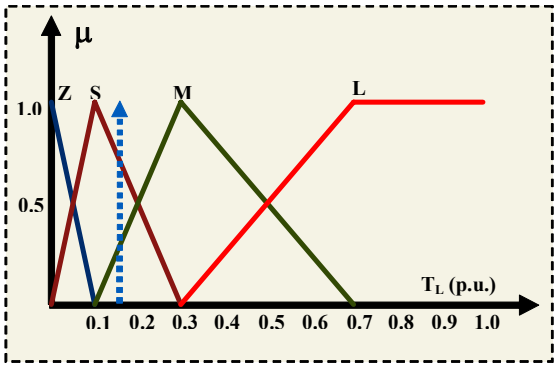


Figure 4: State diagram for efficiency controller.

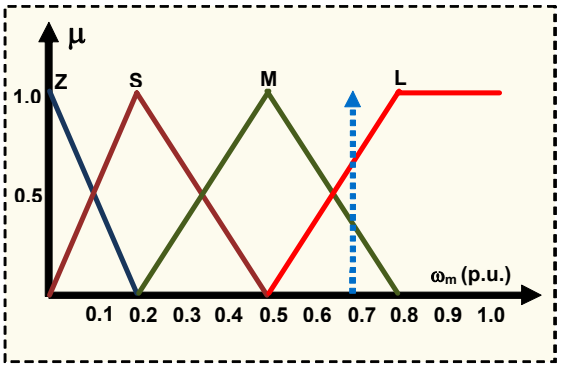
3.1 The Fuzzy Efficiency Controller

The fuzzy sets for the input variables are shown in Figure 5. Both utilize normalized universes of discourses, to make the controller easier to port for different machine ratings. The output variable ($i_{ds}'^*$) is represented by singletons, and is not shown here. The rule base for the fuzzy controller is illustrated in

Table 1. It is typically initialized with rated $i_{ds}'^{*}$ (1 p.u.), and it is progressively updated to incorporate the knowledge of the maximum efficiency points as they are found by the search controller, as previously described.



(a)



(b)

Figure 5: Fuzzy sets for the input variables: (a) load torque and (b) speed.

Table 1: Rule base for the fuzzy controller.

T_L \ ω_m	Z	S	M	L
Z	1	1	1	1
S	1	1	$I_{A(n+1)}$	$I_{B(n+1)}$
M	1	1	$I_{C(n+1)}$	$I_{D(n+1)}$
L	1	1	1	1

The primary flux reference current $i_{ds}'^{*}$ is obtained by fuzzy sup-min inference, and the height method of defuzzification:

$$i_{ds}'^{*} = \frac{\sum_{i=A}^{D} I_{i(n+1)} \times \mu_{Ri}}{\sum_{i=A}^{D} \mu_{Ri}} \qquad (3)$$

At steady state condition, whenever the search controller identifies an optimum flux level, the rule base must be updated. This process can be summarized as follows:

1) Identify the fired rules in the Rule Base (e.g., rules A,B,C,D in Table 1);

2) Compute the degree of truth for each rule, by applying the minimum (min) operator over the degree of membership for the input variables T_L and ω_r: $\mu_{Ri} = \min(\mu_{Tl}, \mu_{\omega r})$;

3) Evaluate the proportionality factor K, given by (4);

$$K = \frac{\sum_{i=A}^{D} \mu_{Ri} \times \sum \Delta i_{ds}(pu)}{\sum_{i=A}^{D} \mu_{Ri}^{2}} \qquad (4)$$

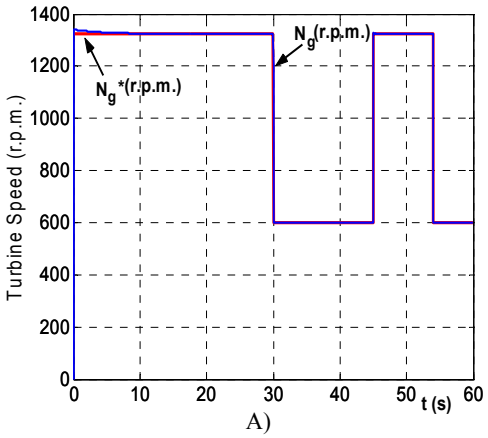
4) Compute the correction term $\Delta I_i(n)=K x \mu_{Ri}$ for each fired rule as the product of its degree of truth and factor K;

5) Get the new value for each fired rule i (i=A,B,C,D) by (5).

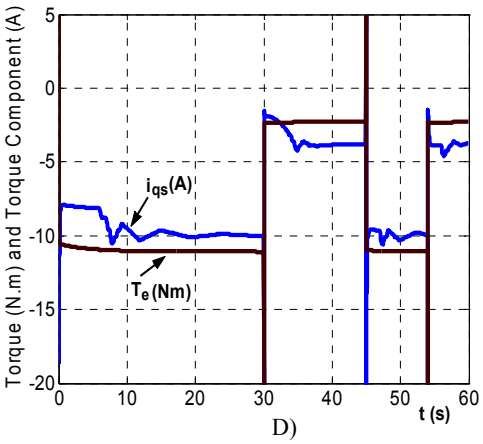
$$I_i(n+1) = I_i(n) + \Delta I_i(n) \qquad (5)$$

3 SIMULATION RESULTS

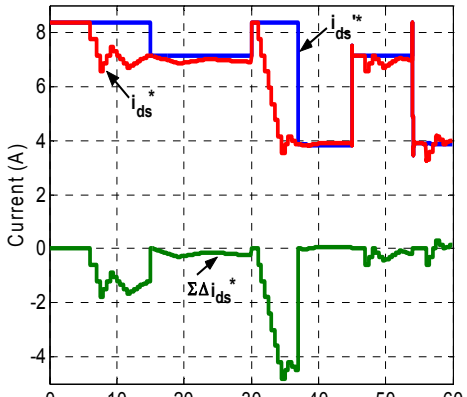
In this simulation a 4 kW squirrel-cage induction machine was used (220/380V, p=2). A reference speed step of 0.88 p.u, 0.40 p.u, 0.88 p.u and 0.40 p.u is applied at t=1s, t=30s, t=45s and t=55s, respectively, as shown in Figure 6 (A). After the initial transient, at t=6s, the search begins. At t = 15s the controller identifies that an optimum point has been found, and proceeds to update the rule base. Up to this point, the output of the fuzzy controller ($i_{ds}'^{*}$) was the rated value for magnetizing current, but from this time on, its output is made equal to the optimum value. Simultaneously, the output of the search controller is reset ($\sum \Delta i_{ds} = 0$), as can be seen in Figure 6 (B). The rotor flux response follows a first order filter profile of the reference current (i_{ds}^{*}) as expected, and is shown here multiplied by a factor of 10, Figure 6 (C). The changes in flux level have a direct impact on the output power, Figure 6 (E), as well as in the torque component of stator current reference (i_{qs}^{*}), as expected, but the electromagnetic torque is unaffected, due to proper feed-forward compensation in i_{qs}^{*}, as shown in Figure 6 (D). At t=45s and t=55s when a previous reference speed step is applied, the rule base recognize and the system operate directly with optimum efficiency. Figures 6 (F) and (G) shown the voltages and currents on the phases A, B and C, respectively.


Turbine Speed (r.p.m.)
N_g*(r.p.m.)
N_g(r.p.m.)
t (s)

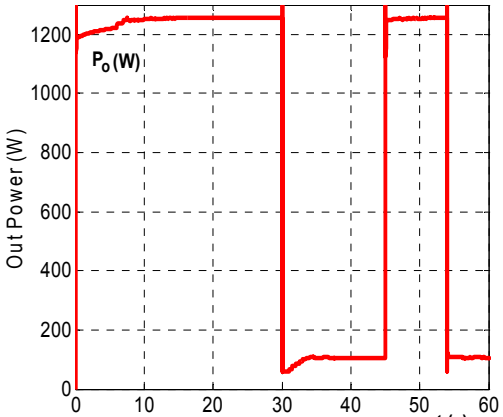
A)


Torque (N.m) and Torque Component (A)
i_{qs}(A)
T_e(Nm)
t (s)

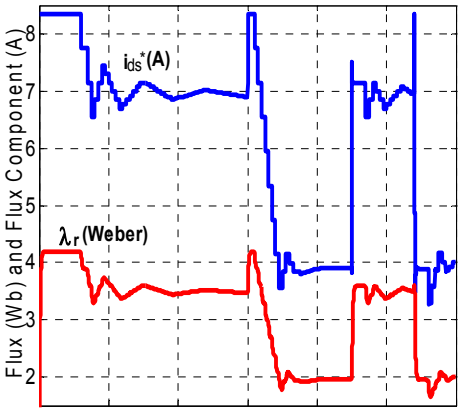
D)


Current (A)
i_{ds}*
i_{ds}*
$\Sigma\Delta i_{ds}$*
t (s)

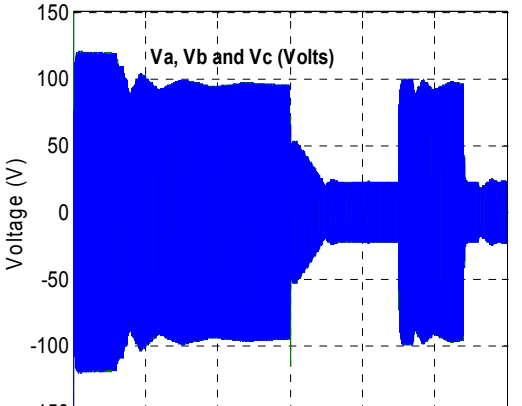
B)


Out Power (W)
P_o(W)
t (s)

E)


Flux (Wb) and Flux Component (A)
i_{ds}*(A)
λ_r(Weber)
t (s)

C)


Voltage (V)
Va, Vb and Vc (Volts)
t (s)

F)

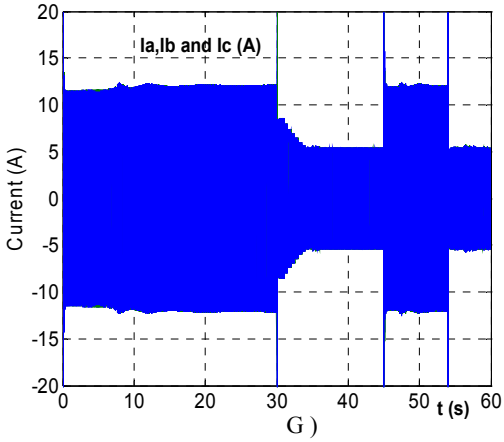


Figure 6: Results of simulation for operation of the fuzzy controller: A) Reference and actual speed on the axis of turbine in r.p.m.; B) Components of *d-axis* current; C) Components of *d-axis* current and rotor flux; D) Components of *q-axis* current and torque estimate; E) Out power; F) Voltage in the phases A, B and C; F) Current in the phases A, B and C.

4 CONCLUSIONS

The proposed control strategy consists of a more effective way to implement the efficiency optimization via flux control in an induction machine. From the analysis of literature can be observed that in the comparative study of the diverse known search techniques, none of them results on a process as fast as the one achieved with the present method. This implies a great energy save, because the system can be tuned all the time and operate at maximum efficiency. Another noteworthy point is that the transition from steady to transitory state occurs without abrupt changes in the system, or without any topological control change, since the displacement from a low torque point to one of higher torque or vice-versa is already programmed in the base of rules. The salient features of this technique are summarized next: i) It is applicable to any machine size, and does not require knowledge of machine parameters; ii) The rule base self tuning is progressive, and does not need any intervention from the operator; iii) Once tuned, the system is capable of operating all the time at optimum efficiency, without delay from one steady state condition to another, with significant energy saving; iv) During transients the rule base is kept active and, as a consequence there is no switching from one control strategy (for steady state) to another (during transients), provided that the tuning has been completed; v) Proper disturbance compensation is included, such that no correction is needed to keep torque and speed constant during the optimization process; and vi) The system is capable of tracking slow parameter deviations, guaranteeing true optimum efficiency.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support of Fha – Foundation for the Development of New Technologies of Hydrogen in Aragón; CIRCE – Centre of Research for Energy Resources and Consumption; as well as CEFET-BA and University of Zaragoza.

REFERENCES

Kumar,Vinod and Joshi, R. R. 2007. Fuzzy Logic Based Light Load Efficiency Improvement of Matrix Converter Based Wind Generator System. In *Journal of Theoretical and Applied Information Thecnology, pp. 79 – 89.* JATIT.

Leidhold, Roberto, Garcia, Guillermo and Valla, Maria Inés. February 2002. Field-Oriented Controller Induction Generator With Loss Minimization. *Transactions on Industrial Electronics. Vol. 49. No 1 pp. 147-156, February 2002.* IEEE.

Sousa, G. C. D. and B. K. Bose, (1995), Fuzzy logic based on-line efficiency optimization control of an indirect vector controlled induction motor drive, *Transactions on Industrial Electronics vol. 42 nº 2, pp. 192-198.* IEEE.

Rüncos, F. at alli, Geração de Energia Eólica - *Tecnologias Atuais e Futuras. WEG Jaraguá do Sul. Brasil.* www.weg.com.br. WEG Máquinas S.A.

Simões, Marcelo Godoy, Franceschetti, N. N. and BOSE, Bimal Kumar. 1999. Otimização de um Sistema de Geração de Energia Eólica Através de Controle Fuzzy. *SBA Controle & Automação. Vol. No 01, pp. 48-58. Brasil. Jan., Fev., Mar., Abril de 1999*. SBA.

Sousa, G. C. D., Bose, B. K. and J. Cleland, (1992), "Loss Modeling of Converter Induction Machine System for Variable Speed Drive", *Proceedings of the IECON'92, International Conference on Industrial Electronics, Control and Instrumentation, vol. 1, pp. 114-120.* IECON.

F. Abrahamsen, Fred Blaabjerg,J. K. Pedersen, P. Z. Grabowski and P. Thogersen, (1998), On the Energy Optimized Control of Standard and High-Efficiency Induction Motors in CT and HVAC Applications, *Transactions on Industry Applications, vol. 34, nº 4, pp. 822-831.* IEEE.

F. Abrahamsen, Fred Blaabjerg,J. K. Pedersen, (1996), State –of-the-Art of optimal Efficiency Control of Low Cost Induction Motor Drives, *Power Electronics and Motion Control, vol. 2, pp. 163-170.* PEMC'96.

J. Moreno-Eguilaz, Miguel Cipolla, Juan Peracula, Paulo J. da Costa Branco, (1997), Induction Motor Optimum Flux Search Algorithms with Transient State Loss Minimization Using a Fuzzy Logic Based Supervisor, Power Electronics Specialists Conference, 28th, Annual IEEE, vol. 2, pp. 1302 – 1308.

Souza, Durval de Almeida, Aragão Filho, W. C. P. and Sousa, G. C. D. August 2007. Adaptive Fuzzy Controller for Efficiency Optimization of Inductions Motors. *Transactions on Industrial Electronics, Vol 54, No 4, pp2157-2164.* IEEE.

Leidhold, Roberto, Garcia, Guillermo and Valla, Maria Inés. 2002. Control para Máximo Rendimiento de Generadores Eólicos de Velocidad Variable, con Limitación de Velocidad y Potencia. *XVI Congresso Brasileiro de Automática. Natal, RN. 2 a 5 de Setembro de 2002.pp. 3121-3126.* CBA.

Deprez, Wim at alli. 2007. Energy Efficiency of Small Induction Machines: Comparasion Between Motor and Generator Mode. *European Copper Institute. Vol. 3, pp 1-6.ISSUE 1. Digest of 11/01/2007.* Leonardo Energy.

EXPONENTIAL OBSERVER FOR A CLASS OF NONLINEAR DISTRIBUTED PARAMETER SYSTEMS WITH APPLICATION TO A NONISOTHERMAL TUBULAR REACTOR

Nadia Barje, Mohammed Achhab
Laboratoire d'Ingénierie Mathematique (LINMA), Université Chouaib Doukkali, El Jadida, Morroco
nbarje@yahoo.fr, achhab@ests.ucam.ac.ma

Vincent Wertz *
CESAME, Universit de Louvain, Louvain-la-Neuve, Belgium
vincent.wertz@uclouvain.be

Keywords: Nonlinear infinite-dimensional systems, state observer, nonlinear tubular reactor.

Abstract: This paper present sufficient conditions to construct an exponential state estimator for a class of infinite dimensional non-linear systems driven in a real Hilbert state description. The theory is applied to a nonisothermal plug flow tubular reactor, governed by hyperbolic first order partial differential equations. For this application performance issues of the exponential state estimator design are illustrated in a simulation study.

1 INTRODUCTION

State estimators for dynamical systems have been the focus of an intensive work in the last decades. The classical theory of the Luenberger observers has been successfully extended from finite dimensional linear systems to a large class of infinite dimensional linear systems by many authors, since the pioneering paper by (Gressang and Lamont, 1975). Later, the theory has been generalized to a class of single input distributed bilinear systems in (Gauthier et al., 1995). The paper of (Bounit and Hammouri, 1997) consider a class of distributed bilinear systems witch are observable for "small inputs" and gives a strong exponential observer. Recently, for nonlinear models of non-isothermal tubular reactors considered in (Laabissi et al., 2001), the paper of (Orlov and Dochain, 2002) presented a reduced-order observer of the concentration, assuming that the temperature is the only available measurement.
The primary objective of this paper is to address the problem of the design of exponential Luenberger-like observers for a class of infinite dimensional nonlinear systems described by the following equation

$$\begin{cases} \dot{x}(t) = Ax(t) + N(x(t)), x(0) \in D(A) \cap D \\ y(t) = Cx(t) \end{cases} \quad (1)$$

Here, A is the infinitesimal generator of a C_0-semigroup on a real Hilbert space H with inner product $< .,. >$ and norm $\| . \|$, $D(A)$ is the domain of A, N is a nonlinear operator from a closed subset D of H into H, $y(t) \in Y$ is the known output function associated to the unknown initial condition $x(0)$, Y is another real Hilbert space and C is a bounded linear operator from H into the Hilbert space Y. Under the assumption that N is locally Lipschitz continuous, it is shown in ((Pazy, 1983), pp. 185-186) that equation (1) has a unique mild solution on some interval $[0, t_{max}),\ t_{max} \in (0, +\infty]$ given by

$$\begin{cases} x(t) = S(t)x(0) + \int_0^t S(t-s)N(x(s))ds, \\ 0 \leq t < t_{max} \end{cases} \quad (2)$$

where $(S(t))_{t \geq 0}$ denotes the C_0-semigroup generated by A. To ensure that the problem is well posed, we shall assume throughout the paper as in (Laabissi et al., 2001) that we have: $t_{max} = +\infty$. An observer design is presented for which a result about the exponential convergence of the estimation error is stated under verifiable conditions.
Our second objective is to apply the previous developed result to the nonlinear model of a chemical plug flow reactor. This model is showed to be described

*This work was finalized when this author was on sabbatical leave at the ARC Centre of Excellence for Complex Dynamic Systems and Control, The University of Newcastle, Australia, whose support is gratefully acknowledged.

by (1). Trajectory analysis of such a model of chemical plug flow reactors has been done extensively in (Achhab et al., 1999) and (Laabissi et al., 2001). For this application, we also introduce a second observer in the case when only one of the two states, namely the temperature, is measured and show the exponential convergence of both estimation errors. A third observer is then introduced to improve the convergence rate of the previous one. Simulations results are then presented in order to highlight the performance issues of the proposed observers.
The paper is organized as follows. In section 2, we consider a general observer design for system (1). Then, we state sufficient conditions under which the related estimation error converges exponentially to zero. The approach developed in the general setting is applied to a chemical plug flow reactor model in section 3. In section 4, simulation results are given in order to illustrate some performance issues of this application. Finally, the paper closes with some remarks and conclusions in section 5. The background of the approach is to be found in (Curtain and Zwart, 1995) and (Cazenave and Haraux, 1998).

2 OBSERVER DESIGN

We state in this section sufficient conditions under which we will be able to show that the estimation error of the Luenberger-like observer converges exponentially to zero.
Let us assume that the following.

A.1. The linear operator A satisfies for all $x \in D(A)$, and $t \geq 0$, $< Ax(t), x(t) > \leq 0$.

A.2. The nonlinear operator N is a k_N-Lipschitz operator on its domain D, where k_N is a positive constant; i.e. for all $x, y \in D$, $\| N(x) - N(y) \| \leq k_N \| x - y \|$.

A.3. The pair (A, C) is approximately observable linear system (i.e. $\forall e \in H$, $\{CS(t)e = 0, \forall t \geq 0, \text{ implies } e = 0\}$), exponentially stable.

A.4. The semigroup $S(.)$ satisfies for all $e \in H$:

$$< S^*(.)C^*CS(.)e, e > \leq \| S(.) \|^2 < C^*Ce, e >,$$

Comment 2.1. *The hypothesis A.3 implies that the linear system*

$$\begin{cases} \dot{x}(t) = Ax(t), x(0) \in D(A) \\ y(t) = Cx(t) \end{cases}$$

is approximately observable on $[0, +\infty)$ *and that the observability gramian* $L_C := C^*C$, *where* $Ce := CS(.)e$ *for all* $e \in H$, *and* C^* *is the adjoint of the linear operator* C, *is bounded positive definite (see (Curtain and Zwart, 1995), p.160), and thus has an algebraic bounded inverse with domain equal to range* L_C.

Consider now the following candidate observer

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + N(\hat{x}(t)) - GC^*(C\hat{x}(t) - y(t)) \\ \hat{x}(0) \in D(A) \cap D \end{cases} \quad (3)$$

where G is a linear bounded operator and $y(t)$ is the known output function of the system (1). One can show that system (3) admits a unique solution $\hat{x}(t)$ which is well defined for any initial condition $\hat{x}(0) \in D(A) \cap D$ and for all $t \in [0, t_{max})$, with t_{max} assumed to be equal $+\infty$.
Setting $e(t) = x(t) - \hat{x}(t)$, the reconstruction error $e(t)$ obeys the following equation:

$$\dot{e}(t) = Ae(t) + N(x(t)) - N(\hat{x}(t)) - GC^*Ce(t) \quad (4)$$

and one obtains the following theorem:

Theorem 2.1. *Let assumptions A.1-A.4 be satisfied. If there exists a bounded linear operator* G *and a positive real number* g *such that* $g > k_N$ *and for* $e \in H$, $e \neq 0$,

$$< GC^*Ce, e > \geq < g \| L_C^{-1} \| \| S(.) \|^2 C^*Ce, e >$$

then, system (3) *is an exponential observer for system* (1). *More precisely, the reconstruction error satisfies* $\| e(t) \|^2 \leq \| e(0) \|^2 e^{-\eta t}$ *where* $\eta = 2(g - k_N)$.

Proof 2.1. *The computation of the derivative of the functional*

$$V_e(t) = \frac{1}{2} \| e(t) \|^2$$

along the trajectories of (4) *yields,*

$$\begin{aligned} \dot{V}_e(t) &= < \dot{e}(t), e(t) > \\ &= < Ae(t), e(t) > + < N(x(t)) - N(\hat{x}(t)), e(t) > \\ &\quad - < GC^*Ce(t), e(t) > \end{aligned}$$

and in addition,

$$\begin{aligned} < GC^*Ce, e > &\geq g \| L_C^{-1} \| \| S(.) \|^2 < C^*Ce, e > \\ &\geq g \| L_C^{-1} \| < L_C e, e > \\ &\geq g < e, e > \end{aligned}$$

indeed, the operator L_C *is self-adjoint and nonnegative (i.e,* $< L_C e, e > \geq 0$ *for all* $e \in H$*), then* L_C *has a unique square root* $L_C^{\frac{1}{2}}$ *self-adjoint, so that* $L_C^{\frac{1}{2}} L_C^{\frac{1}{2}} e = L_C e$ *for all* $e \in H$ *(see (Curtain and Zwart, 1995), p.606), the operator* L_C^{-1} *is also self-adjoint and nonnegative, par consequent has a unique square root* $(L_C^{-1})^{\frac{1}{2}} = (L_C^{\frac{1}{2}})^{-1}$ *(see (Curtain and Zwart, 1995), pp. 603-610).*
and in addition, for all $e \in H$,

$$< L_C^{-1} e, e > \quad \leq \| L_C^{-1} \| < e, e >$$

thus,

$$\begin{aligned} \| L_C^{-1} \| < L_C e, e > &= \| L_C^{-1} \| < L_C^{\frac{1}{2}} L_C^{\frac{1}{2}} e, e > \\ &= \| L_C^{-1} \| < L_C^{\frac{1}{2}} e, L_C^{\frac{1}{2}} e > \\ &\geq < L_C^{-1} L_C^{\frac{1}{2}} e, L_C^{\frac{1}{2}} e > \\ &\geq < (L_C^{\frac{1}{2}})^{-1} L_C^{\frac{1}{2}} e, (L_C^{\frac{1}{2}})^{-1} L_C^{\frac{1}{2}} e > \\ &= < e, e > \end{aligned}$$

Hence,

$$\begin{aligned} \dot{V}_e(t) &\leq \| N(x(t)) - N(\hat{x}(t)) \| \| e(t) \| - g \| e(t) \|^2 \\ &\leq (k_N - g) \| e(t) \|^2 = -\eta V_e(t) \end{aligned}$$

Now, using Gronwall's Lemma (see (Curtain and Zwart, 1995), p. 639), we get

$$V_e(t) \leq V_e(0) e^{-\eta t}$$

Consequently, one may deduce

$$\| e(t) \|^2 \leq \| e(0) \|^2 e^{-\eta t}$$

This shows the exponential convergence of the estimation error and the proof of Theorem (2.1) is thus complete.

2.1 Application to a Nonisothermal Plug-Flow Reactor

The theory developed in the general setting is applied to a chemical non-isothermal tubular reactor with the following chemical reaction:

$$A \rightarrow B$$

The kinetics of the above reaction is characterized by first-order kinetics with respect to the reactant concentration $C(mol/l)$ and by an Arrhenius-type dependence with respect to the temperature $T(K)$, and the dynamics of the process are described by the following two energy and mass balance PDEs (see (Laabissi et al., 2001)):

$$\frac{\partial T}{\partial \tau} = -\upsilon \frac{\partial T}{\partial \zeta} - \frac{4h}{\rho C_p d}(T - T_c) - \frac{\Delta H}{\rho C_p} k_0 C e^{\frac{-E}{RT}}, \quad (5)$$

$$\frac{\partial C}{\partial \tau} = -\upsilon \frac{\partial C}{\partial \zeta} - k_0 C e^{\frac{-E}{RT}}, \quad (6)$$

where the boundary conditions are given, for $\tau \geq 0$, by:

$$T(0,\tau) = T_{in},\ C(0,\tau) = C_{in} \quad (7)$$

and the initial conditions are assumed to be given, for $0 \leq \zeta \leq L$, by:

$$T(\zeta,0) = T_0(\zeta),\ C(\zeta,0) = C_0(\zeta) \quad (8)$$

In the equations above, the following parameters υ, ΔH, ρ, C_p, k_0, E, R, h, d, T_c hold for the superficial fluid velocity, the heat of reaction, the density, the specific heat, the kinetic constant, the activation energy, the ideal gas constant, the wall heat transfer coefficient, the reactor diameter, the coolant temperature. T_{in} and C_{in} are respectively the inlet temperature and the inlet reactant concentration which will be assumed to be two positive constants. τ, ζ and L denote the time and space independent variables, and the length of the reactor, respectively. Finally T_0 and C_0 denote the initial temperature and reactant concentration profiles.

The dynamics will be described by means of an infinite-dimensional system derived from an equivalent nonlinear PDE dimensionless model. Such an approach is standard in tubular reactor analysis (see (Laabissi et al., 2001)) and is briefly developed here. Let here after $H = L^2[0,L] \times L^2[0,L]$, endowed by the inner product

$$< (x_1,x_2),(y_1,y_2) > = < x_1,y_1 >_{L^2} + < x_2,y_2 >_{L^2}$$

and the induced norm

$$\| (x_1,x_2) \| = (\| x_1 \|_{L^2}^2 + \| x_2 \|_{L^2}^2)^{\frac{1}{2}}$$

for all (x_1,x_2) and (y_1,y_2) in H. We will denote here after $<,>_{L^2}$ by $<,>$.

Consider the following dimensionless state variables:

$$x_1 = \frac{T - T_{in}}{T_{in}}, \quad x_c = \frac{T_c - T_{in}}{T_{in}},$$
$$x_2 = \frac{C_{in} - C}{C_{in}}, \quad r(x_1) = e^{\frac{\mu x_1}{1+x_1}}$$

Let us consider also dimensionless time t and space z variables:

$$t = \frac{\tau \upsilon}{L},\ z = \frac{\zeta}{L}.$$

We shall assume in the rest of the paper that the coolant temperature T_c is equal to the inlet temperature T_{in} (i.e $x_c \equiv 0$), since x_c will be eliminated in the equation of the reconstruction error between the plan state and the observer state.
Then we obtain the following equivalent representation of the model (5)-(8):

$$\frac{\partial x_1}{\partial t} = -\frac{\partial x_1}{\partial z} - \beta x_1 + \alpha\delta(1 - x_2) r(x_1) \quad (9)$$

$$\frac{\partial x_2}{\partial t} = -\frac{\partial x_2}{\partial z} + \alpha(1 - x_2) r(x_1) \quad (10)$$

with the boundary conditions:

$$x_1(z = 0,t) = 0, \quad x_2(z = 0,t) = 0 \quad (11)$$

and initial conditions

$$x_1(z,0) = x_1^0, \quad x_2(z,0) = x_2^0 \tag{12}$$

and the parameters α, β, δ and μ are related to the original parameters as follows:

$$\mu = \frac{E}{RT_{in}}, \quad \alpha = \frac{k_0 L}{\upsilon} exp(-\mu)$$
$$\beta = \frac{4hL}{\rho C_p d\upsilon}, \quad \delta = -\frac{\Delta H}{\rho C_p}\frac{C_{in}}{T_{in}}.$$

From a physical point of view it is expected that for all $z \in [0,1]$, and for all $t \geq 0$ (see (Aksikas et al., 2007)),

$$0 \leq T(z,t) \leq T_{max} \text{ and } 0 \leq C(z,t) \leq C_{in}$$

or equivalently

$$-1 \leq x_1(z,t) \leq \frac{T_{max} - T_{in}}{T_{in}} \text{ and } 0 \leq x_2(z,t) \leq 1,$$

where T_{max} could possibly be equal to $+\infty$.
This is also true for the model, as shown by (Laabissi et al., 2001).
The equivalent state space description of the model (9)-(12) is given by the following nonlinear abstract differential equation on the Hilbert space $H = L^2[0,1] \times L^2[0,1]$:

$$\begin{cases} \dot{x}(t) = Ax(t) + N(x(t)) \\ x(0) = x_0 \in D(A) \cap D \end{cases} \tag{13}$$

where, A is the linear operator defined on its domain

$$D(A) := \{x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \in H : x \text{ absolutely continuous, } \frac{dx}{dz} \in H \text{ and } x_i(0) = 0, i = 1,2\}$$

by,

$$Ax := \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -\frac{d.}{dz} - \beta. & 0 \\ 0 & -\frac{d.}{dz} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

The linear operator A satisfies,

$$< Ax, x > \leq -\frac{1}{2}x^2(1), \text{ for all } x \in D(A)$$

which satisfies the hypothesis A.1.
A is the generator of a C_0-semigroup exponentially stable

$$S(t) = \begin{pmatrix} S_1(t) & 0 \\ 0 & S_2(t) \end{pmatrix}$$

satisfying (see (Winkin et al., 2000)), for all $(x_1, x_2) \in H$,

$$(S_1(t)x_1)(z) = \begin{cases} e^{-\beta t}x_1(z-t) & \text{if } z \geq t, \\ 0 & \text{if } z < t, \end{cases}$$

$$(S_2(t)x_2)(z) = \begin{cases} x_2(z-t) & \text{if } z \geq t, \\ 0 & \text{if } z < t, \end{cases}$$

Moreover, (see (Winkin et al., 2000))

$$\| S(t) \| \leq 1 \text{ for all } t \geq 0$$

The nonlinear operator N is defined on

$$D := \{x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \in H : -1 \leq x_1(z) \text{ and } 0 \leq x_2(z) \leq 1, \text{ for almost all } z \in [0,1]\} \text{ by,}$$

$N(x) = (N_1(x), N_2(x))^T$ where for all $x = (x_1, x_2)^T \in D$,

$$N_1(x) = \alpha\delta(1 - x_2)r(x_1)$$
$$N_2(x) = \alpha(1 - x_2)r(x_1)$$

It is proved in (Aksikas et al., 2007) that the function $m(s) := exp(\frac{-k}{s})$ where $k = \frac{E}{R}$, is a Lipschitz continuous function on $[0, T_{max}]$ with a Lipschitz constant l_s given by

$$l_s = \begin{cases} 4\frac{R}{Ee^2} & \text{if } E \leq 2RT_{max} \\ \frac{E}{RT_{max}^2} exp(-\frac{E}{RT_{max}}) & \text{if } E > 2RT_{max} \end{cases}$$

It follows that the constant $k_r := e^{\mu} T_{in} l_s$ is a Lipschitz constant for the function $r(s) := exp(\frac{\mu s}{1+s})$.
We prove that for all

$$x := (x_1, x_2)^T \text{ and } y := (y_1, y_2)^T \in D,$$
$$\| N_2(x) - N_2(y) \| \leq \alpha exp(\mu) \| x_2 - y_2 \| + \alpha k_r \| x_1 - y_1 \|,$$

Observe that $N_1 = \delta N_2$, thus we take $k_N := \alpha(exp(\mu) + k_r)(1 + | \delta |)$ as a Lipschitz constant of N, the hypothesis A. 2 is thus satisfied.
It is proved in (Laabissi et al., 2001) that the system Eq. 13 has a unique mild solution $x(t, x(t=0))$ on $[0, +\infty[$, for all $x_0 \in D$ and that the state remains in D.
Hereafter we consider measurements at the reactor output. In this case, the output function $y(.)$ is defined as follows: we consider a (very small) finite interval at the reactor output $[1-w, 1]$ such that:

$$y(t) = (Cx)(t) := \int_0^1 \chi_{[1-w,1]}(a) I_{2\times 2} x(a,t) da, \ \forall t \in \mathbb{R}^+$$

where

$$\chi_{[1-w,1]}(a) = \begin{cases} 1, & \text{if} \quad a \in [1-w, 1] \\ 0, & \text{elsewhere.} \end{cases}$$

and $I_{2\times 2}$ is either the 2×2 identity matrix operator when both components x_1 and x_2 are measured, or a unit row vector if only one of them is measured. In the first case (i.e. two measurements), it is proved

in (Winkin et al., 2000), that the pair (C,A) is approximately observable if both x_1 and x_2 are measured at the reactor output, thus hypothesis A.3 is satisfied and so the observability gramian $L_C := C^*C$, where $Cx := CS(.)x$ for all $x \in H$ is positive definite and has an algebraic inverse L_C^{-1} with domain equal to *range* L_C, satisfying for $x_d(z,t) = I_d(z,t)$, where $I_d(z,t) = 1$ for all $(z,t) \in [0,1] \times \mathbb{R}^+$:

$$\begin{aligned} < L_C x_d, x_d > &= < CS(.)x_d, CS(.)x_d > \\ &\geq w^2 e^{-2\beta} \parallel x_d(z) \parallel^2 \end{aligned}$$

on have

$$\parallel L_C \parallel^2 \geq w^2 e^{-2\beta} \parallel L_C \parallel$$

and

$$< L_C x, L_C x > \geq w^2 e^{-2\beta} < L_C x, x >$$

Observe that L_C is self-adjoint and for all $y \in$ *range* L_C,

$$\begin{aligned} < L_C^{-1} y, y > &= < L_C^{-1} y, L_C L_C^{-1} y > \\ &\leq \tfrac{1}{w^2 e^{-2\beta}} < L_C L_C^{-1} y, L_C L_C^{-1} y > \\ &\leq \tfrac{1}{w^2 e^{-2\beta}} < y, y > \end{aligned}$$

this implies,

$$\parallel L_C^{-1} \parallel \leq \frac{1}{w^2 e^{-2\beta}}$$

Denote $w_0 = w^2 e^{-2\beta}$

A candidate Luenberger-observer for system (9)-(12) when the state variables are measured is

$$\begin{aligned} \frac{\partial \hat{x}_1}{\partial t} = & -\frac{\partial \hat{x}_1}{\partial z} - \beta \hat{x}_1 + \alpha\delta(1-\hat{x}_2) r(\hat{x}_1) + \\ & \tfrac{g}{w_0} C_1^* C_1 e_1 \end{aligned} \quad (14)$$

$$\frac{\partial \hat{x}_2}{\partial t} = -\frac{\partial \hat{x}_2}{\partial z} + \alpha(1-\hat{x}_2) r(\hat{x}_1) + \tfrac{g}{w_0} C_2^* C_2 e_2 \quad (15)$$

with the boundary conditions:

$$\hat{x}_1(z=0,t) = 0, \quad \hat{x}_2(z=0,t) = 0 \quad (16)$$

and initial conditions

$$\hat{x}_1(z,t=0) = \hat{x}_1^0, \quad \hat{x}_2(z,t=0) = \hat{x}_2^0 \quad (17)$$

with $C = \begin{pmatrix} C_1 & C_2 \end{pmatrix}$ and $e_i(z,t) = x_i(z,t) - \hat{x}_i(z,t)$ for $i = 1,2$, for all $(z,t) \in [0,1] \times \mathbb{R}^+$.

The observer state remains in the set D, the main steps of the proof go along the line of the one given in (Laabissi et al., 2001).

Observe that the model (14)-(17) is in the form of the nonlinear abstract differential equation (3), with the linear operator G chosen as follows: $G = \frac{g}{w_0} I$ where I is the identity operator and g is a positive real number.

For the bounded operator C given above, the C_0-semigroup $(S(t))_{t\geq 0}$ satisfies for all $e = (e_1, e_2)^T \in H$:

$$< S(.)C^*CS(.)e, e > \leq \parallel S(.) \parallel^2 < C^*Ce, e >$$

which satisfies the hypothesis A.4.

Denote $e(t)$ the reconstruction error between the plant state and the observer state. A direct application of Theorem 2.1 yields the following result.

Corollaire 2.1. *Take g such that $g > k_N$ holds. Then, system* (14)-(17) *is an exponential observer for system* (9)-(12)*. More precisely, the reconstruction error $e(t)$ has the property that $\parallel e(t) \parallel^2 \leq \parallel e(0) \parallel^2 e^{-\eta t}$, where $\eta = 2(g - k_N)$.*

It is also interesting to examine the case where only the temperature $x_1(z,t)$ is measured at the output of the reactor. So, the observation operator $C = \begin{pmatrix} C_1 & C_2 \end{pmatrix}$ is given by

$$\begin{aligned} (C_1 e_1)(t) &= \int_0^1 \chi_{[1-w,1]}(a) e_1(a,t) da, \ \forall t \in \mathbb{R}^+ \\ C_2 &= 0 \end{aligned}$$

Recall that the linear operator A is diagonal. The operators A_1 and A_2 satisfy;

$$< A_i x_i, x_i > \leq -\frac{1}{2} x_i^2(1), \text{ for all } x_i \in D(A_i) \text{ and } i = 1,2$$

on their common domain:

$$\begin{aligned} D(A_{i=1,2}) = \{ & x \in L^2[0,1] : x \text{ absolutely continuous}, \\ & \tfrac{dx}{dz} \in L^2[0,1] \text{ and } x(0) = 0 \}. \end{aligned}$$

In the same manner as in (see (Winkin et al., 2000)), we prove that (C_1, A_1) is approximately observable.

In this case, a full-order observer for the dimensionless model (9)-(12) can be constructed as follows:

$$\begin{aligned} \frac{\partial \hat{x}_1}{\partial t} = & -\frac{\partial \hat{x}_1}{\partial z} - \beta \hat{x}_1 + \alpha\delta \sup_{x_2 \in D_2} (1-x_2) r(\hat{x}_1) + \\ & \tfrac{g}{w_0} C_1^* C_1 e_1 \end{aligned} \quad (18)$$

$$\frac{\partial \hat{x}_2}{\partial t} = -\frac{\partial \hat{x}_2}{\partial z} + \alpha(1-\hat{x}_2) \sup_{x_1 \in D_1} r(x_1) \quad (19)$$

with the boundary conditions:

$$\hat{x}_1(z=0,t) = 0, \ \hat{x}_2(z=0,t) = 0 \quad (20)$$

and initial conditions

$$\hat{x}_1(z,t=0) = \hat{x}_1^0, \ \hat{x}_2(z,t=0) = \hat{x}_2^0 \quad (21)$$

Note that in this observer, the nonlinear term is not exactly taken as in the "true system", for technical reasons in the convergence proof.

It can be shown as in (Laabissi et al., 2001) that the observer states remains in the set D.

We state the following result:

Theorem 2.2. *Take g such that $g > k_r \alpha\delta$ holds. Then, system* (18)-(21) *is an exponential observer for the non-isothermal plug flow reactor model* (9)-(12)*. More precisely the reconstruction errors have the properties that $\parallel e_1(t) \parallel^2 \leq \parallel e_1(0) \parallel^2 e^{-\nu_1 t}$ and $\parallel e_2(t) \parallel^2 \leq \parallel e_2(0) \parallel^2 e^{-\nu_2 t}$, where $\nu_1 := 2(g - \alpha\delta k_r)$ and $\nu_2 = 2\alpha exp(\mu)$.*

The proof is similar to that given in Theorem 2.1.
The concentration error converges to zero with convergence rate ν_2 depending only on the internal dynamics of the process. It will be interesting to look for a "closed loop" observer design that will make ν_2 as large as desired. In this case however the full state will need to be observed. The following is given to improve the convergence rate of the concentration error.

To have an a priori given convergence rate of the concentration error $\nu_2^* \geq \nu_2$, one can use the following full order observer:

$$\frac{\partial \hat{x}_1}{\partial t} = -\frac{\partial \hat{x}_1}{\partial z} - \beta \hat{x}_1 + \alpha\delta \sup_{x_2 \in D_1}(1-x_2) r(\hat{x}_1) + \frac{m_1}{w_0} C_1^* C_1 e_1 \quad (22)$$

$$\frac{\partial \hat{x}_2}{\partial t} = -\frac{\partial \hat{x}_2}{\partial z} + \alpha(1-\hat{x}_2) \sup_{x_1 \in D_2} r(x_1) + \frac{m_2}{w_0} C_2^* C_2 e_2 \quad (23)$$

with the boundary conditions:

$$\hat{x}_1(z=0,t) = 0,\ \hat{x}_2(z=0,t) = 0 \quad (24)$$

and initial conditions

$$\hat{x}_1(z,t=0) = \hat{x}_1^0,\ \hat{x}_2(z,t=0) = \hat{x}_2^0 \quad (25)$$

where, for $i = 1,2$

$$(C_i e_i)(t) := \int_0^1 \chi_{[1-w,1]}(a) e_i(a,t) da,\ \forall t \in \mathbb{R}^+$$

where m_1 is a positive real number, $m_2 = \frac{\nu_2^* - \nu_2}{2}$. then we have the following result:

Theorem 2.3. *Consider the full-order observer* (22)-(25) *for the uncontrolled system* (9)-(12) *where* $m_1 > \alpha\delta k_r$. *Then the temperature error* $e_1(t)$ *satisfies* $\| e_1(t) \|^2 \leq \| e_1(0) \|^2 e^{-\nu_1 t}$, *where* $\nu_1 := 2(m_1 - \alpha\delta k_r)$, *and the concentration error* $e_2(t)$ *satisfies* $\| e_2(t) \|^2 \leq \| e_2(0) \|^2 e^{-\nu_2^* t}$, *with convergence rate* $\nu_2^* := 2(\alpha exp(\mu) + m_2)$, *larger than that of the full-order observer* (18)-(21).

In this section, we have thus described three different exponential observers for the plug flow reactor model. The first one (eq. (14)-(17)) is derived directly from our main result (Theorem 2.1). The second one (eq. (18)-(21)) shows that an exponential observer can be constructed even if the concentration is not measured and with only partial knowledge of the nonlinear part of the model. The third one (eq. (22)-(25)) improves the convergence rate of the concentration reconstruction error by reintroducing a measurement of the concentration.

Comment 2.2. *In (Aksikas et al., 2007), a result of asymptotic stability of the system Eq. 13 requires the following condition:*

$$k_N < \beta$$

In order to test the performance of the proposed observers, numerical simulations will be given when the above condition does not holds.

2.2 Simulation Results

Our objective is to illustrate the theoretical results related to the different exponential observers for the plug flow reactor model.
The process model has been initialized with two constant profiles $x_1(0,z) = -1$, and $x_2(0,z) = 0$. The observers have been initialized with $\hat{x}_1(0,z) = 0$, and $\hat{x}_2(0,z) = 1$. The equations have been integrated by using a backward finite difference approximation for the first-order space derivative $\partial/\partial z$

$$\frac{\partial x}{\partial z} \simeq \frac{x(t,z_i) - x(t,z_{i-1})}{\Delta z}$$

with $\Delta z = 1/100$.
In order to be close as possible to possibly unstable nonisothermal plug-flow reactor, we have selected the model (9)-(12) with $\beta = 0.2$. The adopted numerical values for the process parameters are taken from (Smets et al., 2002).

Table 1: Process parameters using for numerical simulations.

Process parameters	Numerical value
L	1 m
v	0.1 m/s
E	11250 cal/mol
k_0	$10^6 s^{-1}$
$\frac{4h}{\rho C_p d}$	$0.02 s^{-1}$
C_{in}	0.02 mol/L
R	1.986 cal/(mol.K)
T_{in}	340 K
$\frac{\Delta H}{\rho C_p}$	-4250 K.L/mol

Figure 1 shows the time evolution of the concentration error e_2 related to the exponential observer (14)-(17). Similar results are obtained for the two other observers.
In order to cover all the assumptions, the design parameter g related respectively to the exponential observer (14)-(17) and the exponential observer (18)-(21) has been taken respectively as $g = 2 * k_N$ and $g = 2 * \alpha\delta k_r$, and the design parameters m_1 and m_2 related to the exponential observer, (22)-(25) have

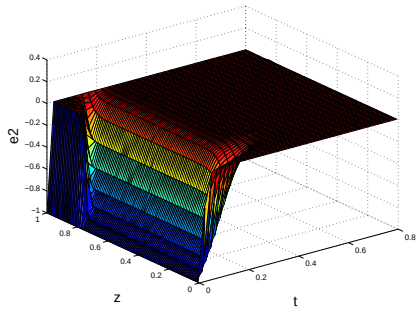
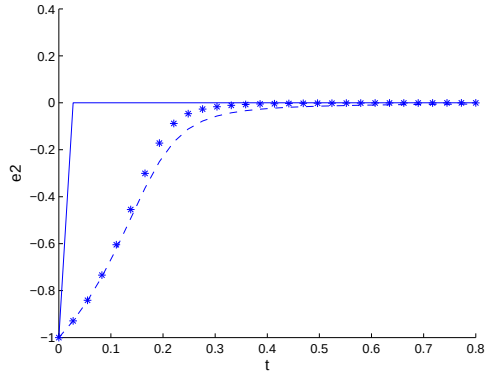


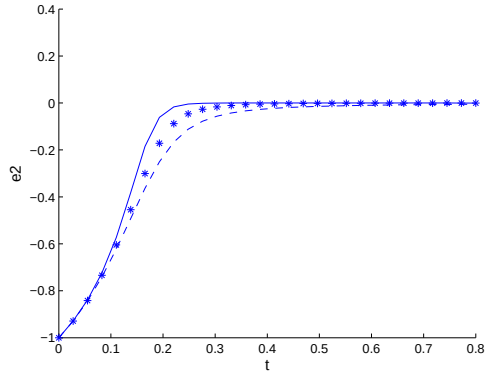
Figure 1: Evolution in time and space of the concentration error.

been taken as $m_1 = 2 * \alpha\delta k_r$ and $m_2 = 10 * m_1$, with $w = 3 * L/4$.

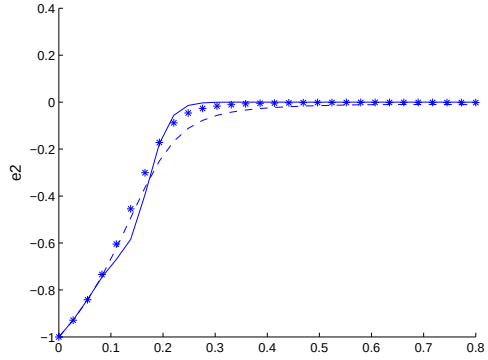
Figure 2 shows respectively the time evolution of the concentration error e_2 at the positions $0.1 * L$, $0.5 * L$ and $0.9L$, for the case where only the temperature is measured on the length interval $[3 * L/4, L]$ (the dashed line) i.e the exponential observer (18)-(21), and for the case where both the temperature and the concentration are measured on the same length interval with the exponential observer (14)-(17) (the solid line) and with the exponential observer (22)-(25) (the star line).
It is seen as expected that the concentration error related to the exponential observer (22)-(25) is faster than the one related to the exponential observer (18)-(21), however it remains slower than that related to the observer (14)-(17), which represents the ideal case, since in that case, the nonlinear part is assumed to be exactly known.



(a) concentration error at z=0.9*L.



(b) concentration error at z=0.5*L.



(c) concentration error at z=0.1*L.

Figure 2: Convergence of the concentration error for the three proposed observers.

3 CONCLUSIONS

This paper presents sufficient conditions to construct an exponential observer for a nonlinear infinite dimensional system driven in a real Hilbert state description. The theory is applied to a non-isothermal plug flow tubular reactor governed by hyperbolic first order partial differential equations. Several observer structures are proposed, depending on the part of the states that are available for measurement and on the knowledge of the nonlinear part of the model. Performance issues of the different observer designs are illustrated by simulation results. The best performance is obviously obtained when the nonlinear term is perfectly known and both states (temperature and concentration) are measured at the end of the reactor. However, we also show that good results can be achieved when only the temperature is measured and when bounds on the nonlinear term are used in the observer dynamics. Finally, an improved convergence rate for the estimation error on the concentration can be obtained when re-introducing a measurement of the concentration at the end of the reactor. These observers include design parameters that can be tuned by the user to satisfy specific needs in terms of convergence rate.

REFERENCES

Achhab, M. E., Laabissi, M., Winkin, J., and Dochain, D. (1999). State trajectory analysis of plug flow non-isothermal reactors using a nonlinear model. In *Proceedings of the 38th IEEE Conference on Decision and Control, Phoenix, Arizona, USA, 663-667, December*.

Aksikas, I., Winkin, J., and Dochain, D. (2007). Asymptotic stability of infinite-dimensional semilinear systems: application to a nonisothermal reactor. In *Systems and Control Letters, vol. 56, 122-132.*

Bounit, H. and Hammouri, H. (1997). Observers for infinite dimensional bilinear systems. European J. of Control, vol. 3, 325-339.

Cazenave, T. and Haraux, A. (1998). *An Introduction to Semilinear Evolution Equations*. Clarendon Press. Oxford.

Curtain, R. F. and Zwart, J. (1995). *An Introduction to Infinite Dimentional Linear Systems Theory*. Springer, New York.

Gauthier, J. P., Xu, C. Z., and Bounabat, A. (1995). Observers for infinite dimentional dissipative bilinear systems. In *J. Math. Systems, Estimation and Control, vol. 5, 119-122*.

Gressang, R. V. and Lamont, G. B. (1975). Observers for systems characterized by semigroups. In *IEEE Trans. Aut. Contr, vol. 20, 523-528.*

Laabissi, M., Achhab, M., Winkin, J., and Dochain, D. (2001). Trajectory analysis of nonisothermal tubular reactor nonlinear models. In *Systems and Control Letters, 42, 169-184.*

Orlov, Y. and Dochain, D. (2002). Discontinuous feedback stabilisation of minimum-phase semilinear infinite-dimensional systems with application to chemical tubular reactor models. In *IEEE Trans. Aut. Contr, vol. 47, 1293-1304.*

Pazy, A. (1983). *Semigroups of Linear Operators and Applications to Partial Differential Equations*. Springer-Verlag.

Smets, I., Dochain, D., and Impe, J. V. (2002). Optimal temperature control of a steady-state exothermic plug-flow reactor. In *AIChE Journal, Vol. 48, No. 2, 279-286.*

Winkin, J., Dochain, D., and Ligarius, P. (2000). Dynamical analysis of distributed parameter tubular reactors. In *Automatica, 36, 349-361*.

A DISTRIBUTED FAULT TOLERANT POSITION CONTROL SYSTEM FOR A BOAT-LIKE INSPECTION ROBOT

Christoph Walter, Tino Krueger and Norbert Elkmann
Fraunhofer Institute for Factory Operation and Automation
Sandtorstrasse 22, 39106 Magdeburg, Germany
Christoph.Walter@iff.fraunhofer.de

Keywords: Distributed control system, fault tolerance, self-tuning controller.

Abstract: Here we present the position control system of a swimming inspection robot for large under-ground concrete pipes that are partially filled with wastewater. The system consists of a laser-based measurement subsystem for position determination and a mechanical rudder to move the robot laterally within the pipe. The required software components are implemented as services following a CORBA-based architecture. To automatically adapt the position control system to different environment conditions, a self-tuning controller is used. The controller has hybrid requirements regarding latency and interarrival times of computed position values. In this contribution, we describe the architectural support for this application as well as how the system deals with excessive latencies due to transient overload.

1 INTRODUCTION

The Emschergenossenschaft based in Germany is currently planning the Emscher sewer system, arguably the largest residential water management project in Europe in years to come. The new emscher sewer will show a total length of 51 km and diameters between 1.4 and 2.8 m in depths up to 40 m under surface. The Emschergenossenschaft engaged the Fraunhofer Institute for Factory Operation and Automation (IFF) in Magdeburg, Germany, as the general contractor to develop automatic inspection and cleaning systems to meet the requirements imposed by legal guidelines.

Large under-ground concrete pipes that are partially filled with wastewater are a hazardous environment for man. Nevertheless, inspection of such pipes must be performed on a regular basis. Many of today's remote controlled inspection systems for underground sewer pipes consist of a single TV-camera and are designed for pipe diameters below one meter. The recent development of automatic inspection systems (Elkmann et al., 2005) equipped with advanced sensors makes it possible for an operator to perform this task from an outside position even for pipes with a diameter between 1.4 to 2.8 meters. The cable-guided damage surveying system (SEK) (see Fig. 1) was developed to be a versatile and easy-to-use tool to detect various kinds of damages above and underneath the water-line with high accuracy.

The inspection process is largely automated (Elkmann et al., 2006) and supervised by an operator from within a service-vehicle outside of the sewer. To ensure an optimal inspection result the system must acquire data from a centered position.

Figure 1: The cable-guided damage surveying system within a sewer pipe.

For this reason, it is our goal to keep the robot near the center of the pipe during the data acquisition phase. We must also deal with different environment conditions. These include water level, flow velocity and pipe diameter. A traditional controller with fixed parameter set was unable to maintain a stable posi-

tion depending on the current conditions. To tackle this problem the position control system of the robot was fitted with a self-tuning controller that is able to identify model based control parameters prior to data acquisition.

The main part of the controller was implemented using a software programmable logic controller (PLC). The input to the controller is the current lateral position relative to the pipe axis. An iterative algorithm using data from various sensors, mainly three banks of laser distance sensors, determines the position. The PC-based implementation of this algorithm has an unknown worst-case execution time (WCET). The actual execution time is depending on the internal state of the software as well as on the current input data. Further delay stems from the transmission of data over the Ethernet network between PLC and PC as well as from the scheduling scheme of the calculation task. While the actual controller is robust against certain latency and jitter of the arrival of position data, we must consider cases when position data does not arrive in time. This may happen due to unusual long execution time of the position determination algorithm or due to transient overload of the communication channel.

Another challenge is the second part of the self-tuning controller, which performs the identification of the system's behavior and which derives an adequate control parameter set. The algorithm used here is very sensitive to jitter of the interarrival times of position measurements and to missing samples.

An architectural approach for solving both problems is described in section 3. In section 4, the actual controller is presented in more detail. Section 5 briefly discusses the dynamic quality of service approach used to avoid failure in case of transient overload.

2 RELATED WORK

Realizing time critical applications using complex distributed systems is a challenging task. On one hand, considerable work has been done to introduce time constraints or timely predictable behavior. On the other hand, a number of approaches have been published to relax time constraints of certain applications by introducing fault tolerance mechanisms.

In the field of communication middleware, the Object Management Group (OMG) has released extensions to the known CORBA-specification to extend it with real-time features (Wolfe et al., 1997). The primary focus here is on the introduction of priorities when executing method calls, which requires appropriate system software support. Software libraries implementing this standard were also made available (Schmidt et al., 1998).

To relax time constraints, approaches that try to balance quality of service and required computation time were considered. This includes the classical "anytime" (Dean, Boddy, 1988) or "flexible" (Hendler, 1989) algorithms, where quality of service can be dynamically traded against shorter computation times. This scheme can be used to ensure the timeliness of computations. The authors of (Nett et al., 1998) have also dealt with meeting task deadlines. They use a dynamic task scheduling scheme with detection and handling of timing faults. Faults are handled via a task-pair strategy with invocation of an alternative exception task in case of a timing fault.

3 DATA ACQUISITION AND PROCESSING

3.1 Modeling Data Flow using Services

We developed a service oriented CORBA-based software architecture focusing on data flow and distributed processing. In our system, an application is constructed by combining software modules that provide functionality to other modules via a generic interface. The modules may be arbitrarily distributed over a number of computers in a network. The interface of the services is data centric. It allows services to offer data via named buffers to other services or clients. Each buffer represents a time series of specific data objects (see Fig. 2). A service can represent a data source (e.g. a sensor), a processing module or a data consumer (e.g. an actuator, a database, or a GUI). Services that represent sensors or sensor systems may provide data that represents a single measurement, a time series of measurements, a vector, an image, or any other types of data that are generated by the sensor system. Other services may not provide sensor data but processing results.

By connecting the services via the publish/subscribe paradigm the data flow of the application is modeled. In this communication model, the services play different roles. The module providing data is called the service provider. The data consumer is called the service client. Please note that in our system a client does not subscribe to specific data elements but to notifications on when such elements become available.

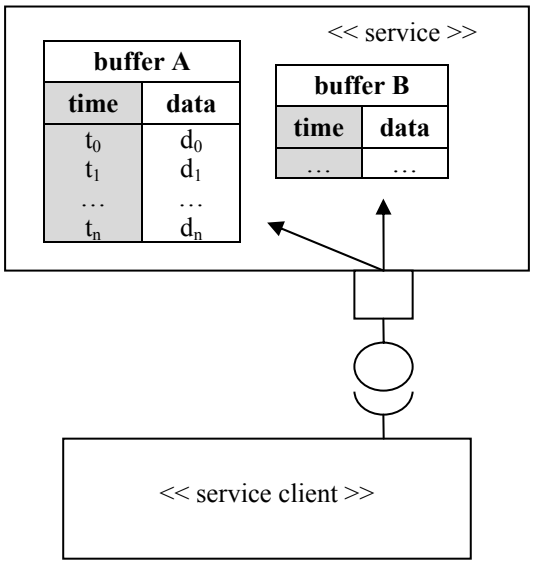


Figure 2: A service provider offers data buffers to clients via a generic interface.

The data objects that are being exchanged between services satisfy a given scheme. They consist of binary data and metadata in XML syntax. The most important element of the metadata is the timestamp, which indicates the recording time of the original data. The timestamp acts as a key for the data object within the given data buffer. The binary data can contain information in any format. A client can request individual data elements or arbitrary blocks of data elements from a buffer. This is achieved by sending structured queries with a simple syntax to the accordant service. The client can then process these data items and/or eventually offer the results to further clients. The foundation of the used timestamps is the Newtonian time model. Timestamps are generated based on synchronized local clocks. A simple clock synchronization is carried out via the network by a dedicated service. The accuracy of synchronization between time-server and any client is better than 100 µs. This is well below the shortest interarrival times of data elements in our application. It was achieved using common PC-hardware, Gigabit-Ethernet, and a Windows NT-based operating system. Accuracy may be improved by using more sophisticated clock synchronization algorithms.

3.2 Implementation of Position Determination

Determining the position of the robot is of great importance. Knowledge of the position is not only required for position control, but also to process and interpret other sensors' data. This includes the registration of images from a high-resolution photographic camera system, the analysis of data from an ultrasonic underwater scanner, the determination of the magnitude of corrosion and incrustations, or the precise measurement of possible deviation of the position of individual pipe segments. These tasks cannot be performed without knowledge of the pose of each sensor during data acquisition.

The heart of the position determination subsystem is an algorithm that was specifically developed for this purpose. It is implemented as a service (the position determination service; PDS). It is based on the measurements of a number of laser ranging sensors plus an inclination sensor. A discussion of the exact procedure can be found in (Elkmann et al., 2006).

Measurements from the laser sensors are acquired by the onboard PLC. The PLC is implemented in software on an embedded PC and tightly coupled with an interface service (IS). The IS has access to the PLC's memory and is configured to insert time-stamped measurements into a buffer queue at a rate of 100 Hz. This rate is consistent with sensor capabilities and sufficient for our application.

The PDS is notified of the arrival of new data elements in the IS's buffer. Its goal is to process all data elements that are produced by the IS. To do this the PDS uses the following strategy of data access: Upon notification of a new data element, it asks the IS for all elements that succeed the last element the PDS has already processed. This is done via a single "get"-operation. By using this technique, it is possible for the PDS to catch up in case of a previous transient overload. For each received data element, the system's position is calculated and eventually made available for further processing by other clients.

4 SELF-TUNING CONTROLLER

4.1 Controller Design

One challenge in the context of stable data acquisition between real time and non-real time systems (hybrid systems) is the implementation of non-linear controllers. This chapter describes the motivation of using a non-linear controller in this area. In addition, it gives a short introduction of calculating them.

When travelling through a partially filled sewer, the position of the SEK is controlled by a rudder arm mounted on top of the robot. The rudder can be

turned relative to the longitudinal axis of the pipe. In turn, buoyancy forces the SEK to change its position. Due to disturbances, a controller has to be integrated for closed-loop navigation. Self-tuning controllers belong to the denomination of adaptive controllers. They are used to fit the plant on a variable environment. In fact, the flow velocity of the wastewater, the water level, and the pipe diameter may change depending on the actual sewer and on current conditions like the time of day or the weather situation. This motivates to use adaptive controllers for the robot. The benefit of adaptive controllers compared to fixed structure designs is the broad field of applications. Here we will focus on the self-tuning controller as a substitute of indirect adaptive controllers.

Intermediate instances are required for the calculation of the controller parameters (see Fig. 3).

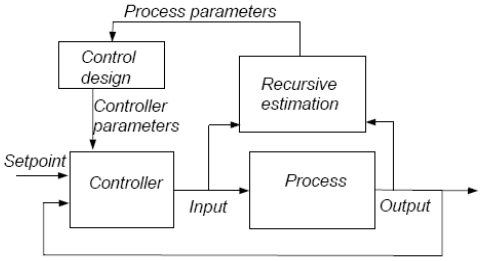


Figure 3: Schematic diagram of a self-tuning controller.

The calculation of a self-tuning controller contains the following steps:

- Recursive estimation of process parameters (system identification)
- Suitable determination of a control law with adjustable parameters
- Calculation of controller parameters based on the desired closed-loop system
- Update the parameters using the IS

The advantage of this method comprises the possibility to secure each step of calculation before the controller parameters are used. In addition, the closed loop can be pushed to a desired behavior while the parameters of the plant are changing.

The plant is depicted by a discrete autoregressive moving average with an extra input model (ARMAX-Model) of fourth-order. The advantage of ARMAX-Models compared to simpler models is the adequate freedom in describing the properties of the disturbance term.

That gives the model:

$$y(t)+a_1y(t-1)+...+a_3y(t-3)= \\ b_1u(t-1)+...b_3u(t-3)+c_1e(t)+...+c_3e(t-3) \quad (1)$$

with

$$\theta=[a_1...a_3 \quad b_1...b_3 \quad c_1...c_3] \quad (2)$$

where y(t) is the plant output, u(t) is the control input and e(t) the prediction error. In this case, the least–squares criterion:

$$V(t)=\frac{1}{t}\sum_{i=1}^{t}\lambda^{t-i}(y(i)-\varphi^T(i)\theta)^2 \quad (3)$$

is being used to identify the transmission behavior of the process. The parameters are time varying in a small range during operation. Hence, it is desirable to base the identification on the most recent data rather than on the older data. This can be achieved by exponential discounting (λ) of old data. Then, it can be shown (Lennard Ljung, 1999) that the extended recursive least square scheme (ERLS) becomes:

$$\theta(\mathrm{t})=\theta(\mathrm{t}\text{-}1)+\mathrm{L(t)}[\mathrm{y(t)}-\varphi^{\mathrm{T}}(\mathrm{t})\,\theta(t-1)] \\ \mathrm{L(t)}=\frac{\mathrm{P(t}\text{-}1)\varphi(\mathrm{t})}{(\lambda+\varphi^{\mathrm{T}}(\mathrm{t})\mathrm{P(t}\text{-}1)\varphi(\mathrm{t}))} \\ P(t)=\frac{1}{\lambda}\left(P(t-1)-\frac{P(t-1)\varphi^T(t)P(t-1)}{\lambda+\varphi^T(t)P(t-1)\varphi(t)}\right) \quad (4)$$

Using ERLS it is advisable to consider the persistent excitation in view of convergence of the parameter estimation. In this step, the indirect method permits to prove the estimated process parameters on separate recorded test series. If the verification is successful, the calculation of the controller parameter is performed.

Fig. 4 depicts exemplary the change of positions predicted by a calculated model towards measured data. The rudder angle as input to the ERLS was oscillating between minus five and five degrees. The calculated model is afterwards the starting point for the controller design.

The idea regarding the controller design is to compensate the plant with the inverse system and a desired transmission behavior.

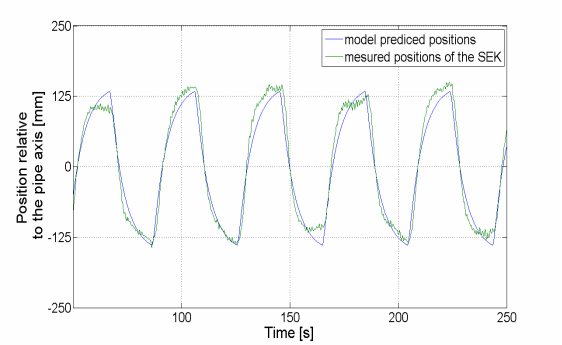


Figure 4: Measured and predicted positions of the SEK.

Assuming that the desired close loop behavior as shown in equation 5 where z-1 is backward shift operator.

$$M(z^{-1}) = \frac{\hat{B}(z^{-1})}{\hat{A}(z^{-1})} \tag{5}$$

The actual closed loop is:

$$G(z^{-1}) = \frac{K(z^{-1})S(z^{-1})}{1 + K(z^{-1})S(z^{-1})} \tag{6}$$

with the controller:

$$K(z^{-1}) = \frac{D(z^{-1})}{C(z^{-1})} \tag{7}$$

and the plant behavior:

$$S(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} \tag{8}$$

The suitable controller law is then given by equation 9 without solving a Diophantine equation.

$$\begin{aligned} D(z^{-1}) &= A(z^{-1})\hat{B}(z^{-1}) \\ C(z^{-1}) &= B(z^{-1})(\hat{A}(z^{-1}) - \hat{B}(z^{-1})) \end{aligned} \tag{9}$$

In terms of the causality of the controller law, we get the following conditions for the implementation:

- The transmission behavior of the plant has to be stable, respect to the inverse implementation in the controller law.
- The polynomial of the decoupled model $\hat{A}(z^{-1}) - \hat{B}(z^{-1})$ should only have zeros within the unit circle.
- The estimated plant model should not include a minimum phase system or must be corrected because of the stability of the open loop.
- The controller law has to be causal.

The calculated controller is therefore less sensitive to noise compared to other designs like the pole placement variant, which is described by (Aström, Wittenmark, 1992).

4.2 Implementation

The self-tuning controller was implemented in the form of two main modules. The first module is the actual controller. It runs as a real-time task within the onboard PLC. Position-data is fed into it by the IS as described in section 3.2. The IS acts also as a client similar to the PDS but with a different strategy of data access: The controller doesn't want to process all data elements that are produced by the PDS, but needs the most recent one. For this reason the IS upon notification of new data requests the most recent data element from the PDS and feeds it into the PLC.

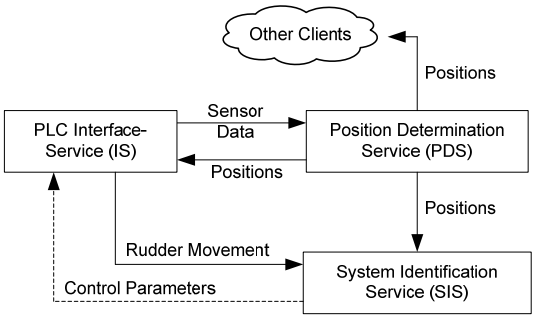


Figure 5: Structure and data flow of controller implementation.

The second module is a service, which computes the control parameter set (system identification service; SIS). It needs two kinds of data elements as input: (1) all sampled rudder arm angles during the system identification process and (2) the corresponding system position. The arrival of these data is not time critical but it is necessary that all samples are being processed. It therefore uses a similar data access strategy as the PDS to gather all data elements that describe the rudder movement from the IS. Furthermore, it requests for each such element the two closest system positions from the PDS and interpolates between them to get the corresponding system position. Requesting closest elements given a certain timestamp is another feature of the data-access mechanism of our architecture. After the SIS has collected enough data, it can compute the control parameters and send them to the main control module via the IS. The described data flow is shown in Fig. 5.

5 DYNAMIC QUALITY OF SERVICE

Even though we have successfully implemented the system identification part of the controller as a non-real-time service while guarantying the required data quality, the actual controller may still suffer from excessive latencies in the arrival of position data from the PDS. This is because of (1) the iterative algorithm used in the PDS and (2) because of large latencies in the communication between the distributed services due to transient overload of the communication channel.

To avoid system failure in such cases we integrated a dynamic quality of service approach that uses the timestamps of the position data available to the controller implemented in the PLC to determine the age of that data. If a certain age is exceeded, the controller uses an alternative position that it computes by a simple prediction mechanism based on the last known positions. This fall back mechanism is triggered if the position data available to the PLC becomes older then 90 ms. While this is a crude method and could easily be improved by using a more accurate model based approach, it is sufficient to deal with short disruptions of communication. If however the position data becomes older then 900 ms, the controller is put into a safe state. In this state, the rudder is moved into centered position, which is the state that most unlikely causes a collision with the wall. The controller resumes as soon as position data not older then 90 ms becomes available again.

We performed experiments with a prototypic implementation of the system under operating conditions. To determine the actual interarrival times of data produced by the PDS, we used a test-client that was consuming position data in the same way as the IS. In Fig. 6, the interarrival times recorded by the test-service are shown. Well under 5 percent of the position values arrived slightly later then usual. Less then 1 percent of that were delayed more than 90 ms. When that happened, approximately 50 percent were delayed more than 900 ms.

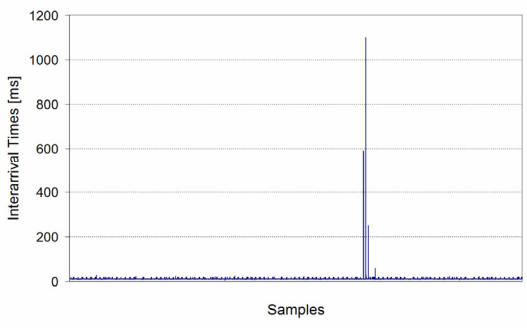


Figure 6: Interarrival times of position data at Test-Service.

When observing the behavior of the robot during the experiment, no significant deviation from the ideal position could be noticed. The system maintained position near the center of the pipe when missing the 90 ms deadline. This was also the case when briefly missing the 900 ms deadline. This can be attributed to the fact that we only had small turbulences that required only slight corrections by the position control system. However, we expect that missing the 900 ms deadline may have a significant effect in case of more severe turbulences.

6 CONCLUSIONS

Here we presented an application of a distributed control system for position control of a boat-like inspection robot for partially filled sewer pipes. We discussed the hybrid requirements of an advanced self-tuning controller regarding timeliness and interarrival times of its input data. We presented a scheme for detecting and handling excessive time delays to prevent system failure by using dynamically reduced quality of the position controller. It was also discussed, how the controller was integrated into a non-real-time service oriented software system. We conclude that in case of our application it is feasible to use soft- or non-real-time components in the control loop. Prerequisites are (1) a deterministic data acquisition with time stamping and (2) a hard-real-time controller core with fault tolerance mechanisms.

REFERENCES

Elkmann N., Reimann B., Schulenburg E., Althoff H., 2005. Automated inspection system for large underground concrete pipes under operating conditions.

In *Proc. International Conference on Field and Service Robotics*.

Elkmann N., Kutzner S., Saenz J., Reimann B., Schultke F., Althoff H., 2006. Fully automatic inspection systems for large underground concrete pipes partially filled with wastewater. In *Proc. of International Conference Intelligent Robots and Systems (IROS06)*.

Wolfe V. F., DiPippo L. C., Cooper G., Johnston R., Kortmann P., Thuraisingham B. M., 1997. Real-Time CORBA. In *Proc. of the Third IEEE Real-Time Technology and Applications Symposium*.

Schmidt D. C., Levine D. L., Mungee S., 1998. The Design of the TAO Real-Time Object Request Broker. In *Computer Communications 21(4), p. 294-324*

Dean T. L., Boddy M., 1988. An analysis of time-dependent planning. In *Proceedings of the Seventh National Conference on Artificial Intelligence, Minneapolis, Minnesota, p. 49-54*.

Hendler J. A., 1989. Real-time planning. In *Working Notes of the AAAI Spring Symposium on Planning and Search, Stanford*.

Nett E., Gergeleit M., Mock M., 1998. An adaptive approach to object-oriented real-time computing. In *Proceedings of ISORC'98, p. 342-349.*

Ljung L., 1999. *System Identification*, PTR Prentice Hall

Astrom K., Wittenmark B., 1992. *Adaptive Control*, Pearson Education

COLLISION AVOIDANCE SYSTEM PRORETA
Strategies Trajectory Control and Test Drives

R. Isermann, U. Stählin and M. Schorn
Institute of Automatic Control, Technische Universität Darmstadt
Landgraf-Georg-Str. 4, 64283 Darmstadt, Germany
risermann@iat.tu-darmstadt.de

Keywords: Collision avoidance, vehicle, object detection, path following, automatic braking, automatic steering.

Abstract: Methods and experimental results of a collision avoidance driver assistance system are described with automatic object detection, trajectory prediction, and path following with controlled braking and steering. The objects are detected by a fusion of LIDAR scanning and video camera pictures resulting in the location, size and speed of objects in front of the car. A desired trajectory is calculated depending on the distance, the width of a swerving action and difference speed. For the trajectory control different control methods were designed and tested experimentally like velocity depend linear feedback and feedforward control, nonlinear asymptotic output tracking and nonlinear flatness based control using extended one-track models with vehicle state estimation for the sideslip angle and cornering stiffness. Automatic braking is realized with an electrohydraulic brake (EHB) and automatic steering with an active front steering (AFS). The various control systems are compared by simulations and real test drives showing the behaviour of a VW Golf with automatic braking or/and automatic swerving to a free track, such avoiding hitting a suddenly appearing obstacle. The research project PRORETA was a four-years-cooperation between Continental Automotive Systems and Darmstadt University of Technology.

1 INTRODUCTION

Driver Assistance Systems for Collision Avoidance have the goal to prevent accidents using braking or evasive maneuvers. An automatic collision avoidance system has to monitor its surroundings, detect an upcoming accident and intervene appropriately to avoid the accident. In case of the system developed by the research project PRORETA – Electronic Driver Assistance System for a Collision Avoiding Vehicle, a cooperation between Technische Universität Darmstadt and Continental AG, the driver is given the chance to avoid the accident himself as long as possible. Therefore, the interventions have to be conducted at the physical last possible moment or the driving dynamics stability boundary.

Using these predictions, a decision is made, whether an intervention is necessary or not and the intervention is planned. The intervention itself is then conducted fully automatically. An ergonomic study accompanied the development of the system. This study investigated how the driver reacts in critical situations and how he reacts on the interventions by PRORETA.

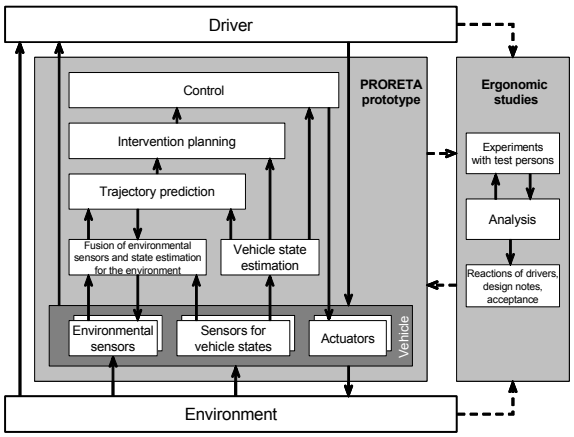


Figure 1: System overview prorate.

In this contribution, the intervention decision, the planning of the intervention and the conduction of the intervention are described. The environment perception is described in detail in (Darms and Winner, 2006). Results from the ergonomic study can be found in (Bender and Landau, K., 2006). The system was tested using a complex two track model followed by extensive driving tests with an experimental vehicle.

2 TEST VEHICLE

A VW Golf IV, which was only equipped with additional sensors and actuators required for the developed functions, served as experimental vehicle, see Figure 2.

Figure 2: Environmental Sensors of the test vehicle.

The driver assistance system uses an active front steering and an electro hydraulic braking system as actuators. For vehicle state estimation only ESP sensors and the sensors of the active front steering and braking system are necessary. For environment perception a laser scanner and a video sensor were used. The chosen design allows to scan the area in front of the vehicle. The detection area of the laser scanner covers an angular range of 22.5° with a resolution of 1.5° and is scanned in a 90 ms cycle. The distance to objects is determined by a time of flight measurement of emitted light impulses. The video sensor is based on a monochrome CMOS image sensor that provides data in a 40 ms cycle. The detection area is 44°, whereas the discretisation with approx. 0.07° is considerably finer than for the laser scanner. By means of image processing algorithms, vehicle rear views and lane markings can be detected in the image, however, a direct distance measurement is not possible, for details see (Darms and Winner, 2006).

3 EVASIVE TRAJECTORY

An evasive trajectory is required between intervention planning and control For investigating several different intervention functions with different types of controllers, the type of intervention is selected using some flags. The flags used in this article are braking, emergency braking and evasion. If braking is chosen, the desired deceleration has to be transmitted. If an emergency braking is chosen, the maximum possible deceleration at every point in time is achieved using braking controllers. For an evasion, the desired position and heading are given for one time step T_B, two time steps T_B and ten time steps T_B ahead in time, Figure 3. The coordinate system used is stationary for the duration of the evasion and is initialized at the beginning of the evasion to match the vehicle coordinate system at that point. The last position, which is supposed to be reached 10 time steps in the future, is used to make sure the controller can react predictively on deviations of the first 2 time steps. Figure 3 shows this interface. Every point $p(t)$ consists of the position (x, y) and the heading ψ of the vehicle. All three points are put together in one matrix transmitted to the controller:

$$\mathbf{P}_{\text{evasion}} = \left[p\left(t+T_B\right) \quad p\left(t+2T_B\right) \quad p\left(t+10T_B\right) \right]$$

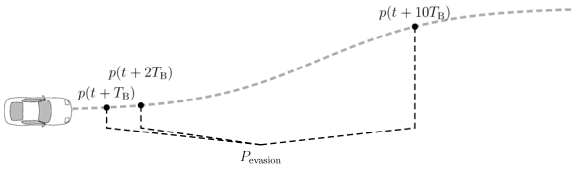


Figure 3: Evasive trajectory between planning and control.

4 PLANNING OF THE EVASIVE INTERVENTION

Primary goal of the evasive trajectory is to reach a predefined lateral offset with the shortest possible traveled path. The designed trajectory has to be feasible regarding the vehicle dynamics laws and after the maneuver the vehicle has to be in a safe and stable state.

Vehicle dynamics laws of the trajectory are taken into account to limit the maximum allowed lateral acceleration. This limit can be adapted to the actual traffic and driving situation, especially weather conditions. The steering actuator also limits the maximum possible jerk.

Since the trajectory is transmitted to the controller using positions, the general relations between the position on the trajectory and the driving dynamics are considered first. This relation is based on the simple equations of the one-track model and the Ackermann equations. The approach shown here uses a relation where the y-position on the trajectory is depended on the x-position:

$$y = f(x) \tag{1}$$

Using geometric transformations, the yaw angle ψ can be expressed as (assuming no side slip)

$$\psi = \arctan\left(\frac{dy}{dx}\right) \tag{2}$$

and therefore its derivative regarding time yields:

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{1}{1+\left(\frac{dy}{dx}\right)^2}\frac{d^2y}{dx^2}v_x \tag{3}$$

Based on this and using the Ackermann relations, the lateral acceleration is:

$$a_y = v\dot{\psi} = \frac{1}{1+\left(\frac{dy}{dx}\right)^2}\frac{d^2y}{dx^2}v_x v \tag{4}$$

Further simplification can be accomplished assuming $v_x = v$.

Often, a sequence of klothoids is used for the evasive trajectory, see e.g. (Ameling, 2002). In general, trajectories for evasive maneuvers have the shape of a lying S. Functions describing such a shape are called sigmoidal functions or sigmoides. In the following a sigmoide of the form

$$y(x) = \frac{B}{1+e^{-a(x-c)}} \tag{5}$$

is used. B is the maneuver width, describing the distance between minimum and maximum y-value. a defines the slope of the sigmoide, where high values for a are leading to a steeper curve. c defines the position of the inflection point and therefore the length of the evasive maneuver, which is s=2c. Looking at equation (5), the sigmoide has its maximum and minimum at infinity, meaning

$$\lim_{x\to-\infty} y(x) = 0 \tag{6}$$

and

$$\lim_{x\to\infty} y(x) = B \tag{7}$$

respectively. Therefore, an additional parameter y_{Tol} is introduced. Using this parameter, the following counts:

$$y(0) = y_{\mathrm{Tol}} \tag{8}$$

and

$$y(2c) = B - y_{\mathrm{Tol}} \tag{9}$$

Figure 4 shows this sigmoide and the respective parameters. The parameters can be chosen according to the driving situation, such that the evasive path is minimal regarding the limitations for maximum lateral acceleration, maximal jerk and dynamics of the steering actuator. The derivation of the respective parameters can be found in (Stählin et al., 2006).

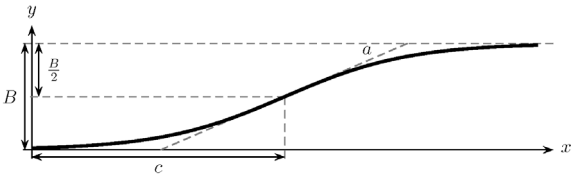


Figure 4: Evasive sigmoide and its parameters.

The most important value for taking a decision, whether the klothoide or the sigmoide should be chosen for the evasive trajectory is the length s of the path of the evasive maneuver, taking into account the limiting factors (maximum lateral acceleration, maximum jerk,…). Table 1 shows a comparison between klothoide and sigmoide for different limiting factors. It can be seen, that the sigmoide always leads to a shorter path for the evasive maneuver. This is due to the linear increase of the lateral acceleration for the klothoide in comparison to the faster and nonlinear increase in lateral acceleration for the sigmoide. Both trajectories can be realized by a controller trajectory.

Table 1: Comparison of the length of an evasion for klothoide and sigmoide with y_A=y_m.

B	v	lateral accel.	jerk	Klothoide	Sigmoide
2m	15m/s	5m/s2	30m/s3	26,83m	22,08m
3m	15m/s	5m/s2	30m/s3	32,85m	29,10m
2m	36m/s	5m/s2	30m/s3	64,30m	53,39m
3m	36m/s	5m/s2	30m/s3	78,85m	70,42m

5 INTERVENTION DECISION

Based on the fused environment data it is decided if a collision is likely to occur and if so, which maneuver has to be carried out to avoid the collision. The strategy is to avoid the collision at the physically last possible moment by an intervention in order to give the driver the possibility to defuse the critical situation by his own actions as long as possible.

In order to determine a threatening collision, predictions are first made for the own vehicle driving tube and the movement of the objects in the environment. By means of these predictions it can then be predicted whether a collision will occur. If this is the case, it is planned in a next step when and which intervention has to be carried out.

Basically, there are three strategies to avoid a collision:

- Braking
- Steering
- Combination of braking and steering

For the intervention decision it is calculated at what distance to the collision location the respective intervention has to be carried out, such that the collision can still be prevented. For a braking intervention the braking distance is calculated. In case of steering interventions the sigmoide is taken as the basis for the evasive trajectory.

In Figure 5 the quantities necessary for the calculation of the evasive trajectory are presented. By means of the vehicle's width b_v and the obstacle's width the necessary evasive width y_A is determined together with a safety distance y_S. Since the evasive width can be reached before the end of the maneuver, an associated maneuver width y_M arises.

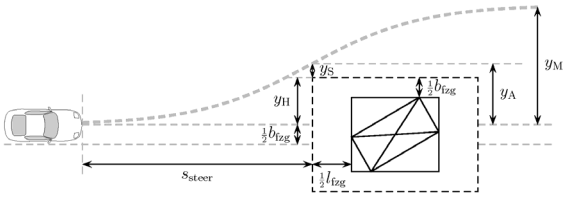


Figure 5: Evasive Quantities for calculating the evasive trajectory (see text for details).

However, the evasive trajectory is the trajectory until the evasive width y_A is reached. The maneuver width is chosen according to the strategy used. If the maneuver width y_M is chosen to be the same as the evasive width y_A, the evasive trajectory length s_{steer} reaches its maximum for given maximal lateral acceleration and maximal lateral jerk. These two last-named parameters also determine the optimal maneuver width which leads to the smallest possible evasive trajectory length s_{steer} and which makes use of the set limits ideally. However, it needs considerably more lateral offset for the same evasive width y_A.

6 LATERAL VEHICLE GUIDANCE

If a collision with an obstacle is no longer avoidable by a reaction of the driver, then, according to the situation, the driver assistance system selects one of the intervention strategies described above. For the realization of the chosen intervention either the active steering and/or the electro hydraulic braking system are used according to the maneuver. If a braking maneuver should be carried out, the vehicle is decelerated (Schorn et al., 2005) by utilization of the maximum force transmission available. The anti-lock braking system ABS supports in this case.

In case a collision can only be prevented by an evasive maneuver or by a combined evasive and braking maneuver the control block receives from intervention planning a trajectory, see Figure 1. The vehicle is driven on this trajectory automatically around the obstacle. Different linear and nonlinear feedback controllers for an evasive maneuver were developed, see e.g. (Schorn and Isermann, 2006), (Schorn et al., 2006). Each lateral guidance feedback control transfers an additional steering angle to the interface of the steering system. Vehicle variables, which cannot be measured directly by sensors the vehicle is equipped with, are estimated, see Figure 1, see also (Schorn and Isermann, 2006), (Isermann, 2006). For combined steering and braking maneuvers different feedback controllers were developed as well.

In the following only the lateral vehicle guidance is regarded. Exemplarily, two of the investigated approaches, a *nonlinear asymptotic output tracking feedback control* and a *speed-dependent local linear feedback control approach with feedforward control* are presented.

6.1 Nonlinear Asymptotic Output Tracking Feedback Control

For model based design of a feedback system, the system behavior is required. The path following feedback control is based on an extended one-track model:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} a_{11} \cdot x_1 + a_{12} \cdot x_2 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 \\ a_{31} \cdot \sin\left(x_1 + x_4\right) \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \\ 0 \\ 0 \end{bmatrix} \cdot u \qquad (10)$$

$$y = y_E = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \cdot \mathbf{x} = x_3 \qquad (11)$$

with

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \beta(t) \\ \dot{\psi}(t) \\ y_E(t) \\ \psi(t) \end{bmatrix} \quad \text{and} \quad u = \delta(\mathrm{t}) \qquad (12)$$

β is the sideslip angle, ψ the yaw angle, y_E the lateral vehicle position and δ the steer angle. The speed dependent parameters follow from front and rear cornering stiffness $c_{\alpha F}$ and $c_{\alpha R}$, length from

front and rear axle to the center of gravity l_F and l_R, velocity v, mass m and moment of inertia J_Z, see e.g., (Isermann, 2006):

$$\begin{aligned} a_{11} &= -\frac{c_{\alpha F}+c_{\alpha R}}{m\cdot v} & a_{12} &= \frac{l_R\cdot c_{\alpha R}-l_F\cdot c_{\alpha F}}{m\cdot v^2}-1 \\ a_{21} &= \frac{l_R\cdot c_{\alpha R}-l_F\cdot c_{\alpha F}}{J_Z} & a_{22} &= -\frac{l_R^2\cdot c_{\alpha R}+l_F^2\cdot c_{\alpha F}}{J_Z\cdot v} \\ a_{31} &= v & & \\ b_{11} &= \frac{c_{\alpha F}}{m\cdot v} & b_{21} &= \frac{c_{\alpha F}\cdot l_F}{J_Z} \end{aligned} \tag{13}$$

The model parameters were determined from construction data and identification experiments, (Schorn, 2007).

The lateral position $y_E(t)=f(x_E(t))$ of the vehicle in an earth-fixed coordinate system has to be controlled using the evasive trajectory described by equation (5). The reference input of the control system $y_R(t)$ is calculated by performing an interpolation.

The vehicle model in (10) and (11) is a nonlinear single-input single-output model of type:

$$\begin{aligned} &\dot{\mathbf{x}}(t)=\mathbf{a}(\mathbf{x})+\mathbf{b}(\mathbf{x})\cdot u(t) \\ &\mathbf{y}(t)=\mathbf{c}(\mathbf{x}(t)) \qquad \text{with} \qquad \mathbf{x}_0=\mathbf{x}(t_0) \end{aligned} \tag{14}$$

Having the system's output $\mathbf{y}(t)$ converging asymptotically to a prescribed reference output $y_R(t)$, the system input $u(t)$ can be calculated as follows (Isidori, 1989), (Schwarz, 1999):

$$\begin{aligned} u(t)=&\frac{1}{\mathrm{L}_\mathbf{b}\mathrm{L}_\mathbf{a}^{d-1}c(\mathbf{x})}\cdot \\ &\left[-\mathrm{L}_\mathbf{a}^d c(\mathbf{x})+y_R^{(d)}(t)-\sum_{i=1}^{d}\alpha_{i-1}\cdot\left[\mathrm{L}_\mathbf{a}^{i-1}c(\mathbf{x})-y_R^{(i-1)}(t)\right]\right] \end{aligned} \tag{15}$$

The relative degree d has to be determined according to (Isidori, 1989), (Schwarz, 1999). For the mentioned plant it yields $d = 2$ assuming $v > 0$ and $(x_1+x_4)\neq\pm\frac{\pi}{2},\pm\frac{3\pi}{2},\ldots$. With this information, the feedback control, equation (15), can be calculated. The elements are given by (Schorn et al., 2006):

$$\begin{aligned} &c(\mathbf{x})=x_3 \\ &\mathrm{L}_\mathbf{a}c(\mathbf{x})=v\cdot\sin(x_1+x_4) \\ &\mathrm{L}_\mathbf{a}^2c(\mathbf{x})=-v\cdot\cos(x_1+x_4)\cdot(a_{11}x_1+a_{12}x_2+x_2) \\ &\mathrm{L}_\mathbf{b}\mathrm{L}_\mathbf{a}c(\mathbf{x})=b_{11}\cdot v\cdot\cos(x_1+x_4) \end{aligned} \tag{16}$$

where L_a are so called Lie-functions (Isidori, 1989), (Schwarz, 1999). The structure of the resulting closed control loop is shown in Figure 6. The command variables, the reference output $y_R(t)$, $\dot{y}_R(t)$, $\ddot{y}_R(t)$ as described above, are calculated in the component "Calculation of command variables". The sideslip angle was estimated with a vehicle state estimator (Schorn and Isermann, 2006). The output of the controller is an additional steering angle $\delta_M(t)=\delta(t)$.

Results from test drives with the experimental vehicle described above will be presented below.

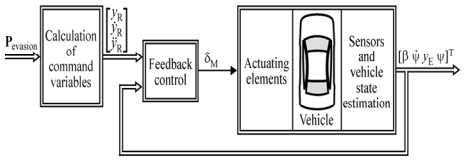


Figure 6: Structure of nonlinear asymptotic output tracking feedback control.

6.2 Speed-dependent Local Linear Feedback Control with Feedforward Control

To guide a vehicle on a desired trajectory, a speed-dependent local linear feedback control approach with feedforward control was developed (Schorn and Isermann, 2006). A scheme of the control system is shown in Figure 7.

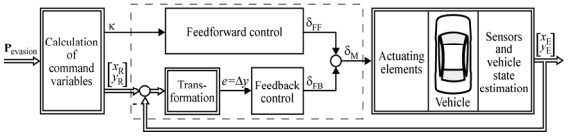


Figure 7: Structure of linear feedback control combined with feedforward action.

Based on the self steer gradient SG a steer angle δ_{FF} is calculated for the feedforward control by means of vehicle velocity v, wheelbase l and curvature $\kappa=\frac{1}{R}$ of the desired trajectory:

$$\delta_{FB}=\left(l+SG\cdot v^2\right)\cdot\kappa \tag{17}$$

A feedback control is added to compensate disturbances and deviations. The parameters of a proportional-derivative (PD) controller is tuned by two parameters only and provides the required dynamics by means of the differential component. Using the vehicle orientation ψ, the control deviation is transformed from an earth-fixed coordinate system into a vehicle-fixed coordinate system as control deviation $e=\Delta y$. The feedback of the vehicle's longitudinal position x_E is necessary for this purpose. The steering system is driven by the sum δ_M of the angles δ_{FF} and δ_{FB} of the feedforward and feedback control. For the implementation of the feedback control in the experimental vehicle the derivative, required for the calculation of the differential component of the control variable, was replaced by a high pass filter.

As the velocity v influences the vehicle's dynamics, because it changes continuously during a

driving cycle. The feedback controllers were designed for different operating points (velocities). Their outputs are weighted and superimposed based on Local Linear Models (LLM) (Schorn and Isermann, 2006), (Nelles, 2001).

7 EXPERIMENTAL RESULTS FROM TEST DRIVES

The developed components environment recognition, intervention decision and feedback control were implemented as a whole system in a test vehicle and tested by means of numerous experiments. This happened using an obstacle that represents the rear view of a car and can be moved laterally on the lane. Two test scenarios can be seen in Figure 8.

In the following sections the most important results from these tests are presented. It is required in each case that the lateral and back lane areas are monitored by additional sensors and thus permit driving maneuvers.

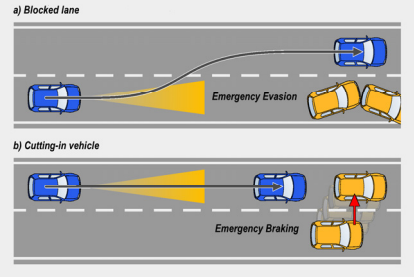


Figure 8: Scenarios for practice system testing.

7.1 Blocked Lane

In the scenario "Suddenly appearing obstacle / blocked lane" from Figure 8a) a lane is blocked unexpectedly. An example for this would be an end of a traffic jam in the case of bad visibility or after a curve. The emergency evasion is then conducted as an automatic intervention. The position of the used obstacle is determined by the environmental sensors and the necessary evasive trajectory is calculated based on the information about the vehicle's surroundings. The vehicle is then guided aside of the obstacle on the predefined evasive trajectory by the lateral guidance controller without the assistance of the driver.

Figure 9 shows results of a test drive with the test vehicle mentioned above, where the *asymptotic nonlinear output tracking feedback control* was used. A comparison of desired command variable and measured position shows that both values match very well. The evasive width y_M is 3m, the desired and the actual position correspond well, apart from a slight overshooting. The steering wheel angle indicates that the driver held the steering wheel in a straight position. The difference between total angle and steering wheel angle is provided only by the controller. The difference at the end of the intervention maneuver follows from the fact that the feedback control has been switched off at very low velocities.

Results from test drives for the *linear feedback control combined with feedforward* are presented in Figure 10.
Again, desired command variable and measured position match very well. The general conditions for this test drive have been the same as for the results shown in Figure 9 regarding evasive width y_M and velocity. The experiments show that the maximal lateral accelerations are $a_y \approx 7\,m/s^2$ and the linear controller needs less maximal steering angle. Both control approaches presented above provide similar accuracies, but the speed dependent linear control system can be implemented and parameterized easier and with smaller computational expense.

7.2 Cutting-in Vehicle

As a second scenario a suddenly cutting-in vehicle is reproduced by moving the dummy obstacle just in front of the vehicle from the right to the left lane (Figure 8b). Evasion is not possible since further obstacles block the right. The necessary maneuver is thus an emergency braking maneuver. By means of the environmental sensors it is recognized that both lanes of the road are blocked and it is calculated at which last possible moment the emergency braking maneuver must be started in order to come to a stop just before the obstacle. Assuming a maximum braking acceleration which is dependent on the road state (dry-wet), the required braking distance of the vehicle is calculated depending on the current speed. The driver assistance system triggers a braking intervention only if this minimal braking distance is reached in order to give the driver the chance to prevent the collision as long as possible by himself. The electro hydraulic braking system then decelerates the vehicle maximally with support by the anti-lock braking system ABS, on dry roads with a deceleration of $a_x \approx 10\,m/s^2$.

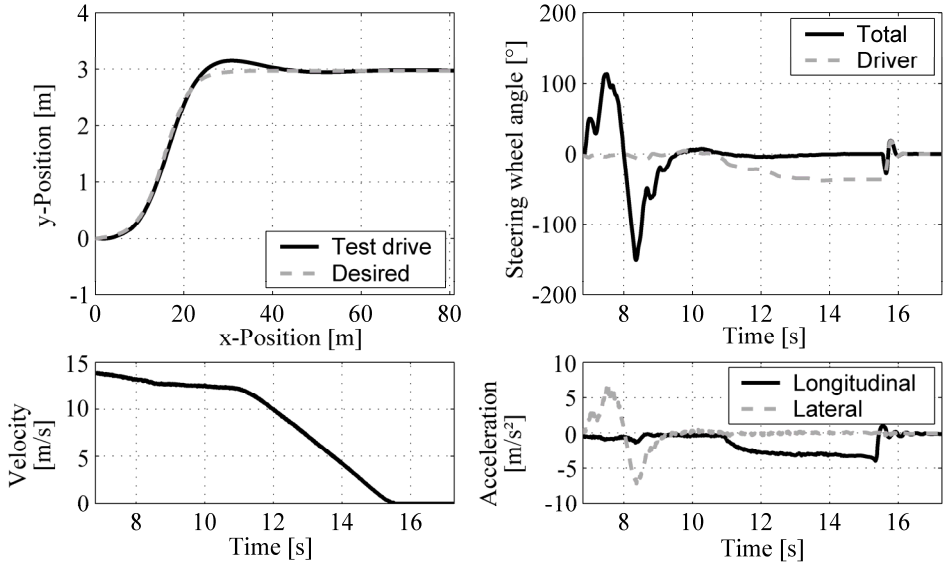


Figure 9: Results for asymptotic nonlinear output tracking feedback (test drive).

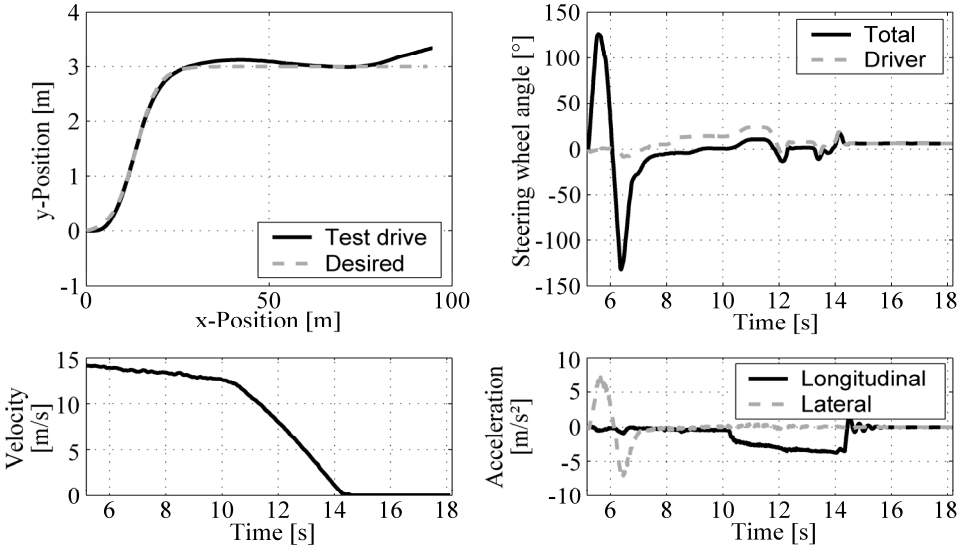


Figure 10: Results for linear feedback control combined with feedforward (test drive).

8 CONCLUSIONS

The described system for accident avoidance which was developed within the scope of the project PRORETA was presented to an audience selected by Continental Teves and TU Darmstadt. The guests had the possibility of experiencing the system within the scope of driving experiments. Every guest drove, amongst others, the scenarios presented in Figure 8. The system worked robustly and faultlessly. However, until such a system is available on the market, some tasks have still to be solved. An important one is the analysis of the oncoming traffic which is examined in a subsequent PRORETA project.

ACKNOWLEDGEMENTS

The authors highly appreciate the financial support for the PRORETA project and good cooperation with Continental Automotive Systems and the support especially by Dr. R. Rieth, J. Diebold, M. Arbitmann, Dr. S. Lüke, B. Schmittner. We also

would like to thank our colleagues within the PRORETA Team, Eva Bender and Michael Darms with Prof. Winner, Prof. Landau and Prof. Bruder for the excellent cooperation. Their results are published e.g. in (Darms and Winner, 2006), (Bender et al., 2006).

REFERENCES

Ameling, C., 2002. *Steigerung der aktiven Sicherheit von Kraftfahrzeugen durch ein Kollisionsvermeidungs-system.* VDI Verlag, Düsseldorf.

Bender, E., Darms, M., Schorn, M., Stählin, U., Isermann, R., Winner, H., Landau, K.. 2007, Anti collision system PRORETA - on the way to the collision avoiding vehicle. Part 1: Basics of the System & Part 2: Results. ATZ (Automobiltechnische Zeitschrift, English Supplement) 109 (4 & 5) 336-341 & 456-463.

Bender, E., Landau, K., 2006. Fahrerverhalten bei automatischen Brems- und Lenkeingriffen eines Fahrerassistenzsystems zur Unfallvermeidung, VDI-Bericht 1931, *AUTOREG 2006*, Wiesloch.

Bender, E.; Landau, K., Bruder, R., 2006. Driver reactions in response to automatic obstacle avoiding manoeuvres. In: *IEA 2006 – 16th World Congress on Ergonomics*, July 10-14, 2006, Maastricht, the Netherlands.

Darms, M., 2007. Eine Basis-Systemarchitektur zur Sensordatenfusion von Umfeldsensoren für Fahreassistenzsysteme. Dissertation. Fortschr.-Ber. Reihe 12, No. 653. VDI-Verlag, Düsseldorf.

Darms, M., Winner, H., 2006. Umfelderfassung für ein Fahrerassistenzsystem zur Unfallvermeidung, VDI Bericht 1931, *AUTOREG 2006*, Wiesloch.

Isermann, R., 1989. Digital control systems. Springer-Verlag, Berlin and New York.

Isermann, R. (ed), 2006. *Fahrdynamik-Regelung, ATZ/MTZ-Fachbuch,* Vieweg, Wiesbaden.

Isidori, A., 1989. *Nonlinear Control Systems – An Introduction.* Springer-Verlag. Berlin.

Nelles, O., 2001. *Nonlinear System Identification*. Springer-Verlag. Berlin.

Schorn, M., 2007. Quer- und Längsregelung eines Personenkraftwagens für ein Fahrerassistenzsystem zu Unfallvermeidung. Dissertation. Fortschr.-Ber. VDI Reihe 12, no 651. VDI Verlag, Düsseldorf.

Schorn, M., Schmitt, J., Stählin, U., Isermann, R., 2005. Model-based braking control with support by active steering. In: *16th IFAC World Congress*, July 4-8, 2005, Prague, Czech Republic.

Schorn, M., Isermann, R., 2006. Automatic Steering and Braking for a Collision Avoiding Vehicle. *4th IFAC-Symposium on Mechatronic Systems*, September 12-14, Wiesloch / Heidelberg.

Schorn, M., Stählin, U., Khanafer, A., Isermann, R., 2006. Nonlinear Trajectory Following Control for Automatic Steering of a Collision Avoiding Vehicle. *American Control Conference*, June 14-16, Minneapolis, Minnesota, USA.

Schwarz, H., 1999. *Einführung in die Systemtheorie nichtlinearer Regelungen*. Shaker Verlag. Aachen.

Stählin, U., Schorn, M., Isermann, R., 2006. Notausweichen für ein Fahrerassistenzsystem zur Unfallvermeidung, VDI Bericht 1931, *AUTOREG 2006*, Wiesloch.

OBDD COMPRESSION OF NUMERICAL CONTROLLERS

Giuseppe Della Penna, Nadia Lauri, Daniele Magazzeni
Computer Science Department,University of L'Aquila, L'Aquila, Italy
dellapenna@di.univaq.it, lauri@di.univaq.it, magazzeni@di.univaq.it

Benedetto Intrigila
Department of Mathematics, University of Roma "Tor Vergata", Roma, Italy
intrigil@mat.uniroma2.it

Keywords: Numerical Controllers, Compression Techniques, Ordered Binary Decision Diagrams.

Abstract: In the last years, the use of control systems has become very common, especially in the *embedded systems* contained in a growing number of everyday products. Therefore, the problem of the *automatic synthesis* of control systems is extremely important. However, most of the current techniques for the automatic generation of controllers, such as *cell-to-cell mapping*, *dynamic programming*, *set oriented approach* or *model checking*, typically generate *numerical controllers* that cannot be embedded in limited hardware devices due to their size.

A possible solution to this problem is to *compress* the controller. However, most of the common lossless compression algorithms, such as LZ77, would decrease the controller performances due to their decompression overhead.

In this paper we propose a new, completely automatic OBDD-based compression technique that is capable of reducing the size of any numerical controller up to a *space savings of 90%* without any noticeable decrease in the controller performances.

1 INTRODUCTION

Control systems (or, shortly, *controllers*) are small hardware/software components that control the behavior of larger systems, the *plants*. A controller continuously looks at the plant *state variables* and possibly adjusts some of its *control variables* to keep the system in the *setpoint*, which usually represents its *correct* behavior.

In the last years, the use of controllers has become very common in robotics, critical systems and, in general, in the *embedded systems* contained in a growing number of everyday products. Therefore, the problem of the *automatic synthesis* of control systems is extremely important.

To this aim, several techniques have been developed, based on a more or less systematic exploration of the plant state space. One can mention, among others, *cell-to-cell mapping* techniques (Leu and Kim, 1998), *dynamic programming* (Kreisselmeier and Birkholzer, 1994) and *set oriented approach* (Grüne and Junge, 2005). Recently, *model checking* techniques have also been applied (Della Penna et al., 2006; Della Penna et al., 2007b) in the field of automatic controller generation.

The controllers generated using all these techniques are typically *numerical controllers*, i.e. tables indexed by the plant states, whose entries are commands for the plant. These commands are used to set the control variables in order to reach the setpoint from the corresponding states. Namely, when the controller reads a state from the plant, it looks up the action described in the associated table entry and sends it to the plant.

However, a main problem of this kind of controllers is the *size* of the table, which for complex systems may contain millions of entries, since it should be embedded in the control system hardware that is usually very limited.

A possible solution to these problems is to derive, from the huge numerical information contained in the table, a small *fuzzy control system*. This solution is natural since fuzzy rules are very flexible and can be adapted to cope with any kind of system. More-

over, there are a number of well-established techniques to guide the choice of fuzzy rules by statistical considerations, such as in Kosko space clustering method (Kosko, 1992), by abstracting them from a neural network (Sekine et al., 1995), by clustering the trajectories obtained from the cell mapping dynamics (Leu and Kim, 1998) and finally by using genetic algorithms (Della Penna et al., 2007a).

However, two considerations are to be made with respect to this approach. First, the detection of the fuzzy rules requires (at least) a *tuning phase* which is not completely automatic but involves some human intervention. Second, the fuzzy rules appear not to be suitable when a very high degree of precision is needed. This is the case, e.g., of the *truck and trailer parking problem* when *obstacles* are to be avoided in the parking lot (Della Penna et al., 2006). In this case, when the truck is near to an obstacle, only a very precise manoeuvre can park it, without hitting the obstacle. So, to achieve the required precision, an exceeding number of fuzzy rules would be necessary.

Therefore, it seems reasonable to pursue another possible approach, that is *to directly compress the control table*. By this we mean *data compression* in the usual sense of *reduction of the number of bits* needed to represent the controller table in the computer RAM memory (Cover and Thomas, 2006; Nelson and Gailly, 1995).

Observe, however, that in this case the compression algorithm should be constrained as follows:

1. the logical content of the table, that is the relation between the states and the corresponding control actions, must be preserved without any loss of information (i.e., we need some kind of *lossless* compression algorithm);
2. the access time to the table must be comparable with the one obtained with a direct representation of the table in the computer memory (i.e., the *decompression overhead* must be minimal).

Consider as an example the straightforward idea of directly compressing the table by mean of (some variant of) the well known Lempel-Ziv algorithm (Ziv and Lempel, 1977). Since this is a lossless compression, the first requirement above is fulfilled, but not the second one, since the access time to the table would be linear in the *uncompressed* size of the table itself (Nelson and Gailly, 1995).

Since all the common compression algorithms have similar problems when applied to controllers, we decided to develop a new *ad-hoc* compression algorithm based on the well known *Ordered Binary Decision Diagrams (OBDDs)* (Bryant, 1986).

Decision diagrams as well as *decision trees* are of wide use in data manipulation as well as in data mining (Hand et al., 2001). Indeed, our choice of OBDDs was motivated by their capability of compressing large state spaces, that is at the hearth of *symbolic model checking* and of its great achievements (Burch et al., 1992; Clarke et al., 1999). Our hypothesis has been that this capability is transferrable to the compression of the controller table which is, in a sense, an augmented state space representation.

Our experimental results show that this is indeed the case. For very large tables we obtain a quite good compression ratio (around 10:1). Moreover, not only the access time is good, but it is often *even better* than the one obtained by a direct representation of the table in memory. This phenomenon is due to the fact that, unless we accept a very sparse table, with huge waste of space, we need to represent the controller using some kind of *open addressing* table accessed through a *hash function*, with a consequent worsening of the access time (Cormen et al., 2001).

A final but very important point to be stressed is that the implemented compression algorithm, which transforms the controller table in an OBDD, operates in a completely automatic way. The only parameter the user should set is the BDD variable reordering method (see Section 4), which can effect the compression ratio. After that, the compression operates in a "zip"-like fashion. However, the best setting of the above parameter can also be discovered by the algorithm by trying all the possible reorderings, thus making the compression process completely automatic. Finally, a further automatic step can also be applied to transform the OBDD into C-code that can be incorporated in any C-program.

The paper is organized as follows: in Section 2 we give a description of OBDDs and in Section 3 we show how we use them to encode numerical controllers. In Section 4 we present our experimental results and Section 5 concludes the paper.

2 ORDERED BINARY DECISION DIAGRAMS

A Binary Decision Diagram (*BDD*) is a data structure used to represent a boolean function (Bryant, 1986). Indeed, any boolean function f can be represented as a binary tree having two kind of leaf values: F (boolean false) or T (boolean true). Each node of the tree (*decision node*) is labeled by a variable of the formula f. The two edges outgoing a decision node represent an assignment of the corresponding variable to false or true, respectively. Therefore, a path from the BDD root to one of its leaves represents a (possibly partial) variable assignment for f, and the cor-

responding leaf value is the value of f for the given assignment.

An advantage of BDDs is that many logical operations, like conjunction, disjunction, negation or abstraction can be implemented by polynomial-time graph manipulation algorithms. Indeed, BDDs are extensively used in the software tools, e.g., to synthesize circuits or to perform formal verifications.

If variables appear in the same order on all paths (or, in other words, if all the nodes on the same tree level refer to the same variable) the BDD is called *ordered* (OBDD).

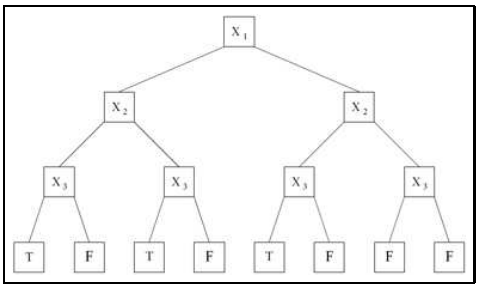


Figure 1: Representation of the boolean function $f(x_1,x_2,x_3) = \bar{x_1} \cdot \bar{x_2} \cdot \bar{x_3} + \bar{x_1} \cdot x_2 \cdot \bar{x_3} + x_1 \cdot \bar{x_2} \cdot \bar{x_3}$.

As an example, Figure 1 shows the OBDD for the boolean function $f(x_1,x_2,x_3) = \bar{x_1} \cdot \bar{x_2} \cdot \bar{x_3} + \bar{x_1} \cdot x_2 \cdot \bar{x_3} + x_1 \cdot \bar{x_2} \cdot \bar{x_3}$.

Usually, OBDDs are also *reduced* (ROBBD) by merging isomorphic subgraphs and eliminating nodes having two isomorphic children. In this way, any boolean formula f can be represented by an *unique* and very compact rooted, acyclic and directed graph.

The size of a reduced BDD is determined both by the function being represented and by the chosen *ordering of the variables*. Thus, a correct variable ordering is of crucial importance to gain the best "reduction factor". The problem of finding the best variable ordering is NP-hard, but there exist efficient heuristics to tackle the problem.

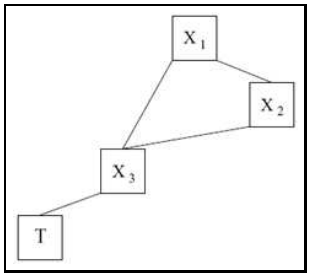


Figure 2: The OBDD in Figure 1 after removal of duplicate nodes and redundant tests.

Figure 2 shows the reduced version of the OBDD in Figure 1. Note that all the paths leading to F have been actually removed from the graph, since they are simply the complement of those leading to T.

A useful way to see BDDs, that will be used in this paper, is that they *encode the compressed representation of a relation*. However, unlike other compression techniques, the actual operations on BDDs are performed *directly on that compressed representation*, i.e. without decompression. Details on this issue are given in Section 3.

3 BOOLEAN ENCODING OF NUMERICAL CONTROLLERS

OBDDs can be used to create some kind of compressed representation of any data that can be encoded as a *logical expression*. In this Section, we describe how a numerical controller can be seen as a boolean formula, and give details on the actual OBDD-based compression algorithm that has been implemented. Moreover, we give some information on how the OBDD-compressed controller can be efficiently queried and transformed in a hardware-embeddable form.

3.1 Boolean Encoding of the Controller Table

As already said, an OBDD can be actually seen as a compact representation of a relation. On the other hand, a controller table containing a set of (state,action) pairs represents a relation $R = \{(s,a)|a$ is the action associated to s in the controller table$\}$ between states and actions. Since BDDs encode formulas, it may be useful to represent R through its characteristic function C_R defined as follows:

$$C_R(s,a) = \begin{cases} T & \text{if } (s,a) \in R \\ F & \text{otherwise} \end{cases}$$

Now, to write a definition of C_R as a boolean formula, we first have to represent its arguments, i.e., states and actions, in terms of logic variables. To this aim, we expand them to their binary memory representation.

Let suppose that states are n-bit values and actions are m-bit values. We write $s[i]$ and $a[i]$ to denote the ith bit of state s and action a,respectively.

Let $x_i,\ i = 1 \ldots n$ and $y_j,\ j = 1 \ldots m$ be $n+m$ boolean variables. A state s is then be represented by the formula

$$f_s(x_1,\ldots,x_n) = \bigwedge_{i=1\ldots n} l_i \text{ where } l_i = \begin{cases} x_i & \text{if } s[i]=1 \\ \overline{x_i} & \text{if } s[i]=0 \end{cases}$$

Each f_s is a boolean formula in n variables that is true if and only if its variables are assigned with the bits of s (denoting, as usual, the boolean true with 1 and the boolean false with 0). In the same way, an action a corresponds to the formula

$$f_a(y_1,\ldots,y_m) = \bigwedge_{i=1\ldots m} l_i \text{ where } l_i = \begin{cases} y_i & \text{if } a[i]=1 \\ \overline{y_i} & \text{if } a[i]=0 \end{cases}$$

.

Therefore, the controller characteristic function C_R can be encoded by the boolean formula

$$f_R(x_1,\ldots,x_n,y_1,\ldots,y_m) = \bigvee_{(s,a)\in R} (f_s \wedge f_a)$$

f_R is a boolean formula in $n+m$ variables that is true if and only if the variable assignment corresponds to the encoding of a (s,a) pair for which $R(s,a)$ holds.

For example, assume that the controller table contains the following 2-bit states $s = 00, s' = 01, s'' = 10$, with the following associated 1-bit actions $u = 0, u' = 0, u'' = 1$. Then the formula for the characteristic relation would be $f_R = \overline{x_1}\cdot\overline{x_2}\cdot\overline{y_1} + \overline{x_1}\cdot x_2\cdot\overline{y_1} + x_1\cdot\overline{x_2}\cdot y_1$.

3.2 Algorithm for the Logic Encoding of Numerical Controllers

```
BDD BDD_encoding(controller_table CTRL) {
 read number N of entries in CTRL;
 read number n of bits in each state of CTRL;
 read number m of bits in each action of CTRL;
 foreach j=1...n create boolean variable x_j;
 foreach j=1...m create boolean variable y_j;
 BDD f_R;
 foreach i=1...N {
  BDD E, f_s, f_a;
  foreach j=1...n //encode state bits
   if (bit(CTRL[i].state,j) == 1) f_s = f_s ∧ x_j;
   else f_s = f_s ∧ x̄_j;
  foreach j=1...m //encode action bits
   if (bit(CTRL[i].action,j) == 1) f_a = f_a ∧ y_j;
   else f_a = f_a ∧ ȳ_j;
  E = f_s ∧ f_a;
  f_R = f_R ∨ E; //disjunction of entries
 }
 return f_R;}
```

Figure 3: Algorithm for the logic encoding of numerical controllers.

The encoding algorithm, whose pseudocode is shown in Figure 3, implements the technique described in the previous Section.

After reading the number of bits in the controller states and actions, the `BDD_encoding` procedure creates the corresponding set of boolean variables x_i and y_i, respectively.

Then, for each entry of the controller table, a new BDD E is created as the appropriate conjunction of the state and action variables. In particular, the code checks every bit in the state and adds to the BDD f_s a conjunction with the corresponding variable, in its positive or negated form, based on the value of the bit. The process is then repeated for the action bits in the BDD f_a, and finally E is obtained as $f_s \wedge f_a$.

The BDD E is then added to the final BDD f_R using a disjunction. When all the controller entries have been processed, `BDD_encoding` returns f_R.

As we can see, the algorithm is actually very simple, since all the BDD manipulation is done by the external BDD package. In particular, our implementation uses the CUDD (CUDD Web Page, 2007) BDD manipulation package. Such package provides a large set of operations on BDDs and many variable dynamic reordering methods, which are crucial to gain the highest possible compression factor (Section 2).

Note that, to ensure the correctness of our approach, we also developed a parallel-query algorithm that tests for correctness and completeness the BDD-encoded controller f_R with respect to the original numerical controller *CTRL*. This procedure simply compares the results obtained by querying the uncompressed and the compressed controller with all the states in the controller table.

3.3 Querying the Encoded Controller

Once we have compressed the controller table in a BDD f_R, we must show how this information can be accessed by the controller itself to perform its task. There are several ways to do this.

A first method relies on the fact that the generated BDD actually encodes a function from states to actions. In other words, given a state, there is only one action associated to such state in the controller table. At the BDD level, this means that if we *restrict* the BDD by assigning to the variables $x_1 \ldots x_n$ the bits of a particular state s, then the resulting BDD has only one *satisfying assignment*, i.e., the one assigning to $y_1 \ldots y_m$ the bits of the corresponding action a.

Thus the action associated to a state can be found by applying two operations (restriction and then deduction of satisfying assignments) on the BDD of f_R. However, since controllers must work in the quickest

and simplest possible way, we may consider an alternative query method that requires less runtime overhead.

From the BDD of f_R, we extract m different BDDs $f_R^1 \dots f_R^m$, one for each bit of the action, in the following way.

$$f_R^i(x_1,\dots,x_n) = \begin{array}{l} \exists (y_1 \dots y_{i-1}, y_{i+1}, \dots y_m) \\ f_R(x_1,\dots,x_n, y_1 \dots y_{i-1}, 1, y_{i+1}, \dots y_m) \end{array}$$

In particular, in each f_R^i we existentially abstract all the action bit variables $y_j, j = 1 \dots m$ except for y_i, which is assigned to the constant 1. The resulting formula f_R^i has n free variables $x_1 \dots x_n$, which correspond to the state bits.

If we assign $x_1 \dots x_n$ to the binary representation of a state s, then f_R^i is true if and only if there exists an action associated to s whose ith bit is 1. Since the action associated to s, if it exists, is unique, f_R^i actually returns the ith bit of the action associated to s (assuming, as usual, that the logical true and false correspond to the values 1 and 0, respectively).

Having these functions at hand, we can rebuild the binary representation of the action a associated to the state s by simply applying each f_R^i to the encoding of s, without any further runtime BDD manipulation. The execution overhead is minimal in this case, since the "computation" of the BDD value for a particular variable assignment requires only a visit of the associated graph.

3.4 BDD into C Code Translation

Embedding and querying the BDD-compressed numerical controller within a small hardware or software device is also an issue that can be addressed in various ways.

Obviously, we cannot require CUDD or another BDD-manipulation package to be present in the controller. However, a BDD (or the set of BDDs obtained using the technique described in Section 3.3) can be simply translated in a C code fragment composed by nested if-then-else statements.

The translation process is very straightforward and requires only a visit of the OBDD graph. Let n be a node of the OBDD associated with the logic variable $V(n)$, and let $T(n)$ and $E(n)$ be the two children of n for $V(n) = true$ and $V(n) = false$, respectively. Then, we can define the C-translation of n as follows:

$$CT(n) = \begin{array}{l} \texttt{IF } (V(n)) \\ \texttt{THEN } CT(T(n)) \\ \texttt{ELSE } CT(E(n)) \end{array}$$

This translation is linear in terms of the required space, and the resulting representation can be easily embedded in a (small) hardware device resulting in good time performances.

4 EXPERIMENTAL RESULTS

To setup our experiments, we first have to fix some BDD encoding parameters, namely the variable ordering in the boolean formulas and the dynamic reordering method used by the BDD package.

Indeed, the BDD structure and therefore the compression ratio can be influenced by the original ordering of the variables in the boolean formulas presented to the BDD package. In particular, we recall that the variables in our BDDs are the state bit variables, namely $x_i, i = 1 \dots n$, and the action bit variables, $y_i, i = 1 \dots m$. Thus, we may consider the variable orderings arising from all the possible combinations of the following conditions:

- the state bit variables and the action bit variables can be ordered with different endianness, that is from the most significant bit to the least or viceversa;
- the state bit variables can be placed before the action bit variables, after them or interleaved.

Namely, we can write the function f_R of Section 3 with any of the ten variable orderings $O1 \dots O10$ shown in Table 1. Note that in $O9$ and $O10$ we assume $n > m$.

Table 1: Possible initial variable orderings.

O1	$x_1 \cdots x_n y_1 \cdots y_m$
O2	$x_1 \cdots x_n y_m \cdots y_1$
O3	$x_n \cdots x_1 y_1 \cdots y_m$
O4	$x_n \cdots x_1 y_m \cdots y_1$
O5	$y_1 \cdots y_m x_1 \cdots x_n$
O6	$y_1 \cdots y_m x_n \cdots x_1$
O7	$y_m \cdots y_1 x_1 \cdots x_n$
O8	$y_m \cdots y_1 x_n \cdots x_1$
O9	$x_1 y_1 x_2 y_2 \cdots x_m y_m x_{m+1} \cdots x_n$
O10	$x_n, y_m x_{n-1} y_{m-1} \cdots x_{m-n} y_1 x_{m-n-1} \cdots x_1$

Moreover, variables can be dynamically reordered by the BDD package during the construction of the final BDD. In our experiments, we used the fourteen dynamic reordering methods $R1 \dots R14$ offered by the CUDD package and shown in Table 2.

In the following experiments we tried all the possible combinations between initial variable orderings and dynamic reordering methods, choosing each time the one that produces the better compression ratio.

All the experiments were performed on a 2.66GHz Pentium 4 with 1GB RAM.

Table 2: CUDD variable reordering methods.

R1	Random reordering
R2	Random pivot reordering
R3	Sift
R4	Converging sifting
R5	Symmetric sifting
R6	Converging symmetric sifting
R7	Group sifting
R8	Converging group sifting
R9	Window permutation (size 2)
R10	Window permutation (size 3)
R11	Window permutation (size 4)
R12	Converging window permutation (size 2)
R13	Converging window permutation (size 3)
R14	Converging window permutation (size 4)

4.1 Inverted Pendulum Controller

The first case study is the numerical controller for the inverted pendulum problem (Kreisselmeier and Birkholzer, 1994), where the controller has to bring the pendulum to equilibrium by applying a torque in the shaft.

The optimal numerical controller was generated using the dynamic programming method described in (Kreisselmeier and Birkholzer, 1994). Table 3 shows the details of such controller. In the table, row *Entries* indicates the number of controller entries (i.e., controlled states) and row *Size*-column *Normal* indicates the controller table size in Kilobytes. Moreover, to compare the effectiveness of the BDD compression with respect to the common compression techniques, we also show the size of the controller table compressed by the LZ77-based (Ziv and Lempel, 1977) algorithm of gzip (GZip Web Page, 2007) (row *Size*-column *LZ*, where the relative size of the compressed file is also shown as a percentage) and the average controller access time in milliseconds for both the uncompressed and the compressed representations (row *Time*-column *normal* and row *Time*-column *LZ*, respectively). Note that, as we expect, access times for the LZ-compressed controller are very high since, in the worst case, the controller must be completely decompressed to find the required table entry.

Table 3: Numerical controller for the inverted pendulum.

	Normal	LZ
Entries	311618	
Size	3043	1295 (42.6%)
Time	8 ms	657 ms

When BDD compression is applied to the controller, we obtain BDDs whose size (in number of graph nodes) is shown in Table 4 for each combination of variable initial ordering ($O1 \dots O10$) and dynamic reordering ($R1 \dots R14$).

Here, the smallest BDD (indicated by the bold number in Table 4) has 61584 nodes. The actual size in Kilobytes and the average access time for the BDD-compressed controller are shown in Table 5. As we can see, BDD compression reduces the controller 26.5% more than LZ, and has also better access times than the LZ-compressed version, since it does not require any decompression to read the table entries.

Table 5: BDD-compressed numerical controller for the inverted pendulum.

	BDD
Entries	311618
Size	489 (16.1%)
Time	1 ms

4.2 Truck and Trailer Obstacles Avoiding Controller

In the second case study, we consider the controller for the *truck and trailer obstacles avoiding* problem. Namely, the controller has to back a truck with a trailer up to a specified parking place starting from any initial position in the parking lot. Moreover, the parking lot contains some obstacles, which have to be avoided by the truck while maneuvering to reach the parking place. Corrective maneuvers are not allowed, that is the truck cannot move forward to *backtrack* from an erroneous move.

Table 6: Results for the truck and trailer obstacles avoiding controller.

	Normal	LZ	BDD
Entries	3256855		
Size	71650	22644 (31.6%)	7038 (9.8%)
Time	89 ms	3173 ms	108 ms

The numerical controller was generated with the CGMurφ tool (CGMurphi Web Page, 2006; Della Penna et al., 2007b). Results are in Table 6. As we can see, the controller has a very big size. However, the best BDD compression scheme ($O5$,$R5$) is able to reduce the size of the controller up to a 90.2% space savings, that is 21.8% more than using LZ77 compression. Moreover, the BDD compression continues to win also with respect to the access times.

Table 4: Number of nodes in the BDD for the inverted pendulum controller with respect to different variable orderings.

	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
R1	92490	108474	65382	64012	156872	83808	149943	86957	92013	145877
R2	142012	145654	66711	78029	130315	79575	144583	89528	148037	142432
R3	65842	61588	65842	61584	138377	75360	145181	143181	65854	127241
R4	65842	71314	70691	61584	130355	71313	75427	135580	65842	65843
R5	65842	65848	70774	70770	144310	70773	144422	70776	65865	142677
R6	65842	65860	70777	70770	135575	70773	135573	70773	65860	130348
R7	66759	66555	92830	61798	144268	61586	136870	61594	69198	65865
R8	65842	61594	75420	61584	145172	61586	135575	61586	65868	65863
R9	92295	92279	65441	61873	163149	121083	255309	121307	101164	178482
R10	76724	77231	**61584**	61588	134758	122539	143860	78044	79227	142072
R11	72724	72781	61584	61584	134023	122157	134254	106597	77295	134723
R12	81329	98992	61630	61641	134162	128731	138481	79723	86284	136761
R13	77034	65868	61584	61584	134023	123547	134275	111460	66122	134257
R14	65842	65868	61584	61584	134275	77960	134275	116568	68455	134079

4.3 Inverted Pendulum on a Cart Controller

The last case study is the numerical controller for the inverted pendulum on a cart problem (Junge and Osinga, 2004). The system consists of a planar inverted pendulum on a cart that moves under an applied horizontal force, constituting the control. The position of the pendulum is measured relative to the position of the cart as the offset angle from the vertical upright position. The controller goal is to set such angle to zero.

Table 7: Results for the inverted pendulum on a cart controller.

	Normal	LZ	BDD
Entries	151394		
Size	1478	90 (6.1%)	215 (14.6%)
Time	3 ms	206 ms	1 ms

The numerical controller was generated with the CGMurφ tool. Results are in Table 7. The number of controller entries is very small with respect to the previous two case studies. We see that on a small controller the BDD compression has a lower compression ratio than LZ, but always better access times (1ms vs. 206ms), since it does not require any decompression to read the table entries.

5 CONCLUSIONS

In this paper, we presented an OBDD based compression algorithm for numerical controllers. The compression algorithm is completely automatic and can be applied to the (state,action) tables generated by any numerical controller generation technique.

Our experiments show that this new algorithm has a very high compression ratio (up to 10:1), that is often more than the ratio obtained on the same data by the most common lossless compression techniques, such as LZ77. Indeed, OBDD compression is not actually lossless, but rather "relation-invariant". That is, the OBDD compression leaves intact the behavior of the state-action relation stored in the table. However, by working on the logic encoding of the relation, the OBDD is capable of *optimizing the representation of the relation*, so reducing its size.

Moreover, accessing the entries of an OBDD-compressed numerical controller does not require any data decompression (as it would happen, e.g., with LZ77), so the controller performances are very good. Also in this case, the optimized representation generated by the OBDD sometimes allows to achieve access times that are even better that those of an hash function used on the uncompressed table.

Therefore, the BDD compression is a technique that can be actually exploited to reduce the size of numerical controllers, generating a compact structure that is easy to store and query. This would allow, e.g., to use *high precision controllers* even in *limited devices*.

Indeed, we are currently studying and implementing algorithms that create *embeddable and executable* forms of the OBDD-compressed controllers. The OBDD-to-C methodology sketched in section 3.4 is only the first step, as we intend to design a translation process to directly create an optimized VHDL (Pedroni, 2004) circuit description from the OBDD. In this way, we would have a completely automatic methodology to generate, from any numerical controller, a small executable definition ready to be em-

bedded in the hardware.

Finally, a last point we want to investigate is the relationship between the compression obtained with the use of the OBDDs and the reduction of fuzzy control systems by means of a *hierarchical* approach (see e.g. (Stufflebeam and Prasad, 1999)). Indeed, there is an evident correspondence between, on one hand, the search of the best ordering of variables needed for the OBDD compression and, on the other, the hierarchical decomposition into subsystems of a fuzzy system.

REFERENCES

Bryant, R. (1986). Graph-based algorithms for boolean function manipulation. *IEEE Trans. on Computers*, C-35(8):677–691.

Burch, J. R., Clarke, E. M., McMillan, K. L., Dill, D. L., and Hwang, L. J. (1992). Symbolic model checking: 10^{20} states and beyond. *Inf. Comput.*, 98(2):142–170.

CGMurphi Web Page (2006). http://www.di.univaq.it/magazzeni/cgmurphi.php.

Clarke, E. M., Grumberg, O., and Peled, D. A. (1999). *Model Checking*. The MIT Press.

Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory*. Wiley.

CUDD Web Page (2007). http://vlsi.colorado.edu/ fabio/.

Della Penna, G., Fallucchi, F., Intrigila, B., and Magazzeni, D. (2007a). A genetic approach to the automatic generation of fuzzy control systems from numerical controllers. In *AI*IA*, volume 4733 of *LNAI*, pages 230–241. Springer-Verlag.

Della Penna, G., Intrigila, B., Magazzeni, D., Melatti, I., Tofani, A., and Tronci, E. (2006). Automatic generation of optimal controllers through model checking techniques. In *Proceedings of 3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO2006), to be published in Informatics in Control, Automation and Robotics III, draft available at the url http://www.di.univaq.it/magazzeni/cgmurphi.php*.

Della Penna, G., Magazzeni, D., Tofani, A., Intrigila, B., Melatti, I., and Tronci, E. (2007b). Automatic synthesis of robust numerical controllers. In *ICAS '07*, page 4. IEEE Computer Society.

Grüne, L. and Junge, O. (2005). A set oriented approach to optimal feedback stabilization. *Systems Control Lett.*, 54(2):169–180.

GZip Web Page (2007). http://www.gzip.org/.

Hand, D. J., Mannila, H., and Smyth, P. (2001). *Principles of Data Mining*. MIT Press.

Junge, O. and Osinga, H. (2004). A set oriented approach to global optimal control. *ESAIM Control Optim. Calc. Var., 10(2):259–270 (electronic), 2004.*

Kosko, B. (1992). *Neural Networks and Fuzzy Systems*. Prentice Hall.

Kreisselmeier, G. and Birkholzer, T. (1994). Numerical nonlinear regulator design. *IEEE Transactions on Automatic Control*, 39(1):33–46.

Leu, M. C. and Kim, T.-Q. (1998). Cell mapping based fuzzy control of car parking. In *ICRA*, pages 2494–2499.

Nelson, M. and Gailly, J. (1995). *The Data Compression Book*. MT Books.

Pedroni, V. (2004). *Circuit Design with VHDL*. MIT Press.

Sekine, S., Imasaki, N., and Tsunekazu, E. (1995). Application of fuzzy neural network control to automatic train operation and tuning of its control rules. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1993*, pages 1741–1746, Yokohama.

Stufflebeam, J. and Prasad, N. (1999). Hierarchical fuzzy control. In *Proceedings of IEEE International Fuzzy Systems Conference*, pages 498–503.

Ziv, J. and Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343.

ENERGY MODEL BASED CONTROL FOR FORMING PROCESSES

Patrick Girard
Industrial Materials Institute, National Research Council, 75 Boulevard de Mortagne, Boucherville, Canada
Patrick.Girard@nrc.ca

Vincent Thomson
Mechanical Engineering, McGill University, 817 Sherbrooke St.W. Montreal, Canada
Vincent.Thomson@mcgill.ca

Keywords: Model based control, forming energy, simulation, real time identification, intelligent agents, on-line diagnostics.

Abstract: Thermoforming consists of shaping a plastic material by deforming it at an adequate deformation rate and temperature. It often exhibits abrupt switches between stable and unstable material behaviour that have neither been identified nor controlled up to now. PID control, although adequate for simple parts, has not been able to control very well the forming of complex parts and parts made of newer materials. In this paper, the state parameters that allow the development of predictive models for the forming process and the construction of control systems are identified. A robust, model based control system capable of in-cycle control is presented. It is based on a simulator continuously tuned and supported in real time by intelligent agents that incorporate diagnostic capabilities.

1 INTRODUCTION

Forming processes are widely used in a number of industries, including automotive, aerospace and home appliances. Forming is an apparently simple process in which a sometimes pre-shaped sheet of plastic material is first heated to the correct forming temperature in a first phase, and then deformed in a second phase at the correct strain rate, generally by pressing it against a mould to impart a specific shape. The deformation of the sheet is insured by using either a vacuum or pressure at a given temperature and deformation rate, sometimes with the assistance of a mechanical plug. After the part is ejected from the mould some additional, post-processing steps may be required, such as cooling at a controlled rate, or annealing to relieve the built in stresses that were induced by this transformation process.

Effective control of forming needs to address the following issues.

- How energy is transferred to the part can be transformed into two separate processing steps, first to bring it to the correct forming temperature, and then to shape it (Figure 1).
- Depending on the rate that the material deforms, variations in the deformation rate produce enormous changes in the viscosity of the material, resulting in very high and unstable variations of the energy required for deformation as shown in Figure 2.

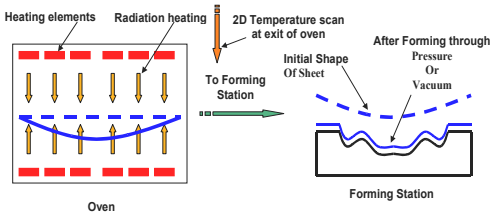


Figure 1: The thermoforming process (Girard et al., 2005).

Up to now forming has been controlled in a very empirical and indirect manner. For example, during the heating phase of the sheet only the temperature of the heating elements has been controlled. The rate of deformation during forming is controlled by applying pressure on the material either as a constant

air or hydraulic pressure, or as the result of a semi-controlled explosion. As a result (Figure 2), the forming process is seen as a seemingly random succession of stable and unstable phases where the triggering point from stability to instability is often neither identified nor taken into account. This makes it very difficult to ensure robustness.

This paper proposes a model based control system based on a simulator that predicts the process energy requirements. A similar approach has been successfully applied to the control and on-line optimization of metal powder grinding (Albadawi et al., 2006).

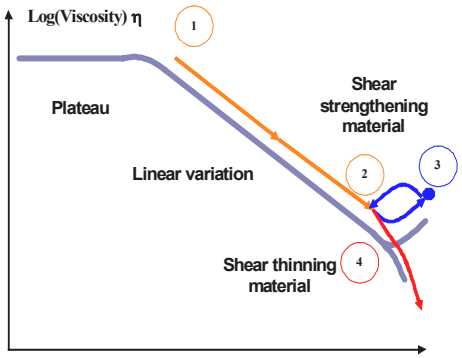


Figure 2: Typical variation of viscosity for thermoplastic materials as forming pressure is applied.

Once a relatively steady state is attained the simulator is tuned on-line and in real time by a number of intelligent agents that identify drifts and variations of the process. The tuned simulator is then used to generate a linear sensitivity matrix and it is upon this matrix that the control model is built, and it provides the response time required for in-cycle control, i.e., while the part is being made. Although the forming process is non-linear, linear control is quite adequate since the operating point predicted by the simulator is close to the actual operating point.

Also, the simulator by itself can actually predict and control the dynamic startup phase of the process. The startup procedure for thermoforming complex, technical parts, for example, can result in up to 5 rejected parts costing $100 each in material (from the thermoforming company, PlastikMP, Richmond, Quebec, Canada).

Further to this introduction, the thermoforming process along with key process parameters needed for effective control is described in Section 2. The present situation for control of thermoforming is given in Section 3. Section 4 outlines the model based control system. Process parameters that can be identified in real time are listed in Section 5. Section 6 presents the real time diagnostic capability of the system, and finally, there is a brief conclusion.

2 IDENTIFICATION OF STATE VARIABLES FOR CONTROL OF THE FORMING PROCESS

The first task is to identify the state variables of the thermoforming process that can be used to control the heating and the forming phase.

2.1 Heating Phase

The purpose of the sheet heating phase is to bring the whole sheet above the minimum forming temperature while remaining below the maximum allowable forming temperature, i.e., be within the forming 'window'. By knowing this, the minimum and maximum amount of energy required for the heating process can be easily calculated. It is also very amenable to use energy as a control parameter since process energy is the main variable.

This means that the in-oven heating cycle can stop when the required energy has been transferred to the plastic sheet. However, the temperature profile inside the sheet still has to be appropriately distributed (usually uniform). This is presently realized in the real world by allowing the sheet to stand for a while outside the oven before forming.

2.1.1 Energy Transfer to the Sheet during Sheet Heating

Representing this transfer of power with transfer functions allows representation of the state of the system by using either the power or the temperature (Figures 3 and 4). It is a feature required of the control system since the operator needs to view the machine parameters for this phase in the usual manner, which is a temperature display in this case.

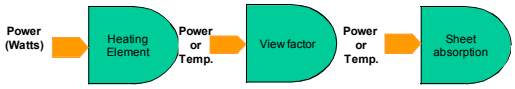


Figure 3: The heating phase as a cascade of energy or temperature transfer functions.

2.1.2 Heat Flux Matrix during the Heating Phase (View Factor)

In the thermoforming process a sheet of material is positionned in an oven and heated by an array of heating elements (shown at the right of Figure 5).

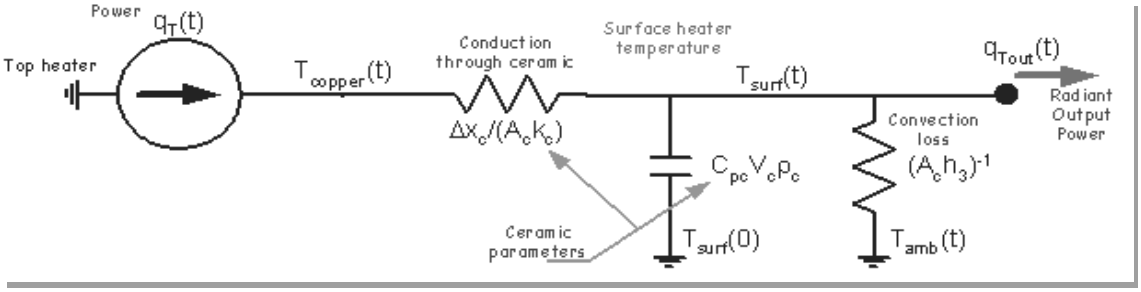
The view factor defines the relation W_{ij} between the heat flux produced by heating element i and the radiative power absorbed by sheet zone j in equation (1). This heat flux can be measured by a sensor as has been demonstrated by Kumar (2005).

$$[Welement_i] \quad [W_{ij}] \quad = [Wzone_j] \qquad (1)$$

The left part of Figure 5 presents the temperature map at the exit of the oven with the holes provided for the heat flux sensors appearing in black. In this picture the heat flux sensor was located at the center.

2.1.3 Energy Absorption by the Sheet (Process Parameter)

Energy is transferred throughout the sheet by two mechanisms: conduction from the surface and radiation absorption. The two related material parameters are the conductivity and the absorptivity of the material, respectively. A major source of uncertainty in the process stems from the fact that these parameters can vary widely from batch to batch, especially for absorptivity which can vary enormously when the colorant supplier is changed for example, and techniques were designed to detect on-line variations in these parameters.



$$q_T = \frac{1}{R_{cond}}\left(T_{copper} - T_{surf}\right) = \frac{R_{conv} C_{surf} s + 1}{R_{conv}}\left(T_{surf} - T_{amb}\right) + q_{Tout}$$

$$C_{surf} = C_{pc} V_c \rho_c \quad R_{cond} = \frac{\Delta x}{A_c k_c} \quad R_{conv} = \left(A_c h_3\right)^{-1}$$

Figure 4: Heating element transfer function (Gauthier et al., 2006).

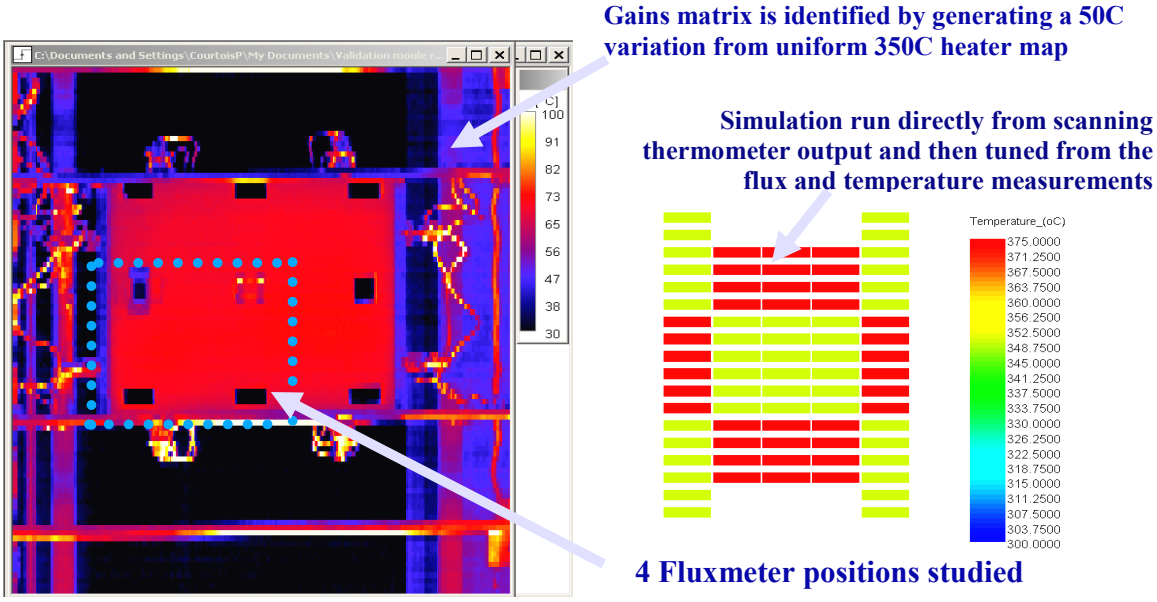


Figure 5: Sheet heat map at the exit of oven and heating element array (Girard et al., 2005).

In general, the temperature increase in a given zone of the sheet and at a given depth for a steady heat flux can be represented with very good precision by the following empiric equation where θ is temperature, t is time, and d is depth into the sheet (Girard et al., 2005).

$$\theta_{t,d} = \exp\left(\left(\frac{a_1}{(1+d)} + \frac{a_2}{t}\right)^{*a_3}\right) \quad (2)$$

Using this equation means that we know the constant heat flux that is required to heat the sheet to a given temperature θ_1 at time t and depth d. The constants a1, a2 and a3 are determined by the fit with modelled data. Figure 6 presents the variation of temperature for a steady heat flux (constant heating element temperature) together with the repeated adjustments needed by a PID controller.

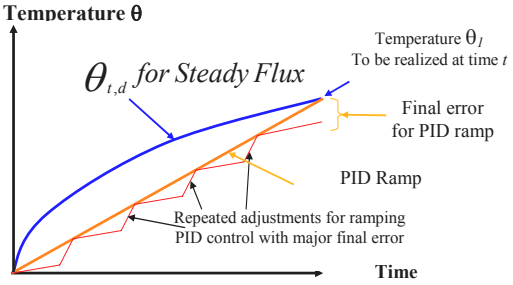


Figure 6: PID ramp and model based temperature control comparison. Model based control achieves final state temperature with more accuracy and less adjustment during heating.

2.1.4 Absorptivity (Material Parameter)

The measurement of the start of heating for a virgin (i.e., not colored) high density polyethylene thick sheet reveals that even at a depth of 11 mm the material temperature starts to increase nearly immediately with the start of radiative sheet heating (Figure 7). Since conduction heating requires several minutes to get to this depth, the only heating mechanism that can allow for such a behavior is radiation absorption.

The energy absorbed, q_{absn}, in layer 'n' contributes to the internal temperature change according to:

$$F_1\alpha(1-\alpha)^n = q_{absn} = \rho c_p \frac{\partial T}{\partial \tau} x \quad (3)$$

where F_1 is the heat flux at the surface of the sheet, α is the absorptivity, ρ is the sheet density, c_p is the heat capacity, ∂T is the sheet temperature difference during time $\partial\tau$, and x is the n^{th} layer thickness. From equation (4) (Kumar, 2005), the absorptivity α can be calculated from the slopes of the temperature increase measured at layers 3 and 1.

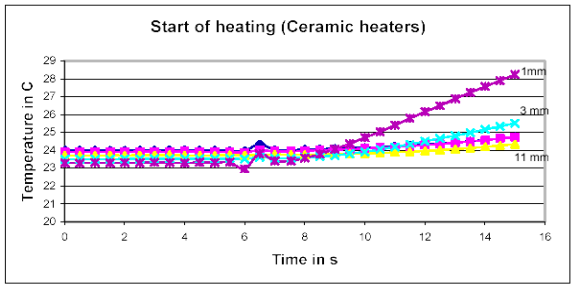


Figure 7: An enlargement of the start of sheet heating (Girard et al., 2005). The data show the temperature increase with time at different depths in a plastic sheet being heated from the surface.

$$\alpha = 1 - \sqrt{\frac{(\frac{\partial T}{\partial \tau})_3}{(\frac{\partial T}{\partial \tau})_1}} \quad (4)$$

2.1.5 Heat Capacity (Material Parameter)

The heat capacity C_p is evaluated during the cooling phase from the cooling rate with a given heat transfer coefficient at the sheet surface (Figure 8) (Zhang, 2004). The total energy, q_{tot}, can also be determined from the heat transfer coefficient as follows

$$q_{tot} \cong q_{conv} \cong \rho C_p x \frac{\Delta T}{\Delta \tau} \quad (5)$$

where the energy from the heating elements hitting the sheet is the convection energy, q_{conv}.

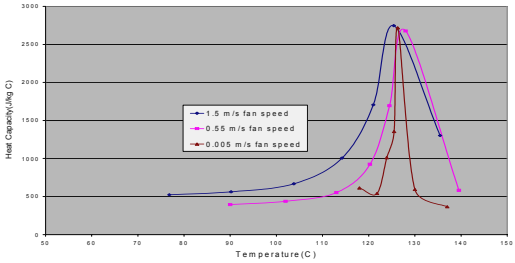


Figure 8: Experimental heat capacity curves determined by different cooling rates obtained on-line by varying fan speed (bottom heating of the sheet at 280°C).

Please note in Figure 8 that different cooling rates predict mostly the same heat capacity, and that the shape of this peak is directly related to the level of crystallinity of the material.

2.1.6 Elastic Modulus (Material Parameter)

During the re-heat phase of the thermoforming process the frozen in stresses induced in the material during the calendering process are relaxed. This results in anisotropic shrinkage of the sheet that causes variation of the sag in the heating oven. The forming of the shrunk sheet results in variations of the final thickness of the produced part. Also, the sag during heating must be adequately controlled since it can result in catastrophic variations in the distance from the sheet to the heating elements.

The elastic modulus, E, and level of frozen-in stresses are the main predictors for sag and shrinkage during the heating phase of the thermoforming process.

It is however difficult to get adequate data for process control and simulation purposes given the variability of sheet material properties from batch to batch and the fact that the variation of E is difficult to evaluate by the usual techniques in the vicinity of the melting point of the material where the experimental data reveals a very sharp inflection point related to the phase change of the material. Since forming takes place in this temperature region simulation models are quite deficient in this crucial area.

To address these issues an on-line identification technique was developed that uses two different steps of the blow forming process (Bahadoran, 2005). The low temperature variation of the elastic modulus is identified from the development of the sag at the entry in the oven (Figure 9) allowing for a better evaluation of E near the melting point. However, an existing forming mould can be used to produce a bubble on-line on the actual forming machine, and the value of the elastic modulus is characterized near the transition point of the material from the variation of the bubble and the blow pressure (Figure 10) (Bahadoran, 2005).

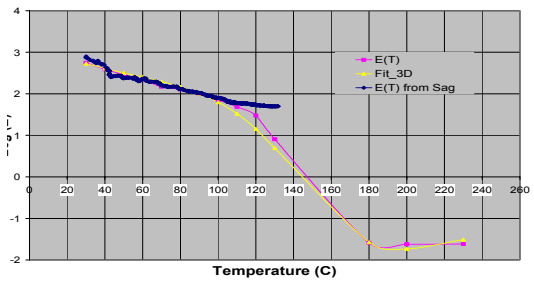


Figure 9: Experimental data obtained on-line showing the variation of the elasticity modulus with temperature at lower temperatures.

This provides a much better definition of the 'elbow' zone of E versus temperature. This technique requires minimal additional instrumentation to an existing machine. Also most any existing mould can be used for this purpose.

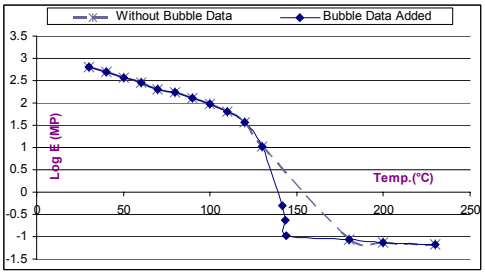


Figure 10: Experimental data obtained on-line showing the variation of the elasticity modulus with temperature, particularly near the melting point.

2.2 Forming Energy during the Forming Phase

Referring to Figure 2, after the sheet has been heated the forming process begins at point 1 by applying a constant pressure from one side of the sheet to be formed. The material starts taking a more or less spherical shape and its thickness diminishes. As the shear rate of the material increases, the viscosity of the material drops. Since the input pressure remains relatively constant, this results in an unstable and very fast evolution towards point 2 when the shear rate rises above a triggering level.

After this point, the forming process can behave either in a stable or an unstable manner depending on the type of thermoplastic.

- If the material is shear strengthening, deformation under constant pressure is relatively easy to control since it will be mostly spherical and bounded at point 3 and then revert back to a stable behaviour. Also, since the deformation is self-stabilizing the shape of deformation tends to be spherical. This is how the blow moulding of PET (polyethylene terephthalate) materials is controlled, for example.
- If the material is shear thinning, a pressure control scheme results in the sheet being ripped apart in an explosive manner. Also, any deformation that starts at a given location typically 'grows' in a random direction and pattern. In this case, the forming process needs to be either bound geometrically by a mould or by the flow rate that is applied to 'blow' the sheet.

Figure 11 presents how the volume and pressure develop for the free blow (forming without a mould) of a PET bottle. It can be seen that the formed part starts by expanding in a smooth manner in phase 1 until it suddenly expands very rapidly in phase 2. Since PET is shear strengthening it will then consolidate in phase 3.

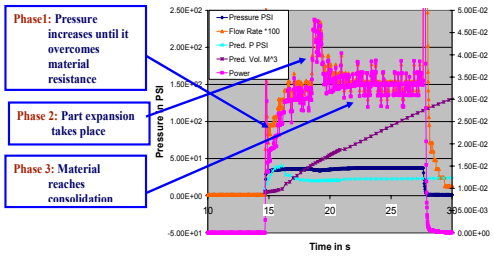


Figure 11: Measuring the pressure/volume relationship for free blow (blowing without a mould) of a PET bottle

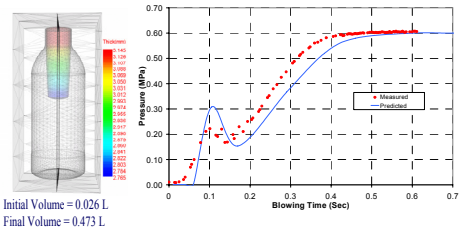


Figure 12: Predicted and measured pressure with the $W_{forming}$ simulation approach (PET bottle in mould).

It is easy to measure on-line the pressure and the flow rate for these processes, and recent developments show that simulations based on the forming energy are much more accurate than those relating constitutive equations to initial temperature and pressure conditions on the sheet (Figure 12) (Mir et al., 2007). Also, minimizing the amount of energy required by the process, which allows for the use of smaller machines, is often one of the objectives of the control system.

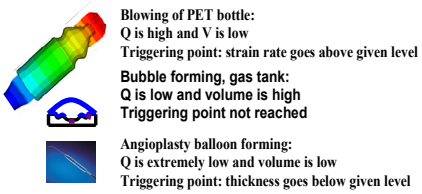


Figure 13: Development of volume flow Q and volume V for some thermoforming processes.

It must be noted that the start and development of phase 2 (Figure 11) is not predicted by the usual techniques, but that it poses no problem with the $W_{forming}$ approach (Figure 12). Also, the 'trigger' point (Figure 13) that starts the expansion phase is not predicted at all with the regular modeling approach.

3 CONTROL TECHNIQUES PRESENTLY USED FOR THERMOFORMING

The control techniques presently used are typically based on indirect sheet temperature control and a forming pressure that is set to a constant value. These techniques have major drawbacks.

- They do not directly control the main parameters of the process.
- They do not monitor and do not control adequately the primary process variables. This is especially true of the forming pressure since it is only regulated at the entry of the mould, and often very imprecisely.
- They do not identify nor take into account the switch point from stability to instability.
- During the unstable phases of the process, minor variations in the input variables of the process can develop into chaotic variations in the end process. These variations are presently not detected and are only taken into account in the system indirectly. They can be:
  - material properties that vary from batch to batch,
  - environmental variations such as ambient temperature or air flow, and
  - variations in machine parameters such as heating elements output or line pressure.
- Present temperature controllers, such as implemented by MAGI Control (Montreal, Canada), use a PID controller to track a ramp that is calculated from the θ_1 temperature to be realized. Figure 5 presents a comparison of the model based and PID ramping approach based on the results obtained by MAGI Control.
- These process uncertainties are compounded by the new 'designer' materials that typically have a very narrow processing window, and also by the very tight dimensional requirements that are required of technical parts

It can be seen that PID control based on a ramp requires repeated adjustments during the heating phase and ends up with a considerable final error. Model based control however only requires the adjustment of the heat flux by integration of the heating curve, which achieves much smoother control and better final temperature precision.

4 MODEL BASED CONTROL SYSTEM

In place of PID control, we are proposing to use a model based control system that is continuously tuned and based on on-line identification of the main parameters of the forming processes that were presented in Section 2. This 'tuning' is achieved by intelligent agents as defined by (Weiss, 1999): "Agents are autonomous, computational entities which sense their environment either by physical or virtual sensors, and then initiate actions by actuators and/or by communicating with other agents." In our case each agent is a fast response routine that monitors a specific aspect of the process, for example, the variation of the specific heat of the material. If this variation is above a specific level, the agent contacts the main system so that the process parameters are adjusted to reflect the change.

4.1 General Specifications of the System

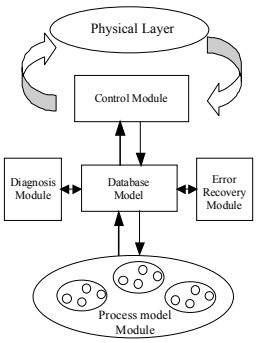


Figure 14: General architecture of the model based control system.

4.1.1 Objective

- Implementation of a generalized controller that is based on a model of the process to be controlled and that updates and tunes the process model in real time.

4.1.2 Inputs

- Real time measurements of the process and equipment data.
- Results of quality control.
- Process submodels implemented as intelligent agents that identify and track parameters and that calculate process state parameters.

4.1.3 Outputs

- Updated control model with control parameters that are sent to the forming process.
- Process and material parameters are estimated during part manufacture and are updated after each part is made.
- Diagnostics of the process are executed during part manufacture.

4.2 Main Control Module

4.2.1 Simulator Agent

The first step for the creation of a model based control system using intelligent agents is to build an accurate process model and simulator. The process model identifies critical state variables and the simulator predicts the parameter adjustments required for the desired outcome from the state variable history. A finite element simulation of the thermoforming process based on the $E_{forming}$ energy in equation (3) is very easy to correlate to real time machine measurements of the flow rate and pressure. This simulation typically requires:

- the geometric description of the part, machine and moulds,
- a material database for the rheological and physical properties of materials, and
- the processing parameters for the part, often called recipe by the manufacturer.

Also, two main challenges need to be addressed for adequate control.

- Inaccessible internal sheet temperatures need to be controlled precisely within the forming window for part quality.
- The execution time frames of the different agents need to be adjusted and synchronized.

4.2.2 Simulation of a Virtual Sensor

The tuned simulation can function as a virtual sensor, also called a soft sensor. The integrated history of the sheet surface temperature as measured directly by infrared sensors with the heat transfer simulation to predict the sheet internal temperature and to indicate the time required for adequate sheet heating inside the oven.

One of the main outputs of the simulation is the predicted heating curve for the material for a constant heat flux (Equation (2)) shown in Figure 6. From this, the system can control the heat flux by controlling the heating element input power, which results in a given temperature at a given depth at time *t* in the material.

4.2.3 Adjusting the Time Frames of the Different Modules

Unfortunately, the time frame of such a simulation is orders of magnitude greater than what is needed to control the thermoforming process in real time. This problem is solved by generating the sensitivity matrix of the simulation every time the simulation is updated. For example, the heat flux in equation (3) can be generated from the tuned simulation. Equations (2, 3) are used to predict the sheet temperatures at different depths for different sheet zones for the time sequenced trajectories of the heating element energy input allowing for heat flux control as shown in Figure 5.

Since the simulation is reasonably accurate, the control system only needs to apply the updated parameters in the vicinity of the initial prediction for the calculated operating point, which allows the use of linear interpolations to adjust the operating point as required, thus achieving a very fast response time.

4.3 Agent based Control

An intelligent agent based system is a loosely coupled network of problem solving entities (agents) that work together to find the answer to problems that are beyond the individual capabilities or knowledge of each entity (Florez-Mendez, 1999). An agent based system was chosen for model based control since it can deal well with multiple submodels and data streams, and can cope with submodels that are very different in size and that operate on dissimilar time scales. These features make agent technology especially suited for building control systems based on models of processes, where the processes are very complex and many process and material parameters are dynamic.

All agents in the architecture operate independently and asynchronously. The control agent acquires sensor data from the physical layer and sends control parameters as they become available. Similarly, the process agents retrieve sensor data and calculate state variables. The retrieval of sensor data and the calculation of state variables are interrupt driven based on detected variations from previous states; thus, calculations are only launched when needed and with the best information available. This design minimizes control cycle time while allowing data to flow asynchronously and implements just in time delivery of the different data streams, while still setting control parameters with complex, but validated parameters. The result is that during a short production period certain parameters are updated infrequently. This is not a problem, since they do not highly impact the operating point of the process. This architecture allows many processes to be controlled in-cycle, i.e., while a part is being made so that near perfect parts can be made every time. If the process is very fast or parameters cannot be measured during part manufacture, cycle-to-cycle control can be done, i.e., parameters measured during or after a part is being made are used to control the next part being manufactured.

Control can be done with a single processor, if the amount of computation is small. Nevertheless, for a complex process like thermoforming, the amount of calculation for process models tends to be large and distributed over different time frames; therefore, multiple processors may be required depending on the complexity of the heating process. With multiple processors, the control system can dynamically allocate the execution of different agents to different processors. Due to the asynchronous operation of the architecture, processes can be optimally controlled for submodel execution times from milliseconds to hours.

5 PARAMETERS IDENTIFIED ON-LINE

If the agent in charge decides that the drift or variation of the parameter warrants an adjustment of the simulator, parameters are changed and the simulation is then re-run and the control models regenerated. For the thermoforming process, the following parameters are continuously monitored in real time.

- absorptivity
- heat capacity
- elastic modulus of the material.

Machine parameters that are monitored on-line are:

- input power of the heating elements
- surface temperature of the sheet in the oven
- forming pressure
- forming energy flow rate
- mould temperature.

Ambient parameters that are monitored on-line are:

- ambient temperature
- oven air temperature
- air velocity in the oven during sheet heating
- air velocity during the cooling phase after forming.

6 DIAGNOSTIC SYSTEM

A diagnosis module monitors the behaviour of the process during control. The agents that monitor the system act to detect any abnormal behaviour based on previously accumulated know-how about the process. They then either try to correct the anomaly (error recovery agent) or stop the machine in the case of a non-recoverable error. In all cases the operator is advised. Diagnostics and error recovery operate independently and asynchronously with respect to the process and control modules. More detail is given in Albadawi et al. (2006).

7 CONCLUSIONS

The model based control system proposed here for the thermoforming process:

- uses the energy required as the main control variable, allowing for easy energy minimization,
- can target a specific temperature at a specific depth and a specific time by adjusting a single state parameter,
- can predict the switch from steady to unsteady state in the process,
- can detect and adjust for a range of variations of material and machine parameters,
- has a response time that is adequate for in-cycle control, and
- inherently minimizes cycle time while respecting the process and material limitations.

Agent technology is an excellent match for control based on process models since it allows distributed intelligence and decision making to be applied to the control problem.

Subsystems for the control system have been developed and a system is being built to test control of an industrial thermoforming process.

REFERENCES

Ajersch, M., 2004. *Modeling and real-time control of sheet reheat phase in thermoforming*. Master's thesis, Department of Electrical Engineering, McGill University.

Albadawi, Z., Boulet, B., DiRaddo, R., Girard, P., Rail, A. and Thomson, V., 2006. Agent-based control of manufacturing processes. *Int. J. Manufacturing Research*, 1(4) 466–481.

Bahadoran, 2005. *Online characterization of viscoelastic and stress-relaxation behavior in thermoforming*. Master's thesis, Department of Mechanical Engineering, McGill University.

Florez-Mendez, R., 1999. Toward a standardization of multi-agent system frameworks. *ACM Crossroad*, 5(4), 18-24.

Gauthier, G., Ajersch, M., Boulet, B., Haurani, A., Girard, P., DiRaddo, R., 2005. A New Absorption Based Model for Sheet Reheat in Thermoforming, *SPE Annual Technical Conference, ANTEC*, Boston, Massachusetts.

Girard, P., Thomson, V., Boulet, B., DiRaddo, R.W., 2005. *Advanced in-cycle and cycle-to-cycle on-line adaptive control for thermoforming of large thermoplastic sheets*. *SAE World Congress*, Detroit, MI, USA.

Kumar, V., 2005. *Estimation of absorptivity and heat flux at the reheat phase of thermoforming process*. Master's thesis, Department of Mechanical Engineering, McGill University.

Mir, H., Benrabah, B., Thibault, F., 2007. The use of elasto-visco-plastic material model coupled with pressure-volume thermodynamic relationship to simulate the stretch blow molding of polyethylene terephthalate, *Numiform*, Porto, Portugal.

Weiss, G., 1999. *Multiagent systems*, MIT Press, Cambridge, USA.

Zhang, Y., 2004. *Generation of component libraries for the thermoforming process using on-line characterization*. Master's thesis, Department of Mechanical Engineering, McGill University.

LHTNDT: LEARN HTN METHOD PRECONDITIONS USING DECISION TREE

Fatemeh Nargesian and Gholamreza Ghassem-Sani
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
nargesian@ce.sharif.edu, sani@sharif.edu

Keywords: AI Planning, hierarchical task network planning, machine learning, decision tree, domain knowledge.

Abstract: In this paper, we describe LHTNDT, an algorithm that learns the preconditions of HTN methods by examining plan traces produced by another planner. LHTNDT extracts conditions for applying methods by using decision tree based algorithm. It considers the state of relevant domain objects in both current and goal states. Redundant training samples are removed using graph isomorphism. Our experiments, LHTNDT converged. It can learn most of preconditions correctly and quickly. 80% of our test problems were solved by preconditions extracted by ¾ of plan traces needed for full convergence.

1 INTRODUCTION

Hierarchical Task Network (HTN) planning is a promising and applicative research topic in Artificial Intelligence. The basic idea of HTN was first developed in mid-70s (Sacerdoti 1975; Tate 1977). Its formal underpinnings were developed in mid-90s (Erol, Hendler, and Nau 1996). An HTN planning problem consists of an initial state, a set of tasks to be performed as problem goal, and a domain description. HTN domain description contains a set of operators as primitive tasks (they can be performed directly in the domain during execution time) and a set of methods describing possible ways of decomposing tasks into subtasks and subtasks into primitive tasks. Each method has a precondition. In order to apply a method on a planning state, its precondition must be satisfied in that state. Planning is done by applying methods to decompose non-primitive tasks into subtasks, and applying operators to primitive tasks to produce actions. Planning ends when all of the tasks mentioned in the goal set are satisfied, then the planner has found a solution plan; otherwise the planner will need to backtrack and try other applicable methods and actions that are not considered yet (Ilghami, *et al.* 2005).

HTN is a configurable planner whose domain knowledge is provided by a human domain expert to achieve satisfactory performance. Therefore, such planners' functionality depends on domain-specific problem solving knowledge to be accurate. It should be born in mind that the designer of a domain for a configurable planner generally has many valid alternative ways of specifying the domain, and it is well known that the exact form of the domain can have a large impact on the efficiency of a given planner. Even if a human designer can identify some of the complex manner in which the tasks and operators in a domain description interact, he will likely be faced with tradeoffs between efficiency and factors such as compactness, comprehensibility and expressiveness. Consequently, there are obvious advantages to a planner that can evolve its domain theory via learning. Learning is the process of using past experiences and percepts to improve one's ability to act in the future. The extensive survey and analysis of research work related to machine learning as applied to planning reveals that machine learning methods can be used in learning and improving planning domain theory (Zimmerman and Kambhampati 2003). In this paper, we discuss a learning algorithm for evolving HTN planning domain automatically.

In recent years, several researches have reported work on integrating learning methods and HTN planning. An example is a system called HICAP, developed by Munoz-Avila *et al.* 1999, which uses planning in military environment. HICAP integrates SHOP, a hierarchical planner, (Nau, *et al.* 1999) and a case-based reasoning (CBR) system called NaCoDAE (Aha and Breslow 1997). Learning HTN domain means eliciting the hierarchical structure

relating tasks and subtasks. Existing work on learning hierarchies, extracts hierarchy from a collection of plans having primitive operators' descriptions (Ruby and Kibler 1991; Reddy and Tadepalli 1997; Choi and Langley 2005). The idea is that the tasks are the same as goals that have been achieved by the plans. Reddy and Tedepally, in X-Learn system, uses inductive generalization to learn task decomposition constructs, named D-rules, which relate goals, sub-goals and conditions of goal decomposition.

Research on learning macro-operators (Korf 1987; Mooney 1998; Botea, Muller, and Schaeffer 2005) falls in the category of learning hierarchical structures in planning domains. ALPINE (Knoblock 1993) and PARIS (Bergmann and Wilke 1995) systems are good examples of using abstraction in planning. Knoblock presented a completely automated approach for generating abstractions of problem solving using a tractable, domain-independent algorithm. The inputs of this system are the definition of a problem space and the problem to be solved; and the output is an abstraction hierarchy that is tailored to particular problem. Two recent studies (Ilghami *et al.* 2005; Xu and Munoz-Avila 2005) propose eager and lazy learning methods respectively to learn the preconditions of HTN methods. Ilghami, in CaMeL, assumes that method definitions are available and uses Candidate Elimination Algorithm to extract methods' preconditions from plan traces. In HDL (Ilghami, Nau, and Munoz-Avila 2006), there is no prior information about the methods and it learns HTN domain description by examining plan traces produced by another HTN problem-solver. Another recent work, by Langley and Choi 2005, learns a special case of HTNs, known as teleoreactive logic programs. Rather than a task list, this system uses a collection of Horn Clause-like concepts. The most recent work, by Hogg 2007, presents HTN-MAKER, an offline and incremental algorithm for learning task models. HTN-MAKER receives as input a collection of plans generated by a STRIPS planner (Fikes and Nilsson 1971), an action model, and a collection of task definitions; and produces a task model. When combined with the action model, this task model results in an HTN domain model.

Here, we introduce LHTNDT (Learn HTN using Decision Tree), an algorithm that uses a decision tree based learning method for learning preconditions of HTN methods. It is assumed that system has the knowledge of general structure of decomposing tasks into subtasks. But this knowledge is incomplete in case that it does not have sufficient information about where to use the method to be successful and efficient. LHTNDT learns conditions for efficient application of methods by doing analysis on plan traces that are known to be successful or unsuccessful for certain problem instances. The preconditions are shown in the formalism of a decision tree.

The paper is organized in 5 sections. Section 2 overviews the inputs to the learning algorithm and its output. Section 3 discusses the learning algorithm. Section 4 reports empirical results of applying learning method on a planning domain. Finally section 5 draws conclusions and describes future work.

2 INPUTS AND OUTPUTS OF LHTNDT

Inputs to the learning algorithm are annotated plans. We use a set of optimal plan traces, which contain not only the correct methods used for a planning problem, but also information about possible decisions that could be made while this plan was being generated. This form of input is often preferable because it will result in faster and more accurate learning, because plan traces contain much more information about the domain being learned than plans (Ilghami *et al.* 2005). The annotated plan traces are in the form of trees whose nodes are the states that the planner has passed through in order to achieve a goal. For each node we consider not only the methods that are on the path, to the goal state but also other applicable methods that have caused failure in planning or optimal planning.

The annotated plan traces are in the form of trees whose nodes are the states that the planner have passed through in order to achieve the goal.

The decision tree based learning algorithm is used for extracting complete domain knowledge for HTN. By using decision trees, planner is not obliged to test all method preconditions during test time. In the proposed algorithm, one decision tree is generated for each domain method. The target attribute of theses trees specify whether to use the corresponding method or not.

Thus, the output of LHTNDT is certain decision trees as preconditions of the domain methods. According to the structure of decision trees, the precondition learned for HTN by LHTNDT is a disjunctive normal form of sentences.

```
Inputs: S={S1, …, Sn}, a set of world states
        M={m1, …, mn}, the set of operators in HTN domain (D)
        Π={ Π1, …, Πn}, a set of plan traces
        (Ii, Gi, D), planning problem
        I: Initial State
        G: Goal State
Output: DT, a set of decision trees, each of them represents the precondition of
a method

LHTNDT(S, I, G, M, Π)
DT=0, Samplesi=0
FOR each plan trace Πi ∈ Π
     FOR each node n in Πi, whose si ∈ S except I and G
          FOR each method mi ∈ M considered in n
               Let samp be a training sample
               samp.CurrentState = n.State
               samp.GoalState = Πi .GoalState
               IF applying mi to n results in a non-leaf node
           Samp.Type = "Positive"
               ELSE
                    Samp.Type = "Negative"
               ExtractImportantObjects (samp, mi)
               IF (AddSample (Generalize (samp, mi), Samplesi, mi)))
              MakeDTTrainingSample (samp, Samplesi)
FOR each method mi in M
     LearnDT(Samplesi, mi, DT)
RETURN DT
```

Figure 1: The LHTNDT algorithm.

3 LHTNDT ALGORITHM

Pseudo-code of LHTNDT is given in Figure 1. For learning the preconditions of applicable methods of a state, we consider both the current and goal states of the plan. The attributes of the decision trees are the combination of domain predicates and all objects of the domain. These attributes are doubled when they are considered for both appearing in the goal and current states. The values that these attributes can take are *True, False* and *Don't Care*. The third value provides the system with the ability to check just the attributes that are important as preconditions of the method, during the test phase. LHTNDT starts by defining and initializing decision trees. Then for each node in each given plan trace, it produces a training sample for each applicable method. If applying the method on a node results in a non-goal leaf node, it reveals that using the method in that situation resulted in failure or a non-optimal plan. Thus, the sample extracted from it should be of type *Negative*. Otherwise, the produced sample is considered as *Positive* sample. The sample is added to training set if it does not have common structure with other existing samples. Finally, LHTNDT learns certain decision trees for domain methods. The algorithm converges when learned decision tress for methods converge. In other words, hypothesis spaces for preconditions converge to a fixed point. The subroutines of LHTNDT are as follows:

- **ExtractImportantObjects(*samp*, m_i)**, is a function that iteratively extracts the objects of the domain that are somehow related to method m_i. The main idea is inspired by Ilghami *et al.* 2005. The relative situation of domain objects in the current state and also their ultimate situation in the goal state specifies whether to apply the method or not. Considering the objects that are directly or indirectly related to the objects on which method is applied, assists the system not to learn the facts that are not useful. The function starts by initializing relevant objects to the set of objects that have appeared in a predicate of current state with at least one of the objects in the arguments of method m_i. Then, in each subsequent iteration, it alternatively processes goal state and current state and adds those remaining objects that are in a predicate in which one of the objects in the relevant objects

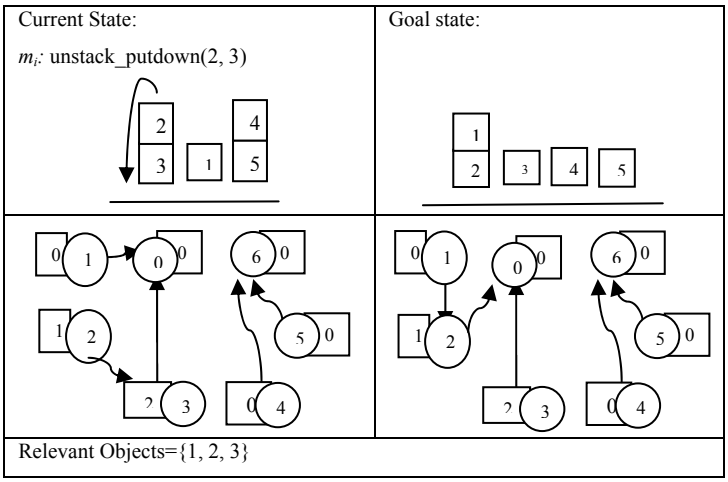


Figure 2: Graphs for a training sample in Blocks World domain (node 0 is considered as table, node 6 is defined for specifying non-relevant objects).

set appeared as an argument. These iterations continue until no new objects are added to the relevant objects set.

- **Generalize(*samp*, m_i)**, make the generalized form of the method and also the current and goal state mentioned in the training sample *samp*.
- **AddSample(*samp*, $Samples_i$, m_i)**, where $Samples_i$ is the set of all unique training samples for method m_i. This function creates a graph for each sample. The graph has two components, one for the initial state and the other for the goal state. The nodes of the graph are domain objects, named as various variables. Each node is labeled by a number indicating whether the corresponding object is a relevant object. The edges of the graph are labeled with the name of the predicate that two nodes are appeared in. The nodes that are arguments of m_i, are labeled with their argument order in the method. These graphs are also the generalized form of the samples. An example of the graph produced for a sample is depicted in figure 2. In order to prevent from adding a sample with the same structure as existing samples, we perform the test of graph isomorphism between the graph of *samp* and the graphs of other samples. If *samp* is isomorphic with none of the samples in $Samples_i$, *samp* is a new sample and should be added to the $Samples_i$ set. Thus, the function returns *True*, otherwise it returns *False*. The idea of using graphs for extracting new samples can also be used for the predicates of more than 2 arguments. The algorithm should be modified in such a way to group and merge nodes that are involved in a common method and adjoin node groups.
- **MakeDTTrainingSample(*samp*, $Samples_i$)**, assigns value to the attributes of decision tree in order to make a training sample. It maps sample graph nodes to the generalized objects in attributes by matching the node of the first argument of the method of *samp* with the first generalized object defined in attributes and continuing this matching by traversing the graph and mapping generalized object of attributes and newly-met objects of the graph. For each of the attributes, if none of the objects in the attribute is mapped with one of the objects in sample relevant object set, the value of the attribute is set to *Don't Care*. Otherwise, if the ground predicate made out of the attribute and its mapped objects is among initial state or goal state (according to the attribute name), the value of the attribute is *True* and in the case it is not, the value is *False*.
- **LearnDT($Samples_i$, m_i, *DT*)**, creates a decision tree for each of the methods according to training sample in $Samples_i$.

4 EMPIRICAL EVALUATION

One of the challenges in any learning algorithm is how to define a criterion to evaluate its output. In this section, we explain some points about implementing LHTNDT algorithm. Then, we discuss our experiments to figure out how many plan traces are needed to converge and also the precision

of the planning domain that LHTNDT has learned by just ¾ of plan traces needed for convergence.

As mentioned in previous section, LHTNDT extracts training samples from certain annotated optimal plan traces. Thus, LHTNDT, unlike previous systems which require plan traces produced by an expert or a planner working with a hand-tailored domain, does not need any priori knowledge. For accuracy of training samples, we needed all optimal solution plans of a problem. So that, we firstly used an iterative deepening Depth First Search (DFS) algorithm modified for planning. But this planner often failed in domains with many objects because of their time and space complexity. In order to fix this problem, we modified HSP_a*, which is an optimal temporal planner, a member of heuristic search planners. HSP* produces parallel optimal plans. Since optimal parallel plans are not certainly optimal sequential plans, optimal sequential plans are extracted from HSP_a* outputs and plan traces are created for them. We have assumed a method as a sequence of actions (a two level hierarchy of tasks). Therefore, the operators for planning are action sequences (macro-operators) that are defined as methods' structure. The domain used in our experiments is the blocks world, with 4 operators and 3 methods. It is based on the block-stacking algorithm discussed in (Gupta and Nau 1992). For training the system, we generated 4 sets of random problems with 3, 5, 6, and 10 blocks using BWSTATES program (Slaney and Thiébaux 2001). The uniqueness of the problems is checked by making a graph for each of the problems and testing its isomorphism relationship with other problem graphs. The nodes of the graphs are domain objects and the edges are labeled with predicate names and a number showing which state this predicate belongs to: the initial or goal states. Figure 3 compares the number of plan traces LHTNDT needed to converge.

Although more information is provided in plan traces of larger problems, the number of plan traces needed by LHTNDT to converge smoothly increases as more objects are defined in the domain. The reason is that the number of attributes and the situations that should be learned (current and initial states) increase. It should be also born in mind that a plan trace contains more negative samples than positive ones and this may cause later convergence.

An HTN planner is implemented for testing the preconditions learned by LHTNDT. This planner acts just like SHOP (Nau, *et al.* 1999), but in case that more than one method is recommended, by decision tree, to be applied, a random method is chosen. Often, there is not enough information in the input or there are not enough training samples to derive an optimal output. We used just ¾ of problems needed for convergence of LHTNDT for training and tested the learned preconditions on a random test set. In the diagram depicted in figure 4, the precision of method preconditions learned by ¾ problems needed for convergence is shown. Although the planner could not find a plan for all test problems, but it can be seen that LHTNDT can learn most of method preconditions even before full convergence, because it tries to extract as much information as it can form input plan traces. The precisions are high because we have taken into account the condition of objects both in the current state and goal state.

5 CONCLUSIONS AND FUTURE WORK

LHTNDT is an algorithm that learns method preconditions for HTN planner. Its input consists of annotated optimal plan traces produced by another planner (HSP*). LHTNDT takes into account the structure of current and goal state for learning conditions of applying a method. Diverse structure of training samples are extracted by checking the graph isomorphism among created graphs for samples. In our experiments in blocks world domain, although LHTNDT needed various plan traces to converge, according to the results for precision of planning domain before full convergence of preconditions, it learned most of necessary conditions of methods relatively quickly. Regarding the fact that no structurally repetitive training sample is considered for learning and also the relative state of relevant domain objects in both current and goal states are explored, it can be claimed that LHTNDT produces correct preconditions for HTN planning domain.

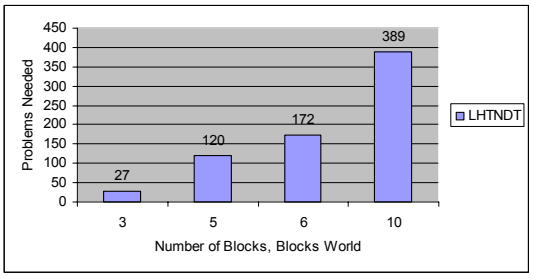


Figure 3: Number of plan traces needed to converge.

A decision tree represents the preconditions of domain methods as disjunctive normal form of sentences. We believe that some rules can be extracted from the decision trees. Therefore, axioms can be used in presenting the preconditions. If the existence of recursion is verified in a method precondition decision tree, the rules will be written in such a way that the rules that are extracted from smaller domains can be used for larger domains. LHTNDT algorithm learns preconditions of HTN methods. Our future work will include developing techniques to learn HTN methods' structures from plan traces produced by another planner.

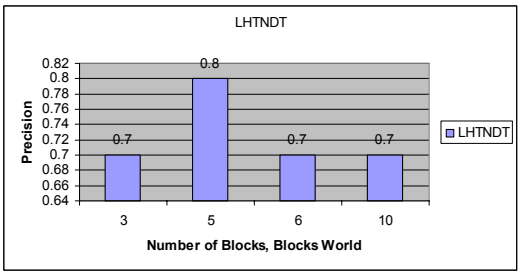


Figure 4: Precision of method preconditions learned by ¾ of plan traces needed for full convergence.

REFERENCES

Aha, D. W.; and Breslow, L. A. 1997. Refining Conversational Case Libraries. In *Proceedings of the Second International Conference on Case-Based Reasoning*, pp. 267-278. Providence, RI: Springer Press/Verlag Press.

Bergmann, R.; and Wilke, W. 1995. Building and Refining Abstract Planning Cases by Change of Representation Language. *Journal of Artificial Intelligence Research*, pp. 53-118.

Botea, A.; Muller, M.; and Schaeffer, J. 2005. Learning Partial-order Macros from Solutions. In *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS-05)*. AAAI Press.

Choi, D.; and Langley, P. 2005. Learning Teleoreactive Logic Programs from Problem Solving. In *Proceedings of the 15th International Conference on Inductive Logic Programming.* Springer.

Erol, K.; Hendler, J.; and Nau, D. S. 1996. Complexity Results for Hierarchical Task-Network Planning. *Annual of Mathematics and Artificial Intelligence* 18(1), pp. 69-93.

Fikes, R. E.; and Nilsson, N. J. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence (2)*, pp 189-208.

Gupta, N.; and Nau, D. 1992. On the complexity of Blocks-world Planning. *Artificial Intelligence* 56(2-3): pp. 223-254.

Hogg, Ch. 2007. From Task Definitions and Plan Traces to HTN Methods, *Workshop of International Conference on Automated Planning and Scheduling (ICAPS-07)*.

Ilghami, O.; Nau, D. S.; Munoz-Avila, H.; and Aha, D. 2005. Learning Preconditions for Planning from Plan Traces and HTN Structure.

Ilghami, O.; Munoz-Avila, H.; Nau, D. S.; and Aha. D. 2005. Learning Approximate Preconditions for Methods in Hierarchical Plans. In *Proceedingsof the 22nd International Conference on Machine Learning*, Bonn, Germany.

Ilghami, O.; Nau, D. S.; and Munoz-Avila, H. 2006. Learning to Do HTN Planning, *International Conference on Automated Planning and Scheduling (ICAPS-06)*.

Knoblock, C. 1993. Abstraction Hierarchies: An Automated Approach to Reducing Search in Planning. *Norwell, MA: Kluwer Academic Publishers.*

Korf, R. E. "Planning as Search: A Quantitative Approach", *Artificial Intelligence,* 33(1), pp. 65-88, 1987.

Mooney, R. J. 1998. Generalizing the Order of Operators in Macro-Operators. *Machine Learning,* pp. 270-283.

Munoz-Avila, H.; McFarlane, D.; Aha, D. W.; Ballas, J.; Breslow, L. A.; and Nau, D. S. 1999. Using Guidelines to Constrain Case-based HTN Planning. In *Proceedings of the Third International Conference on Case-Based Reasoning,* 288-302. Providence, RI: Springer Press.

Nau, D. S.; Cao, Y.; Lotem, A.; and Munoz-Avila, H. 1999. SHOP: Simple Hierarchical Ordered Planner. In *Proceedings of the Sixteenth International Joint conference on Artificial Intelligence,* 968-973. Stockholm: AAAI Press.

Reddy, C.; and Tadepalli, P. 1997. Learning Goal-decomposition Rules using Exercises. In *Proceedings of the International Conference on Machine Learning (ICML-97).*

Ruby, D.; and Kibler, D. F. 1991. Steppingstone: An Empirical and Analytic Evaluation. In *Proceedings of the 9th National Conference on artificial Intelligence:* pp. 527-531.

Sacerdoti, E. 1975. The Nonlinear Nature of Plans. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence,* Tiblisi, USSR, pp. 206-214.

Slaney J. K.; and Thiébaux S. 2001. Blocks World revisited. *Artificial Intelligence* 125(1-2): pp. 119-153.

Tate, A.; 1977. Generating Project Networks. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence,* Cambridge, MA, pp. 888-893. MIT Press.

Xu, K.; and Munoz-Avila, H. 2005. A Domain-independent System for Case-based Task Decomposition without Domain Theories. In *Proceedingsof the 20th National Conference on Artificial Intelligence (AAAI-05)*. AAA Press.

Zimmerman, T.; Kambhampati, S. 2003. Learning-assisted Automated Planning: Looking back, tracking Stock, Going Forward. *AI Magazine,* 73-96.

MERGING OF ADVICES FROM MULTIPLE ADVISORY SYSTEMS
With Evaluation on Rolling Mill Data

Pavel Ettler
COMPUREG Plzeň, s.r.o., Plzeň, Czech Republic
ettler@compureg.cz

Josef Andrýsek, Václav Šmídl, Miroslav Kárný
Department of Adaptive Systems, ÚTIA, AV ČR, Praha, Czech Republic
andrysek@utia.cas.cz,smidl@utia.cas.cz,school@utia.cas.cz

Keywords: Advisory system, Bayesian decision-making, Bayesian model averaging, multiple-participant decision-making.

Abstract: The problem of evaluation of advisory system quality is studied. Specifically, 18 advisory strategies for operators of a cold rolling mill were designed using different modelling assumptions. Since some assumptions may be more appropriate in different working regimes, we also design a new advising strategy based on the on-line merging of advices. In order to measure actual suitability of the advisory systems, we define two measures: operator's performance index and coincidence of the observed operator's actions with the advices. A time-variant model of advisory system suitability is proposed. Merging of the advices is achieved using Bayesian theory of decision-making. Final assessment of the original advisory systems and the new system is performed on data recorded during 6 months of operation of a real rolling mill. This task is complicated by the fact that the operator did not follow any of the recommendations generated by the advisory systems. Validation was thus performed with respect to the proposed measures. It was found that merging of the advising strategies can significantly improve quality of advising. The approach is general enough to be used in many similar problems.

1 INTRODUCTION

Theory and algorithms for design of advisory system based Bayesian decision-making theory have been consistently developed for years (Kárný et al., 2005). The theory was applied primarily to the probabilistic mixtures (Titterington et al., 1985) and the resulting algorithms were implemented in a Matlab toolbox Mixtools (Nedoma et al., 2005) which is also available as a platform-independent library. The first industrial application of the algorithms was designed for the twenty-high cold rolling mill in Kovohutě Rokycany within the international project ProDaC-Tool and was commissioned in autumn 2002 (Quinn et al., 2003), (Ettler et al., 2005a).

Development of the library continued, and within two years, its new version was ready for testing. The new library extended the number of possible settings of the advisory system. Great care was taken to eliminate the need for tuning knobs, and only discrete set of choices (such as class of models describing system behavior) was allowed. Thus, after three years of operation, the use of the original advisory system was suspended, it was replaced by a new experimental version and an experiment was undertaken to test suitability of these choices. To minimize the impact of the experiments on the production, mill operators were not asked to follow recommendations made by the system running in the experimental mode. Thus, we can not evaluate quality of the new systems from behavior of the closed loop, but we have to design evaluation criteria using only the open loop data.

Since only a small number of discrete choices is available, all possible combinations of these choices yield 18 different advisory systems, *advisers*. Each adviser is capable to generate recommendations for operators, *advices*. Thus, rather then simply choosing the best system, we also explore the possibility of merging advices from all the advisers. The merging rule is designed via a proposed model of evolution of advising quality. Relations of this approach to the Bayesian model averaging (Raftery et al., 1997) and

multiple participant decision-making (Kárný et al., 2007) is discussed.

2 COLD ROLLING MILL

A reversing cold rolling mill is essentially used to reduce the thickness of a strip of metal. This is done by passing the strip between rolls in alternating direction under high applied pressure and strip tensions. Several basic types of cold rolling mills are distinguished according to the arrangement of working and backup rolls. Data for experiments came from the twenty-high rolling mill mentioned in the Introduction. For this machine the strip thickness is measured by contact meters on both sides of the rolling mill, providing measurements of the input and output thickness and its deviation from the nominal value. A target thickness is defined, and this needs to be achieved with high accuracy depending on the actual nominal thickness and on the type of material. A typical required tolerance in the considered data set was $\pm$ 10μm (microns).

Strip thickness variation h_2 on the output side of the rolling mill is considered as the main system output and the only criterial variable for further considerations. The output is, under normal conditions, securely controlled by the AGC (Automatic Gauge Control) (Ettler and Jirkovský, 1991). The term "normal conditions" is worth a discussion: perfectly working hydraulic roll-positioning system, operating strip thickness measurement together with values of rolling force, strip tensions and speeds and other adjustments from the technologically correct ranges are prerequisites. Nevertheless, performance of the system may not be optimal in all regions within these ranges due to e.g. some hardly observable vibrations, unequal cooling and lubrication conditions, etc. Thus even if the AGC keeps the thickness deviation well in tolerance, its performance can be further improved by tuning of its working conditions. This is a task for an experienced operator. An advisory system was designed to support potentially inexperienced operators (Quinn et al., 2003; Ettler et al., 2005b). Evaluation of quality of advices and their potential improvement is considered next.

The operator directly adjusts variety of variables (*actions*), we consider just three of them: input and output strip tensions and output strip speed. Actual values of these actions form three-dimensional vector, u_t. The operator makes his decision according to his experience, using the provided digital measurements of key internal variables but also his senses (e.g. hearing an unusual noise). On the other hand, the advisory system must depend only on the measured quantities. In the considered experiment, the advisory system operated on ten variables including the three operators actions listed above. The full vector of observed data will be denoted by $d_t = [y_t', u_t']$.

Behavior of the operator and its improvement is difficult to quantify. We define a quantitative criterion, *operator's performance index*, on a batch of 1000 subsequent data records:

$$P = \frac{E(h_2^2)}{E((h_1 - \overline{h_1})^2)}, \tag{1}$$

Here, $E(h_2^2)$ is the expected value of square deviations of output thickness from the desired value, and $E((h_1 - \overline{h_1})^2)$ is the expected value of square deviations of input thickness from the mean value of the batch. The expected values are evaluated empirically with respect to all data in the batch. In effect, (1) measures the ratio of output quality to input quality. An experienced operator is able to adjust conditions for the AGC so that good output quality is achieved even if the input quality is low, yielding small values of P. On the other hand, high values of P indicate worse output quality than optimum when input quality was relatively good, which is a sign of suboptimal settings.

3 ADVISORY SYSTEM DESIGN

An advisory system is a special case of a control system, control actions of which are not implemented automatically, but only displayed to an operator who has the freedom to follow or ignore the advice. However, the advices should be designed in such a way that if followed, the system achieves the optimal performance. Thus, an advisory system can be designed using methodology developed for design of adaptive controllers (Kárný et al., 2005). This methodology is divided in two phases:

Off-line Phase: a family of parametric models of the system is chosen, and the best model within this family is identified using historical data. Bayesian approach to this step involves the tasks of prior elicitation, parameter estimation, model selection, and model validation. Then, desired behavior of the closed-loop system (i.e. the original system controlled by an ideal controller) is formalized in the form of a *target* model.

On-line Phase: the optimal advising strategies are designed such that the closed loop of the controlled system complemented by the advising strategy is the closest to the target behavior. The

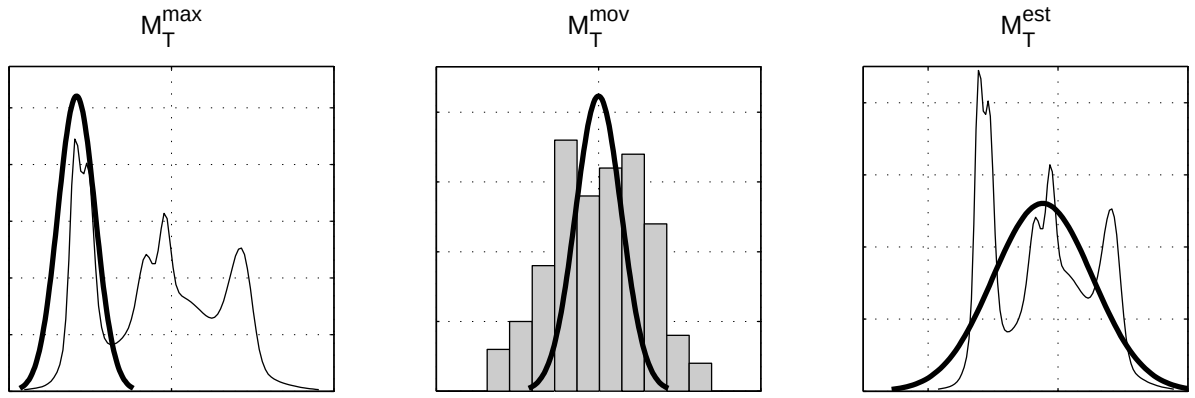


Figure 1: Graphic representation of construction of the target mixture M_T. Shown for a single dimension for the sake of clearness - M_T is depicted by the thick line, M_I or histogram by the thin line. Values of covariance for $M_T^{\max}$ and M_T^{mov} are chosen by the user while covariance of M_T^{est} is the result of estimation. Normalization is omitted here.

resulting optimized advising strategy is then presented to the operator in a simplified form. Typically, a small number of low dimensional projections of the probability distribution describing the advising strategy is automatically selected. Specific recommendations such as "Increase the output strip tension to 25 kN" are also provided.

The advising system implemented in Mixtools uses Gaussian mixtures as the main modeling family. Probabilistic mixture is a convex combination of probability densities on the same variables, which are called components. Gaussian mixture is a probabilistic mixture with Gaussian components, i.e.

$$M: \quad f(d_t) = \sum_{i=1}^{c} w_i f(d_t|\mu_i, \Sigma_i). \qquad (2)$$

Here, d denotes the modeled data, $f(d|\mu_i, \Sigma_i)$ is the ith Gaussian component with mean value μ_i and variance Σ_i, w_i is the weight of ith component, c denotes number of components. This choice of the model of the system is motivated by universal approximating properties of mixture models (Maz'ya and Schmidt, 2001). All introduced parameters, i.e. w_i, μ_i, Σ_i and c are considered to be unknown for all i.

Under this choice, an advisory system is designed as follows: (i) the unknown parameters are estimated in the off-line phase, yielding a mixture with estimated parameters, M_I, (ii) the target behavior of the closed loop is defined as M_T with specific choice of parameters in (2), and (iii) the part of M_I representing control strategy, is replaced by a parametric model, parameters of which are then optimized to minimize a statistical divergence to the target M_T; the result of this optimization is a new mixture M_A, for details see (Kárný et al., 2005). In each of these steps, it is possible to make several modeling choices, as described now in detail.

3.1 Variants of the System Model

A principal distinction in modeling of the system is the choice of static or dynamic model. The static approach models all observed data as independent realizations from the same density. The dynamic approach models also temporal dependence between subsequent data. This distinction is demonstrated in mixture models as follows:

M_I^{stat}: all observed data, d_t, $t = 1, \ldots, T$, are assumed to be generated from model (2) with time-invariant parameters μ_i, Σ_i, w_i.

M_I^{dyn}: observed data at time t, d_t, are assumed to be generated from model (2) with time-variant mean value, $\mu_i = \theta_i d_{t-1}$, and time-invariant parameters Σ_i, w_i.

3.2 Variants of the Target Mixture

Theoretically, the user can specify an arbitrary mixture model as his desired behavior. In practice however, he is concerned mostly with variables that are critical for the overall performance. If no suitable expert knowledge is available for the remaining non-critical variables, the target model on these must be chosen. Three variants of this choice of M_T were considered for experiments:

$M_T^{\max}$: Means of the non-critical variables in a single-component M_T are given by the maximum marginal probability of M_I in particular axes as depicted in Fig. 1, left. Thus, M_T remains unchanged during on-line operation of the system.

M_T^{mov}**:** Means of the non-critical variables vary in time according to the outputs of the moving-average filter processing actual data. Therefore M_T slightly differs for every step of the on-line operation as indicated in Fig. 1, middle.

M_T^{est}**:** A set of historical data from high-quality operating regimes were used to estimate parameters of a single-component mixture. This mixture was used as M_T afterward. During on-line computation M_T is not changed. An example of M_T^{est} is displayed in Fig. 1, right.

3.3 Variants of the Advising Strategy

Since the advising system is not hard-linked with the controlled system, its advices does not have to consider only the directly adjustable parameters. An advice does not have to be a numeric value, but it can suggest the operator to move into another operating mode of the machine, without being explicit about numerical values of adjustable parameters. Thus the freedom in design of an advisory system is with which parameters of the advising strategy are to be optimized. This decision influences the resulting advising strategy and thus the advisory mixture, M_A. Three variants of M_A were considered for experiments:

M_A^{acad}**:** optimization is done only with respect to weights w_i of the advising strategy. In effect, M_A^{acad} is composed of the same components as M_I (i.e all μ_i and Σ_i are the same), however, the component are weighted by different weights w_i.

M_A^{ind}**:** optimization is done with respect to the means μ_i. Weights w_i are assumed to be given by the process. Thus M_A^{ind} differs from M_I in component means and variances but not in their weights.

M_A^{simult}**:** is a combination of both previous approaches, i.e. both the weights and the component parameters are being optimized.

4 MERGING OF ADVICES

The Bayesian theory that was used for design of the advisory system ensures that all nuisance parameters were set (or integrated out). The discrete choices mentioned above are the only degrees of freedom considered in this experiment. All possible combinations of these yield $2 \times 3 \times 3 = 18$ different complete advisory systems, advisers. The task is to assess suitability of these advisers for the production of a real rolling mill. An ideal experiment would be to run a selected task 18 times under the same conditions, each time following different adviser. Since this is practically not feasible, we need to find an alternative evaluation method.

Moreover, each adviser may be more suitable for different operating conditions, and the best advising strategy is then to merge advices of all advisers together. In order to do that, we need to estimate the relation between advices of each adviser and operator's performance index, P_t. Let us consider a measure of coincidence of the current operator's actions with the recommendations of the ith adviser at time t:

$$C_{i,t} = E\left(1 - \frac{\max\left(|u_t - u^\star_{i,t}|, u_t\right)}{u_t}\right). \qquad (3)$$

Here, u_t denotes the observed actions, $u^\star_{i,t}$ recommended actions by the ith adviser, and $|\cdot|$ denotes absolute value. Thus, $C_{i,t} = 1$ when the operator follows recommendation the ith adviser exactly. Furthermore, lets assume that P_t is related to C_i via an unknown function, $P_t = g_i(C_i)$. In order to estimate local approximation of this function, we seek a parametric model of this relation. Since the advisers were designed to improve quality of control, we assume that $g_i()$ is a monotonic function. Application of the Taylor expansion at operating point $\overline{C}_{i,t}$ at time t yields

$$P_t = g_i(\overline{C}_{i,t}) + g'_i(\overline{C}_{i,t})(C_{i,t} - \overline{C}_{i,t}) + e_t, \qquad (4)$$

where $g'_i()$ denotes the first derivative of $g_i()$, $\overline{C}_{i,t}$ is the fixed point of expansion, and e_t is an aggregation of higher order term in the expansion. (4) motivates the following parametric model

$$P_t = b_{i,t} + a_{i,t}C_{i,t} + \sigma_{i,t}v_t, \qquad (5)$$

where $a_{i,t}$, $b_{i,t}$, and $\sigma_{i,t}$ are unknown time-variant parameters and v_t is a Gaussian distributed disturbance, $v_t \sim \mathcal{N}(0,1)$. (5) constitutes a linear regression, parameters of which can be estimated using recursive least squares. The lack of knowledge about evolution of parameters $a_{i,t}, b_{i,t}, \sigma_{i,t}$ in time motivates the use least squares with forgetting (Kulhavý and Zarrop, 1993), which is appropriate for slowly varying parameters.

We consider an advice to be optimal if it minimizes the operator's performance index in the next step:

$$u^{\text{mer}}_{t+1} = \arg\min_{u_t} E(P_{t+1}|u_{t+1}). \qquad (6)$$

The expected value is with respect to all unknown parameters

$$E(P_{t+1}|u_{t+1}) = \sum_{i=1}^{18} \alpha_{i,t} f(P_{t+1}|C_{i,t+1}(u_{t+1})), \qquad (7)$$

where $f(P_{t+1}|C_{i,t+1}(u_{t+1}))$ is obtained by integrating (5) over $b_{i,t}, a_{i,t}, \sigma_{i,t}$, and α_i denotes probability that the ith adviser is reliable,

$$\alpha_{i,t} = f(i_t = i|P_t, C_t) \propto f(P_t|C_{i,t}, i). \qquad (8)$$

Here, $\propto$ denotes equality up to a normalizing constant, and $f(P_t|C_{i,t}, i)$ is obtained by integrating (5) over $b_{i,t}, a_{i,t}, \sigma_{i,t}$.

We note the following:

- Evaluation of expectation (7) and its weights (8) is closely related to Bayesian model averaging (Raftery et al., 1997). The only difference of our approach is the recursive evaluation of time-variant weights $\alpha_{i,t}$.
- One possible interpretation of this approach is to consider each adviser as a decision-making unit (DMU) in multiple participant decision-making (Kárný et al., 2007). If the units are not aware of each other presence, they generate individual advices. If they are forced to cooperate in order to maximize common aim—i.e. maximum expected increase in performance (6)—the final advice is a result of negotiation defined via (7)–(8).

5 EXPERIMENTS

The approach was tested on a data set collected during 6 month of production of a cold rolling mill consisting of more than 4,2 million of 10 dimensional data records. The set contains data from a wide range of operating conditions such as different materials or different passes through the mill. The quality of final product was within the required range for great majority of the data, and so was the operator's performance index, see Fig 2. This implies that the AGC low-level controller worked very well, and thus the space for improvement that can be achieved via the use of an advisory system is rather small. Hence, evaluation of the designed advisers is challenging.

Both operator's performance index and coincidence was computed for each model for each of the 4227 data batches. These numbers form irregular clusters, discouraging parametric modeling of the relation. Hence, we propose to split all data records in two sets: (i) high-quality data, $P < \hat{P}$, and (ii) low-quality data, $P \geq \hat{P}$. Here, $\hat{P}$ denotes a chosen threshold of quality which can be chosen e.g. from histogram on Fig 2. For each of the data set, we evaluate median value of coincidence $C_{i,t}$ for all advisers as their representative statistics. These values are displayed for selected advisers in Fig. 3. Interpretation of these results is as follows: advices generated by a good advising system should coincide with the operators actions at high-quality data region, but should differ in low-quality data regions, pointing (hopefully) in the direction of improvement. The line in the middle indicates a region where an adviser's coincidence is of the same value for both high-quality and low-quality data. An ideal adviser should be in the right-bottom quadrant of the plot.

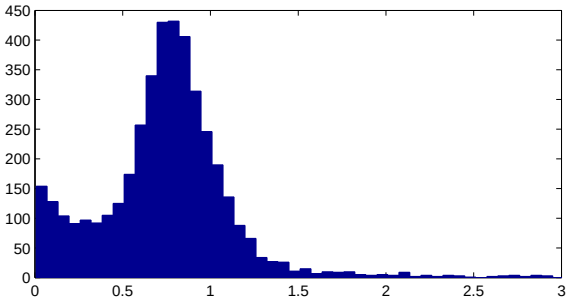


Figure 2: Histogram of operator's performance index for the considered data set. Only the range between 0 to 3 is displayed for clarity. Data records with $P > 3$ are infrequent but considerable.

Due to the lack of data in low-quality data set, the results are sensitive to the choice of the threshold $\hat{P}$, see results for $\hat{P} = 1.1$ and $\hat{P} = 1.4$ in Fig. 3. This sensitivity leads to a different choice of the best adviser from the original 18, specifically adviser $M_I^{\text{stat}}, M_T^{\text{mov}}, M_A^{\text{ind}}$, denoted by $\square$, and adviser $M_I^{\text{dyn}}, M_T^{\text{max}}, M_A^{\text{simult}}$ denoted by $\diamond$. Notably, however, the merged adviser, denoted by $\circ$, is performing well in both criteria.

This result should be taken only as qualitative for two reasons: (i) sensitivity of the criteria as described above, and (ii) for computational reasons, several approximations were used in evaluation of (6). Namely, integration over all parameters was replaced by conditioning on point estimates, and (7) was minimized only in the direction of its gradient in u_t. Nevertheless, the results indicate that merging of advices yields more robust adviser that any of the original ones.

6 CONCLUSIONS

A set of advising systems (advisers) was designed using different assumptions. A new adviser was constructed via on-line prediction of suitability of the original advisers for current working conditions and merging their recommendations. Performance of these advisers was assessed on real data. Evaluation of results was complicated by lack of data generated by an incorrectly set machine. Nevertheless, the re-

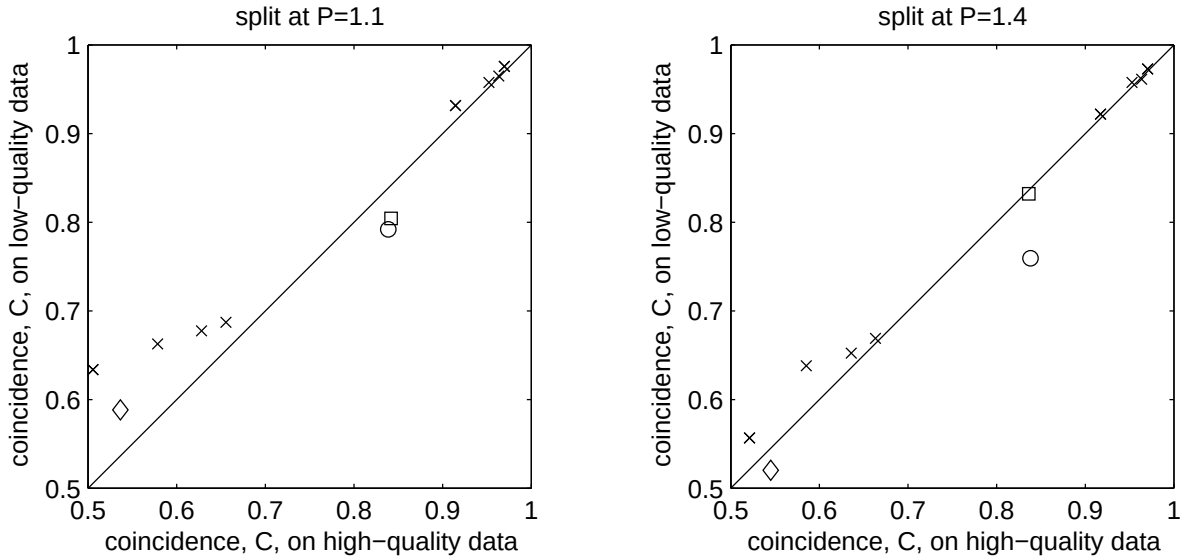


Figure 3: Comparison of coincidence C on high-quality and low-quality data for competing advisers. Two different thresholds between high and low quality were considered: $\hat{P} = 1.1$ (left) and $\hat{P} = 1.4$ (right). Original advisers with $C > 0.5$ are marked by $\times$, $\circ$ are denotes the merging adviser, $\square$ and $\diamond$ denote two remarkable original advisers, $M_I^{\text{stat}}, M_T^{\text{mov}}, M_A^{\text{ind}}$ and $M_I^{\text{dyn}}, M_T^{\text{max}}, M_A^{\text{simult}}$, respectively.

sults clearly show that merging of several advisers has the potential to provide better advices and it is more robust to the chosen evaluation metric.

ACKNOWLEDGEMENTS

Support of grants AV ČR 1ET 100 750 401 and MŠMT 1M0572 (DAR) is gratefully acknowledged.

REFERENCES

Ettler, P. and Jirkovský, F. (1991). Digital controllers for škoda rolling mills. In Warwick, K., Kárný, M., and Halousková, A., editors, *Lecture Notes: Advanced Methods in Adaptive Control for Industrial Application (Joint UK-CS seminar)*, volume 158, pages 31–35. Springer Verlag.

Ettler, P., Kárný, M., and Guy, T. V. (2005a). Bayes for rolling mills: From parameter estimation to decision support. In *16th IFAC World Congress*, Praha, CZ.

Ettler, P., Kárný, M., and Guy, T. V. (2005b). Bayes for rolling mills: From parameter estimation to decision support. In *Accepted for the 16th IFAC World Congress*, Praha, CZ.

Kárný, M., Böhm, J., Guy, T., Jirsa, L., Nagy, I., Nedoma, P., and Tesař, L. (2005). *Optimized Bayesian Dynamic Advising: Theory and Algorithms*. Springer, London. to appear.

Kárný, M., Kracík, J., and Guy, T. (2007). Cooperative decision making without facilitator. In Andrievsky B.R., F. A., editor, *IFAC Workshop "Adaptation and Learning in Control and Signal Processing" /9./*. IFAC.

Kulhavý, R. and Zarrop, M. B. (1993). On a general concept of forgetting. *International Journal of Control*, 58(4):905–924.

Maz'ya, V. and Schmidt, G. (2001). On approximate approximations using Gaussian kernels. *IMA Journal of Numerical Analysis*, 16(1):13–29.

Quinn, A., Ettler, P., Jirsa, L., Nagy, I., and Nedoma, P. (2003). Probabilistic advisory systems for data-intensive applications. *International Journal of Adaptive Control and Signal Processing*, 17(2):133–148.

Raftery, A., Madigan, D., and J.A.Hoeting (1997). Bayesian model averaging for linear regression models. *Journal of Thhe American Statistical Association*, 97(437):179–191.

Titterington, D., Smith, A., and Makov, U. (1985). *Statistical Analysis of Finite Mixtures*. John Wiley, New York.

Nedoma, P., Kárný, M., Böhm, J., and Guy, T. V. (2005). Mixtools Interactive User's Guide. Technical Report 2143, ÚTIA AV ČR, Praha.

SHORT PAPERS

SELF-ORGANISATION OF GAIT PATTERN TRANSITION
An Efficient Approach to Implementing Animal Gaits and Gait Transitions

Zhijun Yang, Juan Huo and Alan Murray
Institute of Micro and Nano Systems, School of Engineering and Electronics
Edinburgh University, Edinburgh EH16 6XD, U.K.
{*Zhijun.Yang, J.Huo, Alan.Murray*}*@ed.ac.uk*

Keywords: Central pattern generator, oscillatory building block, gait transitions, Self-organisation, Hopfield network.

Abstract: As an engine of almost all life phenomena, the motor information generated by the central nervous system (CNS) plays a critical role in the activities of all animals. Despite the difficulty of being physically identified, the central pattern generator (CPG), which is a concrete branch of studies on the CNS, is widely recognised to be responsible for generating rhythmic patterns. This paper presents a novel, macroscopic and model-independent approach to the retrieval of different patterns of coupled neural oscillations observed in biological CPGs during the control of legged locomotion. Based on the simple graph dynamics, various types of oscillatory building blocks (OBB) can be reconfigured for the production of complicated rhythmic patterns. Our quadrupedal locomotion experiments show that an OBB-based artificial CPG model alone can integrate all gait patterns and undergo self-organised gait transition between different patterns.

1 INTRODUCTION

Animal gait analysis is an ancient science. As early as two thousand years ago, Aristotle described the walk of a horse in his treatise (Aristotle, 1936). In modern biological research, it is widely believed that animal locomotion is generated and controlled, in part by central pattern generators (CPG), which are networks of neurons in the central nervous system (CNS) capable of producing the rhythmic outputs (Stein, 1978),(Grillner, 1985),(Pearson, 1993). The constituents of the locomotory motor system are traditionally modelled by nonlinear coupled oscillators, representing the activation of flexor and extensor muscles by, respectively, two neurophysiologically simplified motor neurons (Linkens et al., 1976),(Tsutsumi and Matsumoto, 1984),(Bay and Hemami, 1987). Despite its mathematical accuracy and ability to mimic some basic oscillatory features, this approach provides, however, neither a sufficiently detailed description of the real biological mechanisms nor a model simple enough for application purpose. Based on the graph dynamics, in this paper we present a structural approach to the modelling of the complex behavioural dynamics with a new concept of oscillatory building blocks (OBB) (Yang and Franca, 2003), (Yang and Franca, 2008). For the first time we present that the OBB model is able to self-organise its different gait pattern outputs under the control of a selecting signal flow in the neuro-musculo-skeletal system. Through appropriate selection and organisation of suitably configured OBB modules, different gait patterns and transitions between different patterns can be achieved for producing complicated rhythmic outputs, retrieving realistic locomotion prototypes and facilitating the very large scale integrated (VLSI) circuit synthesis in an efficient, uniform and systematic framework.

2 METHOD

Out-of-phase (walking and running) and in-phase (hopping) are the major characteristics of observed gaits in bipeds, while in quadrupeds more gait types were observed and enumerated (Alexander, 1984), as walk, trot, pace, canter, gallop, bound and pronk. Unlike bipeds and quadrupeds, hexapod locomotion can have more complicated combinations of leg movements. Despite the variety, however, some general symmetry rules should still be obeyed and remained as the basic criteria for gait prediction and construction. For instance, it is a generally accepted view that multi-legged (usually more than six legs) locomotion often display a travelling wave sweeping along the chain of oscillators (Collins and Stewart, 1993),(Gol-

ubitsky et al., 1998). In order to construct a novel, uniform OBB-based CPG architecture to integrate all gaits as well as the self-organised gait transitions of a specific legged animal, we need a case study of the simple graph dynamics as a start.

2.1 A Case of Graph Dynamics

Consider a system consisting of a set of processes and a set of atomic shared resources represented by a connected graph $G = (N, E)$, where N is the set of processes and E the set of edges defining the interconnection topology. An edge exists between any two nodes if and only if two corresponding processes share at least one atomic resource (Fig. 1).

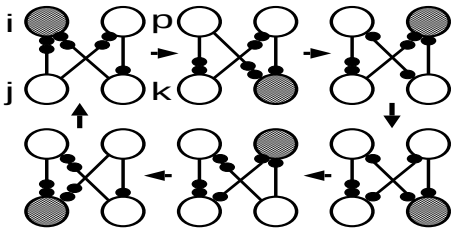


Figure 1: A simple graph dynamics representing an operation cycle, $r_i = r_j = 2$, $r_k = r_p = 1$, dark-filled circle represents a node is operating.

Between any two nodes i and j, $i, j \in N$, there can exist e_{ij} unidirected edges, $e_{ij} \geq 0$. The reversability of node i is r_i, i.e., the number of edges that shall be reversed by i to each of its neighbouring nodes, indiscriminately, at the end of the operation. Node i is an r_i-sink if it has at least r_i edges directed to itself from each of its neighbours. Only r_i-sink node can operate and reverse r_i edges towards each of its neighbours, the new set of r_i-sinks will operate, and so on. In order to avoid operation deadlock, the shared resources between any two nodes, e_{ij}, must satisfy $e_{ij} = r_i + r_j - gcd(r_i, r_j)$, and $max\{r_i, r_j\} \leq e_{ij} \leq r_i + r_j - 1$ (Barbosa, 1996), where *gcd* for the greatest common divisor. This simple graph dynamics can be used to construct the artificial CPGs by implementing OBB modules as asymmetric Hopfield-like networks, where operating sinks can be regarded as firing neurons in purely inhibitory neuronal networks.

2.2 Dynamics of an OBB Module

An OBB module is defined to have a pair of r_i-sink and r_j-sink nodes, n_i and n_j, sharing the number of e_{ij} resources. The postsynaptic membrane potential of neuron i at t instant, $M_i(t)$, depends on three factors, i.e., the potential at the last instant $M_i(t-1)$, the impact of its coupled neuron output $v_j(t-1)$, and the negative feedback of neuron i itself $v_i(t-1)$, without considering the external impulse. The selection of system parameters, such as the neuron thresholds and synapse weight, are crucial for modelling. In our model, let $r = max(r_i, r_j)$ and $r' = h(r)$, where h is a function of highest integer level and multiplying it by 10, e.g., if $r_i = 81$ and $r_j = 341$ then $r' = h(r) = h(max(81, 341)) = h(341) = 10^3$. We can further design neuron i and j's thresholds θ_i, θ_j and their synapse weights w_{ij}, w_{ji} as follows,

$$\begin{cases} \theta_i = max(r_i, r_j)/(r_i + r_j - \gcd(r_i, r_j)) \\ w_{ij} = max(r_i, r_j)/r' \\ \theta_j = (min(r_i, r_j) - 1)/(r_i + r_j - \gcd(r_i, r_j)) \\ w_{ji} = min(r_i, r_j)/r' \end{cases} \tag{1}$$

The difference equation in the discrete time domain of this system can be formulated as follows: each neuron's self-feedback strength is $w_{ii} = -w_{ij}$, $w_{jj} = -w_{ji}$. The activation function is a sigmoidal Heaviside type. It is worth noticing that k is a local clock pulse of each neuron, a global clock is not required. Thus we have,

$$\begin{cases} M_i(t+1) = M_i(t) + w_{ji} v_j(t) + w_{ii} v_i(t) \\ M_j(t+1) = M_j(t) + w_{ij} v_i(t) + w_{jj} v_j(t) \end{cases} \tag{2}$$

where,

$$\begin{cases} v_i(t) = max(0, sgn(M_i(t) - \theta_i)) \\ v_j(t) = max(0, sgn(M_j(t) - \theta_j)) \end{cases} \tag{3}$$

We consider the designed circuit as a conservative dynamical system in an ideal case. The total energy is constant, no energy loss or complement is allowed. The sum of two neurons' postsynaptic potential at any given time instant is normalised to one. It is clear that this system has the capability of self-organised oscillation with the firing rate of each neuron arbitrarily adjustable.

If a neuron has more than one connections, then its firing state depends on the interactions of all the connections,

$$V_i(t) = \prod_{j=1}^{n} v_i^j(t) \tag{4}$$

For instance, neuron i in Fig. 1 has connections with both neuron j and k, the output of i is expressed as $V_i(t) = v_i^j(t) \times v_i^k(t)$, here $v_i^j(t)$ and $v_i^k(t)$ are obtained from the above dynamical equations.

3 QUADRUPEDAL GAIT MODEL

Generally seven types of primary gait patterns are identified for a quadruped in literature (Alexander,

1984). These can be summarised in Table 1.

Table 1: Quadrupedal primary gaits and description.

Gaits	Pattern description	
	Ipsilateral legs	Contralateral legs
walk	quarter cycle out	half cycle out
trot	half cycle out	half cycle out
pace	in phase	half cycle out
gallop	half cycle out	quarter cycle out
bound	half cycle out	in phase
pronk	in phase	in phase
jump	quarter cycle out	in phase

The proposed CPG model, with adaptable, different model parameters, is able to generate all the gait patterns shown in Table 1. The signals selecting gaits and controlling transitions in the CPG, usually in the spinal cord, come from the higher neural level in the prefrontal, premotor and motor cortices following the dog's interaction with its environment.

Figure 2: Trot gait of a dog.

We choose a dog's risk-avoiding behaviour to build a case study of an CPG model which is made of the OBB modules. Suppose a dog is initially wandering around in a walk gait. Suddenly it is frightened by something and makes an abrupt left turn and escapes with a faster trot gait (Fig 2). In this scene the dog's trajectory includes three different stages in sequence: walk, left turn (a special gait) and trot. These locomotion behaviours are controlled by the different motor driving outputs from its CPG. The activity of a dog's leg is simplified and generated by the activity of a flexor and an extensor motor neuron, respectively. The firing of a neuron in the OBB drives the flexor which lifts the leg from the ground in a swing stage. The idleness of a neuron activates the extensor which lands the leg on the ground in a stance stage.

The OBB network topologies for the rhythmic pattern generation, and the phase relationship for aforementioned three gait patterns are shown in Fig 3, where the firing cell (gray colour) denotes its corresponding leg is lifting from the ground. In the right, black-white bar, black bars specify the firing of flexors while white ones for the firing of extensors. It is clear that the special, left turn gait has a gait pattern circulation as shown in Fig. 1.

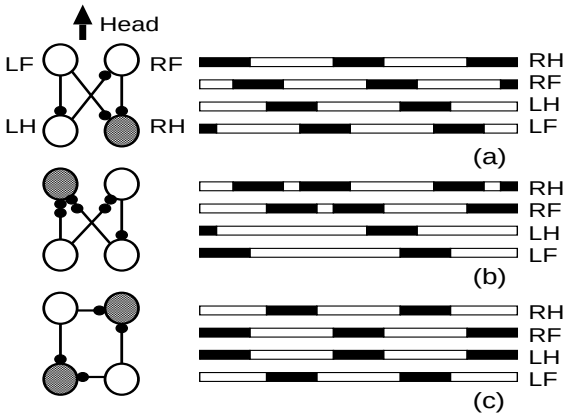


Figure 3: The schematic diagram of three gaits and their phase relations. The little black circles denote the shared resources, gray cells are firing and white ones are idle. The left side topology is only one configuration of a circulation period of a pattern. RF - right front, RH - right hind, LF - left front, LH - left hind. (a) walk, (b) left turn, (c) trot. .

From the biological knowledge we know that locomotion speed is decided by both the coordinated phase relation among legs and the duty factor, which is the proportion of an extensor's firing duration in one period. When an animal's locomotion speed increases, the extensor firing time (corresponding to stance) will decrease drastically while the flexor firing time (to swing) keeps basically constant (Pearson, 1976). This results in a lower duty factor, a relatively longer swing stage and a faster locomotion speed. As the same with adjusting coordinated phase relationship, the duty factor can also be modified by changing the reversabilities of two coupled cells.

4 SIMULATION RESULTS

A schematic circuit diagram of OBB-based asymmetric Hopfield-like neural network is shown in Fig. 4. A computer simulated example of quadruped risk-avoiding behaviour is conducted using this network and the pre-defined cell reversabilities. There are totally six possible connections between any two cells in this CPG architecture (as shown in Fig. 3), among which four connections are selected for a given gait pattern. The selecting signals, labelled from a_1 to a_6, control the selection of an individual connection, respectively, and come from the higher cortex. The change of these signals denotes a change of connection topology. This change, together with the change of reversabilities of the cells, decides the transition of gait patterns.

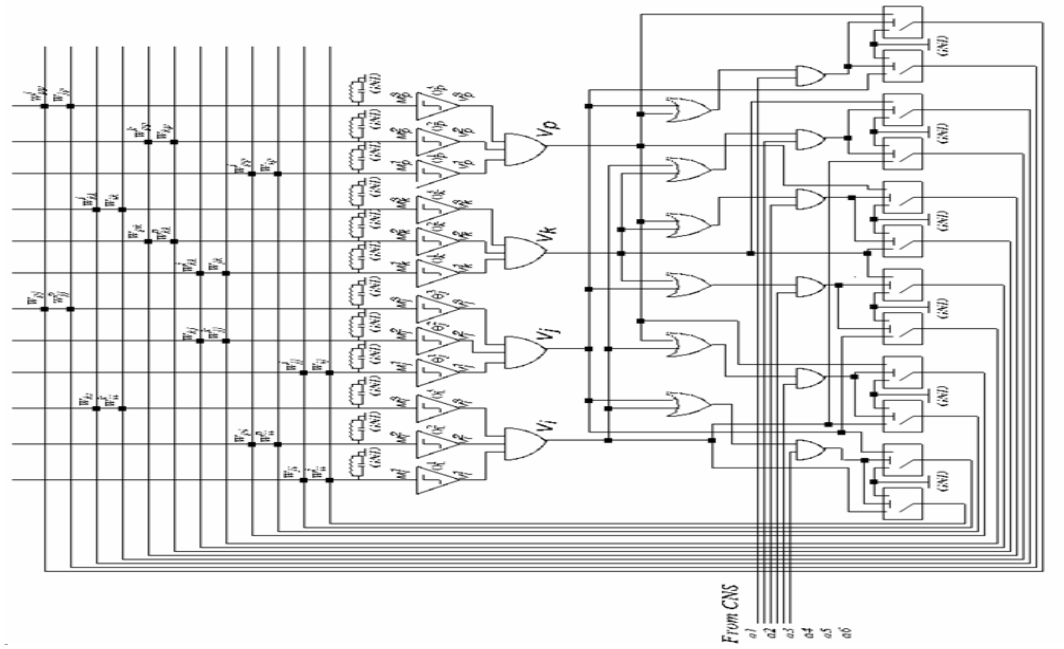


Figure 4: Quadruped CPG architecture. The cell output V_i, V_j, V_k, V_p correspond to LF, LH, RH, RF respectively. The predefined cell reversabilities are $r_i = r_j = r_k = r_p = 1$ for the walk and trot gaits, $r_i = r_j = 2$ and $r_k = r_p = 1$ for the left turn gait. The initial membrane potentials are $M_i^j(0) = M_i^k(0) = M_j^p(0) = M_p^k(0) = M_k^j(0) = 0.45$, $M_j^i(0) = M_k^i(0) = M_k^p(0) = M_p^j(0) = M_j^k(0) = 1.05$, $M_i^p(0) = 0.55$, $M_p^i(0) = 0.95$. The gait select signals are $[a1, a2, a3, a4, a5, a6] = [111001]$ for walk and trot, $[001111]$ for left turn. The cell thresholds and weights follow equation 1.

An array of twelve comparators, three for a cell, is used to compare the sub-cells' membrane potentials with thresholds. If a connection is absent then the relative comparator is disabled. The enabled connections of a cell converge to an AND gate, to implement equation 4 for the cell output. If a cell has an output, this output will make three OR gates, corresponding to its connections with all rest cells, have an output. These OR gate outputs are selected by the control signals from CNS at an array of AND gates, whose values are fed to an array of swithes, and form a loop to the input weight matrix for the comparator array.

According to equation 4, The output of a cell is the multiplication of two terminal outputs connecting with the cell's two neighbours. A cell will fire if and only if its sub-cells are firing simultaneously. An additional control (not shown) is needed to ensure that all shared resources on an edge will not reverse unless the output of a cell is firing. The system parameters are given in the legend of Fig. 4. The choice of initial membrane potential for every sub-cells is to make all cells work, i.e., being firing or idle, in an appropriate configuration of a starting gait's circulation period. After the choice of the membrane potentials the system will run adaptively in its gait transition which is controlled by the CNS signals. There is a self-organisation period observable in Fig. 5 from walk to left turn gait. Another self-organisation between left turn to trot should exist whose effect may be minimised if the system parameters are occasionally appropriate for the next locomotion pattern at the transition time instant.

5 CONCLUSIONS

We have presented a novel central pattern generator model capable of generating a whole range of legged locomotion gait patterns. This CPG architecture is generalisable to mimic any gait patterns of any legged animals from biped to centipede. Self-oganisation in gait transition is observed in control of a limited number of signal flow bits from the high level cortex. This modular system is simple and highly compatible to circuit implementation.

ACKNOWLEDGEMENTS

This work is supported by a British EPSRC grant EP/E063322/1, a Chinese NSF grant 60673102 and

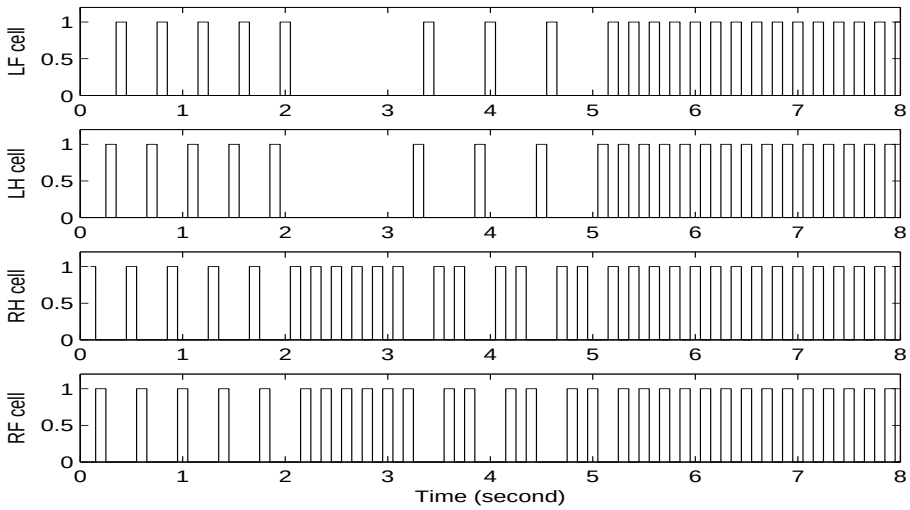


Figure 5: Cell output. Cell activity is equivalent to flexor activity and contrary to extensor activity in phase. The virtual dog starts with walk for 2 seconds. After that the left turn gait lasts for 3 seconds, which include a one-second self-organisation period. From the fifth second on it uses the faster trot gait to avoid the risk.

a Jiangsu Province grant BK2006218. We thank the useful discussions with Dr. Felipe M.G. Franca and Mr. Rodrigo Rodovalho.

REFERENCES

Alexander, R. (1984). The gait of bipedal and quadrupedal animals. *Intl J. Robotics Research*, 3:49–59.

Aristotle (Translated version of 1936). *On Parts of Animals, Movement of Animals, Progression of Animals. By A.S. Peek and E.S. Forster (translators)*. Cambridge, MA: Harvard University Press.

Barbosa, V. (1996). *An introduction to distributed algorithms*. Cambridge, MA: The MIT Press.

Bay, J. and Hemami, H. (1987). Modeling of a neural pattern generator with coupled nonlinear oscillators. *IEEE Trans. Biomedical Engr.*, 34:297–306.

Collins, J. and Stewart, I. (1993). Coupled nonlinear oscillators and the symmetries of animal gaits. *Journal of Nonlinear Science*, 3:349–392.

Golubitsky, M., Stewart, I., Buono, P., and Collins, J. (1998). A modular network for legged locomotion. *Physica D*, 115:56–72.

Grillner, S. (1985). Neurobiological bases of rhythmic motor acts in vertebrates. *Science*, 228:143–149.

Linkens, D., Taylor, Y., and Duthie, H. (1976). Mathematical modeling of the colorectal myoelectrical activity in humans. *IEEE Transactions on Biomedical Engineering*, 23:101–110.

Pearson, K. (1976). The control of walking. *Scientific American*, 235:72–86.

Pearson, K. (1993). Common principles of motor control in vertebrates and invertebrates. *Annual Review of Neuroscience*, 16:265–297.

Stein, P. (1978). Motor systems with specific reference to the control of locomotion. *Annual Review of Neuroscience*, 1:61–81.

Tsutsumi, K. and Matsumoto, H. (1984). A synaptic modification algorithm in consideration of the generation of rhythmic oscillation in a ring neural network. *Biological Cybernetics*, 50:419–430.

Yang, Z. and Franca, F. (2003). A generalized locomotion cpg architecture based on oscillatory building blocks. *Biological Cybernetics*, 89:34–42.

Yang, Z. and Franca, F. (2008). *A general rhythmic pattern generation architecture for legged locomotion, In Advancing artificial intelligence through biological process applications (A. Porto, A. Pazos, W. Buno, editors)*. Hershey, PA: Idea Group Inc.

PARALLEL MACHINE EARLINESS-TARDINESS SCHEDULING
Comparison of Two Metaheuristic Approaches

Marcin Bazyluk, Leszek Koszalka
Chair of Computer Systems and Networks, Wroclaw University of Technology, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
marcin.bazyluk@pwr.wroc.pl, leszek.koszalka@pwr.wroc.pl

Keith J. Burnham
Control Theory and Applications Centre, Coventry University, Coventry, U.K.
k.burnham@coventry.ac.uk

Keywords: Task scheduling, parallel machines, heuristics, genetic algorithms, tabu search.

Abstract: This paper considers the problem of parallel machine scheduling with the earliness and tardiness penalties (PMSP_E/T) in which a set of sequence-independent jobs is to be scheduled on a set of given machines to minimize a sum of the weighted earliness and tardiness values. The weights and due dates of the jobs are distinct positive numbers. The machines are diverse - each has a different execution speed of the respective jobs, thus the problem becomes more complex. To handle this, it two heuristics are employed, namely: the genetic algorithm with the MCUOX crossover operator and the tabu search. The performances of the both approaches are evaluated and their dependency on the shape of the investigated instances examined. The results indicate the significant predominance of the genetic approach for the larger-sized instances.

1 INTRODUCTION

The problem of job scheduling on parallel machines (PMJS) has been considered recently in many scientific and industrial communities. In the classical example of this type of problem the collection of independent jobs is investigated. Each of the tasks is to be assigned to the single set of the available machines which may be identical or different. The problem considered here extends the PMJS with the weighted earliness and tardiness penalties calculated for the execution of a job before or after its due date. A single job consists of the following parameters, namely its weight, also defined as the importance, and its due date. Baker and Scudder (Baker and Scudder, 1990) provide a survey on the job scheduling problem involving a due date determination. In this paper, the due date for the each job is considered as a specific moment in the time. If a given job is accomplished at its due date, the requirements are fulfilled and the penalty is not subjected. The specification of a single machine is implemented as a vector describing its processing speed of the individual jobs. It is to be noted, that the identical machines are characterized by the identical vectors.

In the case, when the due dates of all of the jobs are set to 0 (thus all the jobs require the fastest possible execution with no earliness penalty) and their weights are equal, the problem considered is NP-hard (Du and Leong, 1990). It is to be remarked, that when the weights are arbitrary positive numbers, the task becomes NP-hard in a strong sense. On the other hand, the problem considered here may be thought of as even more complex. Kanet examined the problem of minimizing the total weighted earliness with the tardiness on a single machine and common due dates in (Kanet, 1981). He proposed the algorithm characterized by a polynomial complexity. His work was extended by Hall and Posner to the case of several identical machines in (Hall and Posner, 1991). This paper forms a further extension with the due dates allowed to be different.

The example solution to the problem considered, is a schedule of all jobs that are assigned to the separate machines in a set of queues - one for each machine. The premise is to minimize the sum of the penalties for all the jobs. The penalty for a single job is calculated according to the Equation (1) where p_i and w_i are the penalty and the weight of the job i, respectively. $e_i = max\{0, d_i - c_i\}$ denoted the earliness of the job i and $t_i = max\{0, c_i - d_i\}$ is the tardiness of the job i with d_i being its corresponding due date

and c_i its completion time (the time instant when the processing of a job has been finished). The following formula is introduced, see (2), accounting for the total earliness and tardiness penalty (p), where I denotes the set of jobs.

$$p_i = w_i e_i + w_i t_i \tag{1}$$

$$p = \sum_{i \in I} w_i e_i + \sum_{i \in I} w_i t_i \tag{2}$$

2 PROBLEM FORMULATION

Let the problem of the scheduling of the N independent lobs on a set of M available parallel machines be considered. Each job may be completed by any of the available machines, however the time required differs, in general, depending on the machine chosen. If the job has been accomplished too early or too late, the penalty is calculated for the earliness or tardiness, respectively, according to the difference between the due date of the job and the real execution time. It is noted, that the penalty is also proportional to the weight of the particular job. By considering the possible decisions during the job scheduling procedure, two optimization techniques have been developed to minimize the total penalty for all the jobs. The mathematical model presented here is a modification of that proposed by Cao, Chen and Wan in (Cao et al., 2005). The next two subsections introduce the nomenclature and the definitions required, to formulate the problem.

2.1 Known Parameters

i, j	$= 1,2,\ldots,N$, job indeces
m	$= 1,2,\ldots,M$, machine index
z_{im}	processing time of the i-th job on the m-th machine, $z_{im} = \{1,2,\ldots,10\}$
w_i	weight of the i-th job , $w_i = \{1,2,\ldots,10\}$
d_i	due date of the i-th job , $d_i = \{1,2\ldots,5\}$
c_i	execution time of the i-th job
e_i	earliness of jo i, $e_i = max\{0, d_i - c_i\}$
t_i	tardiness of jo i, $t_i = max\{0, c_i - d_i\}$

2.2 Decision Variables

$$x_{ijm} = \begin{cases} 1 & \text{if job } j \text{ follows job } i \text{ on machine } m, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{im} = \begin{cases} 1 & \text{if job } i \text{ is executed by machine } m, \\ 0 & \text{otherwise,} \end{cases}$$

$$i = 0,1,\ldots,N,\ j = 1,\ldots,N,$$
$$j \neq i, m = 1,\ldots,M$$

With the decision variables defined, the following recursive formula (Equation (3)) is proposed for the calculation of the moment of the job completion. The job completion is defined as the sum of its processing time together with the moment when the job that is executed directly before the one being considered here, is completed.

$$c_j = \sum_{i=0}^{N} \sum_{m=0}^{M} x_{ijm} c_i + z_{jm} \tag{3}$$

2.3 Assumptions

$$\text{Min } T_c = \sum_{i=1}^{N} w_i (e_i + t_i) \tag{4}$$

$$\sum_{i=1, i \neq j}^{N} \sum_{m=1}^{M} x_{ijm} = 1 \tag{5}$$

$$\sum_{i=1, i \neq j}^{N} x_{ijm} = y_{jm}, \tag{6}$$

$$\sum_{j=1, j \neq i}^{N} x_{ijm} \leq y_{im} \tag{7}$$

$$c_j + A(1 - x_{ijm}) \geq c_i + z_{jm} \tag{8}$$

In all the assumptions introduced, the parameters are defined as follows:

$$i = 1,2,\ldots,N,\ j = 1,2,\ldots,N,\ i \neq j,$$
$$m = 1,2,\ldots,M,\ c_i > 0,\ c_j > 0$$

In the model presented, Equation (4) describes the objective function as the sum of the penalties for all the jobs which are to be minimized. The following equations defines the restrictions to which the decision variables and the parameters are subjected. Equation (5) ensures that, first, every job is processed on a single machine and, second, that its execution is not divided into separate parts. Equation (6) imposes a restriction that the j-th job must immediately follow any other job on the m-th machine (or be left in the first position in the case when i = 0). Thereby, the intervals in the machine usage are not allowed. Equation (7) states that, if the i-th job is processed on the m-th machine, it will be immediately followed by at most one other job on this machine. Therefore, the jobs can be executed one at the each time instant. Equation (8) expresses the finite completion time of each job, where the scalar A is a large positive number.

Due to the evident complexity of the problem considered, the investigations are restricted to the class of heuristic algorithms. In Sections 3–4 two heuristic based approaches are presented.

START

- Find the first solution.

LOOP

- Find the best solution which is not tabu.
- Update the global best solution.
- Add a rule to the tabu list.
- If the fixed number of the iterations are completed, STOP;
  else if the fixed number of the iterations without the improvement regarding the the best global solution is completed, STOP;
  else repeat LOOP.

Figure 1: Tabu search algorithm.

3 TABU SEARCH ALGORITHM APPROACH

The tabu search algorithm (TSA) was developed mainly due to the work of the Polish mathematicians, namely E. Nowicki and C. Smutnicki (Nowicki and Smutnicki, 2005). It belongs to the class of local search algorithms, where a concept of the local neighbourhood of a given solution is considered. The local neighbourhood is defined as a set of solutions that are different from the basic solution with respect to a single attribute. Many different types of neighbourhood can be found in the literature, however the swap and insert types, are considered in this paper.

TSA consists of the subsequent steps to determine the best representative of the neighbour solutions set, where a single step is conducted in each iteration. It may occur that one of the neighbour solutions is superior in respect to the basic solution (i.e. the solution which the neighbourhood is considered at the given iteration). In this case, the step is made towards the direction of the newly found neighbourhood solution. This, therefore, allows the algorithm to alleviate the possibility to be trapped in local extremes of the objective function considered. To avoid repetitive steps between the same solutions in the adjacent extremes, the tabu list consisting of the assumed *a priori* number of the rules regarding the quality of the solutions visited recently, is proposed. Figure 1 provides a more formal description of the proposed algorithm.

There exists, however, an exception when the solution found is superior to the best one to date. In this case the algorithm steps towards its direction unconditionally and the tabu list is erased. It is remarked, that a considerable amount of time is required to estimate even a single solution in each iteration of the algorithm, for the instances relatively large. This leads

START

- Generate the first population.

LOOP

- Find the best chromosome in the current population.
- Update the global best chromosome.
- Choose the representing chromosomes for the crossover.
- Create a new population.
- Execute a swap mutation with the fixed probability.
- Execute a bit mutation with the fixed probability.
- If the fixed number of the iterations are completed, STOP;
  else if the fixed number of the iterations without the improvement regarding the the best global solution is completed, STOP;
  else repeat LOOP.

Figure 2: Genetic algorithm.

to the conclusion that the quality of the first generated solution from which the algorithm starts is essential. In this paper, a modified algorithm is suggested, to handle this task.

The swap move is carried out via the exchange of the positions between two randomly chosen jobs taken from the list assigned to a single machine. Insert one removes a random job from a one machine and allocates it on the second machine, directly before randomly chosen job from its list.

4 GENETIC ALGORITHM APPROACH

For more details on the subject of the genetic algorithms (GAs), see (Schmitt, 2000) and (Davis, 1991). Due to the fact that many different approaches have been described in the literature for solving the problem considered, this section presents the exact parameters of the algorithm implemented.

The more formal definition of the algorithm proposed is presented in figure 2. The crossover operator is the basis of the genetic algorithms, hence, it is essential to focus on its proper implementation. The issue that was shown experimentally in (Bazyluk et al., 2006) for a simpler type of job scheduling problem is that the utilization of a popular PMX leads to an impossibility to obtain satisfying results. Therefore, the multi-component uniform order-based (MCUOX) crossover operator proposed by Sivrikaya and Ulusoy in (Sivrikaya-Serifoglu and Ulusoy, 1999) was chosen for the purpose of this paper.

It is noted, that in the scheme chosen a single gene

accommodates both the object and the associated selection associated with this object. In the case of the problem considered, this refers to the situation where each gene corresponds to the job-machine pair. The construction of a descendant from two parent chromosomes is presented in Figure 3.

START

- With the first position on the parents and the descendant.

LOOP

- Choose one parent randomly.
- Find the first job corresponding to the chosen parent which has not been assigned to the descendant.
- If the machine assigned to the job is the same for both parents, make the same selection for the job;
  else choose one of the machines from the parents randomly and assign the job to it.
- Assign the job-machine pair to the first empty position of the descendant.
- If all genes of the descendant chromosomes are set, STOP;
  else proceed to the next gene on the descendant and repeat LOOP.

Figure 3: MCUOX crossover algorithm.

The mutation implemented comprises two mechanisms. The first, randomly selects two positions on a chromosome and exchanges their contents. The second, incorporates the reassignment of the machine to a randomly chosen job in a chromosome. It is to be noted, that this operation may lead to the same machine selection as before.

The selection of the chromosomes for the crossover from a population builds on a rule that the probability of choosing a chromosome to be a parent in the next generation is proportional to its fitness. The value of the fitness function is for a given chromosome is defined by Equation (9) where K is the size of the population and $F_k = CT_c(k)^{-1}$ the fitness function of the chromosome k that is inversely proportional to the objective function with C being a fixed constant.

Table 1: Experiment parameters.

tabu list size (TSA)	10
population size (GA)	25
swap mutation probability (GA)	0,1
bit mutation probability (GA)	0,05
crossover probability (GA)	0,9

$$P_k = \frac{F_k}{\sum_{k=1}^{K} F_k} \tag{9}$$

The appropriate tuning of the parameters of the GA is one of the crucial issues that influences significantly its efficiency and effectiveness, see (Grefenstette, 1986) for more detailed discussion. For this purpose, in the problem considered, a meta-genetic algorithm with a regular PMX operator was additionally implemented.

5 NUMERICAL ANALYSIS

5.1 Generation of Test Instances

Both implemented algorithms were validated for the same instances of the various sizes considered. Due to the fact that no benchmark problems were found in the literature, such as in particular the presented shape, the test benchmarks were generated randomly. After the tuning of the parameters the following values, presented in Table 1, were set.

5.2 Research Results

The implemented algorithms were evaluated on a IBM-compatible machine, equipped with the Intel Pentium M 740 1,73GHz processor and 512 Mbytes of RAM. The first experiment was conducted for a set of 100 small instances comprising of 10 jobs and 3 machines. The evolution of the average objective function, denoted (T_c), for the both algorithms is depicted in Figure 4. It is observed, that TSA begins with the solution that is nearly twice as large as T_c, however it relatively quickly overtakes GA (first intersection of the graphs in Figure 4). This property can be noted for all the instances consisting of the job sizes 100 or less, in general. In the case of TSA, as opposed to GA, it seems less difficult to determine the optimal solution, but, on the other hand, significantly more computation time is required. This is due to the property of the slow movement across the arguments area of the TSA based algorithm. It is to be remarked, that the aforemention drawback of TSA is not significant for the relatively small instances. It can be expected that the GA would obtain the solution of a similar quality as the TSA after a certain number of iterations, which can be observed in Figure 5 (the second intersection).

In the following investigation the impact is placed to examine the two aforementioned intersection

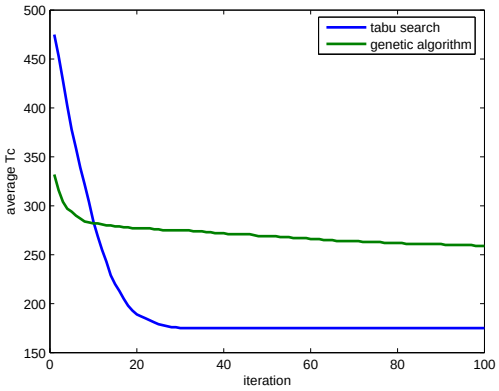


Figure 4: Objective function evolution.

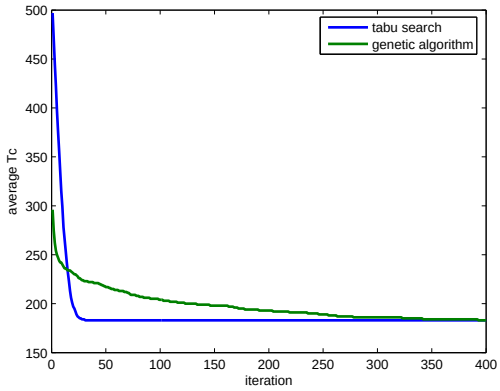


Figure 5: Objective function evolution.

points. In all the experiments conducted the three areas of the objective function evolution were observed:

1. predominance of GA in the first iterations (up to the 1st intersection)
2. predominance of TSA for a given set of iterations (up to the 2nd intersection)
3. predominance of GA again (after the 2nd intersection)

The purpose of the following investigation is to analyze the location of the both intersection points with the accordance to the size of the particular instance.

Analogously to the results presented previously, see Figures 4-5, the first intersection is observed relatively quickly (it is a one of the first iterations) when the instances smaller than 100 jobs and 10 machines are considered. Therefore, the usage of the TSA has proven to be a considerably better choice, when the relatively short calculation time is to be expected. On the other hand, in the cases when the location of the optimum solution is of the prime interest, the GA algorithm is the more appropriate choice. The radical changes are observed for the relatively large instances of 100 jobs or more. In these cases, the TSA is not able to move sufficiently fast across the arguments area and, thus, the first intersection point, after a larger number of iterations, is reached. The example average objective function evolution for 100 instances of 200 jobs and 10 machines is presented in Table 2. It is noted, that with the growth of the instance size, the corresponding time required for a completion of a single iteration increases, accordingly. After 10 iterations it may be readily observed that the TSA evolved from 59509 to 59137 with the decrease of $0,63\%$. On the other hand, the utilization of the GA leads to the decrease from 55828 to 54381 at the ratio of $2,59\%$. The time required for carrying out the experiment in the same configuration setup, for the time

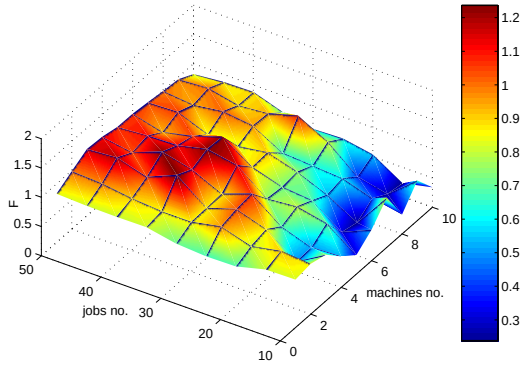


Figure 6: Effectiveness proportion of GA to TSA for different instance sizes.

long enough to reach the first intersection point, was approximately calculated to be few hours.

Table 2: Objective function evolution.

Iteration	T_c (TSA)	T_c (GA)
1	59509	55828
2	59470	55016
3	59396	54859
4	59346	54758
5	59315	54670
6	59271	54539
7	59243	54529
8	59210	54455
9	59174	54410
10	59137	54381

The further work will comprise the investigation of the three-dimensional graphs to illustrate the comparison results as a function of the both attributes of the

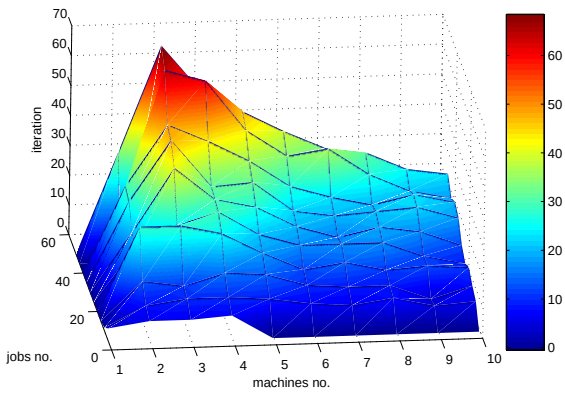


Figure 7: Location of first intersection point for different instance sizes.

instance size i.e. the number of the jobs and the number of the machines. Figure 6 illustrates the proportion, denoted (F), of the objective function value obtained via the GA and the TSA. The experiment was carried out for 100 instances of the each size taken from the set of 1 to 10 machines and 10 to 50 jobs. Both algorithms were investigated for 25 iterations. The value of $F > 1$ on the graph corresponds to the situations where the efficiency of GA was superior to TSA, $F < 1$ otherwise.

Figure 7 presents the average number of iterations, when the first intersection was observed. The first intersection point can be interpreted as a point starting from which the GA is superior over the TSA.

6 CONCLUSIONS

In the paper the problem of the parallel machine job scheduling with the weighted earliness and tardiness has been addressed. Two heuristic algorithms, that proven to be efficient, have been proposed and numerically validated. The investigation of the sensitivity of both approaches as a function of the size of the problem instance, has been carried out. The results obtained suggest that the TSA is appropriate to handle the relatively small and medium instances. On the other hand, utilization of the GA coupled with MCUOX crossover operator, becomes more beneficial with the increase of the problem instances. An important property was observed, namely a significant deterioration of the efficiency of the TSA for instances containing $2-5$ machines in comparison with other values was noted, see Figures 6-7. Considering these figures from the instance of (1 machine, 10 jobs) to (10 machines, 50 jobs) a constant improvement of the GA in comparison with the TSA up to the point of its predominance, can be seen. It has been shown experimentally that the point is located in the vicinity of the instance size of 100 and 200 jobs, see (Figure 5). Furthermore, the execution time of GA with MCUOX increases with a decreasing rate as a function of the increasing problem size whilst the increasing rate is observed for the TSA.

The problem extension could consist of the sequential dependency of jobs and the possibility of introducing the idle time intervals between subsequent execution of jobs. Further improvement of the proposed heuristics can be achieved by considering a hybrid algorithm that inherits the advantages of both approaches i.e. the GA and the TSA.

REFERENCES

Baker, K. and Scudder, G. (1990). Sequencing with earliness and tardiness penalties: a review. *Operations Research*, 38:22–36.

Bazyluk, M., Koszalka, L., and Burnham, K. (2006). Using heuristic algorithms for parallel machines job scheduling problem. *Computer Systems Engineering*.

Cao, D., Chen, M., and Wan, G. (2005). Parallel machine selection and job scheduling to minimize machine cost and job tardiness. *Computers and Operations Research*, 32:1995–2012.

Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.

Du, J. and Leong, J. (1990). Minimizing total tardiness on one machine is np-hard. *Mathematics of Operations Research*, 483–495:1990.

Grefenstette, J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems Man and Cybernetics*, SMC-16:122–128.

Hall, N. and Posner, M. (1991). Earliness-tardiness scheduling problems, i: weighted deviation of completion times about a common due date. *Operations Research*, 39:836–846.

Kanet, J. (1981). Minimizing the average deviation of job completion times about a common due date. *Naval Research Logistics Quarterly*, 28:643–651.

Nowicki, E. and Smutnicki, C. (2005). Metaheuristic optimization via memory and evolution. *Kluwer Academic Publishers*, pages 165–190.

Schmitt, L. (2000). Theory of genetic algorithms. *Theoretical Computer Science*, 259:1–61.

Sivrikaya-Serifoglu, F. and Ulusoy, G. (1999). Parallel machine scheduling with earliness and tardiness penalties. *Computers & Operations Research*, 26:773–787.

DATA MINING AND KNOWLEDGE DISCOVERY FOR MONITORING AND INTELLIGENT CONTROL OF A WASTEWATER TREATMENT PLANT

S. Manesis, V. Deligiannis and M. Koutri
University of Patras,Electrical & Computer Engineering Department, Patras 26500, Greece
{stam.manesis, bdeligiannis, mkoutri}@ece.upatras.gr

Keywords: Intelligent control, Knowledge-based systems, Data mining, Wastewater treatment, Expert-Fuzzy Logic.

Abstract: Intelligent control of medium-scale industrial processes has been applied with success but, as a method of advanced control, can be further improved. Since intelligent control makes use of knowledge-based techniques (such as expert systems, fuzzy logic, neural networks, etc.), a data mining and knowledge discovery subsystem embedded in a control system can support an intelligent controller to achieve a more reliable and robust operation of the controlled process. This paper proposes a combined intelligent control and data mining scheme for monitoring and mainly for controlling a wastewater treatment plant. The intelligent control system is implemented in a programmable logic controller, while the data mining and knowledge discovery system in a personal computer. The entire control system is basically a knowledge-based system which improves drastically the behavior of the wastewater treatment plant.

1 INTRODUCTION

Data mining is a fast growing research field aiming at the extraction of valuable knowledge from massive databases. Due to the increasing use of computing in the context of several applications, data mining can actually be applied to various problems related to the operation of man-made systems and their interactions with other natural ones. These interactions are becoming significantly important as populations are growing and world's sensitivity for the environment is increasing. An area of particular success has been in data mining for wastewater treatment systems and surface water systems (streams, lakes and rivers), where complex problems can be solved that are unsolvable by any other means (Condras et al. 2002; Condras and Roehl 1999). Data mining tools have been adapted for unsteady continuous systems, as wastewater treatment plants where the hydro-dynamical, biological and physical phenomena are highly coupled, in order to monitor the wastewater quality and detect dangerous faults of the process (Victor Ramos et al. 2004).

On the other hand, *intelligent control* (DeSilva 1995; Harris et al. 1993), the discipline that performs human-like tasks in environments of uncertainty and vagueness with minimal interaction with human operators, has had a significant impact in the process industry. The cement industry was, in fact, the first process industry to apply intelligent control techniques in the late 1970s in the form of fuzzy control, and today hundreds of industrial plants worldwide are controlled by such controllers (Boverie et al. 1991; Jamshidi et al. 1993; King 1992).

A fundamental attribute of intelligent control is its ability to work with symbolic, inexact and vague data which human operators comprehend best. Indeed, its ability to deal with incomplete and ill-defined information, an inherent characteristic of wastewater treatment plants, permits implementation of human-like control strategies which have hitherto defied solution by any of the conventional hard control techniques. Fuzzy logic and artificial neural networks (Harris et al. 1993) are two examples of soft computing which have migrated into the realm of industrial control over the last two decades. Chronologically, fuzzy control was the first and its application in the process industry has led to significant improvements in product quality, productivity and energy consumption. Fuzzy control is now firmly established as one of the leading advanced control techniques in use in industry. Over

the last two decades or so wastewater engineering has undergone significant advances in both theory and practice. Experience gained from the operation of numerous wastewater plants, coupled with the results of recent research in the field, has led to improved plant design and wastewater management. Today, effective control of wastewater treatment plants (Rodriguez-Roda et al. 2002; Manesis et al. 1998; Katebi et al. 2000) is of critical importance not only for economic reasons but also to satisfy stringent environmental constraints.

Expert fuzzy control systems have been developed based on human operators' experience, the knowledge of which is acquired by way of extensive interviews. However, the heuristic knowledge of an operator, although it can contain some important consequences about the operational behavior of a plant, can not be based on a large number of measurements and trend diagrams. A data mining and knowledge discovery system can be used to improve or optimize, as well as to evaluate the behavior of the controller. Such a research and development project, called TELEMAC (Lambert 2004; Dixon et al. 2007), within the European IST program is now working on data mining which opens up the prospect of learning from data in order to manage wastewater treatment plants better. Classification techniques for concept acquisition have been also applied in order to build knowledge bases that can help human experts to manage wastewater treatment plants (Serra et al. 1994). Usually, a Data Mining and Knowledge Discovery (DMKD) system (Huang and Wang 1999; Sanguesa et al. 1997; Dixon et al. 2004; Dixon et al. 2007) for process monitoring and control is based on simple measurements of the controlled variables from which association rules may be found. As a valuable addition, a DMKD system based on both measurements and actions of a fuzzy controller is the main idea presented in this work. Particularly, the paper describes a new hybrid scheme in which the induction rules are a priori given but modifiable while the DMKD system searches on both inputs and outputs of the fuzzy controller, records the activity of rules and hence constitutes a part of the overall controller. Experimental results from a four months operation period of the treatment process are presented in last section.

2 WASTEWATER TREATMENT

Wastewater treatment plants typically have two principal stages as shown in Fig.1, the *primary* stage which includes the bar racks, grit chamber and primary settling tank whose objective is the removal of the organic load and solids in the wastewater to a degree of 30-50% and the *secondary* stage whose objective is the biological treatment of the organic load.

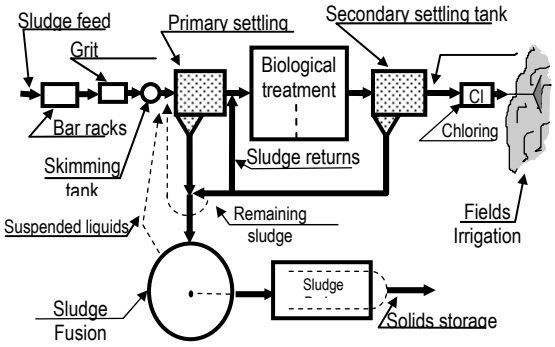


Figure 1: Schematic of a typical wastewater treatment plant.

The removal of organic load (biochemical oxygen demand or BOD, mixed liquid suspended solids or MLSS) in conjunction with secondary treatment performed in the final stage, leads to an overall treatment level of the order of 80-90%. In all wastewater treatment plants it is necessary the oxygen content in the aerated zone to be also subjected to close control. This is achieved by a suitable control strategy involving the following three *manipulated* variables:

(1) the oxygen supply to the aerated zone (O_2Feed)
(2) the mixed liquid returns rate from aerated zone to the anoxic one (*R_ml*)
(3) the sludge returns rate from settling tank to the biological reactor (*R_Sludge*).

The quantities which are appropriately measured by suitable instrumentation and constitute the controlled variables of the plant are:

(1) the ammonia concentration in the reactor ($N\text{-}NH_3$)
(2) the nitrate concentration in the reactor ($N\text{-}NO_3$)
(3) the dissolved oxygen in the reactor (*DO*)
(4) the temperature in the reactor (*TEMP*)
(5) the mixed liquid suspended solids concentration in the reactor (*MLSS*)
(6) the difference in biochemical oxygen demand between the entrance and exit of the secondary settling tank (*D(BOD)*)

These six variables constitute therefore the inputs to

the intelligent controller. By the nature of the process and the interaction of the controlled variables it is obvious that effective control can only be achieved by means of a multivariable controller behind of which complex knowledge must exist.

3 INTELLIGENT CONTROL OF A WASTEWATER TREATMENT PLANT

Having established the principal controlled and manipulated variables, the next task in developing an intelligent controller for a wastewater treatment plant, using linguistic techniques, is to establish a set of linguistic descriptors for each manipulated variable. These are expressions of the type *Very high, High, Low, OK* etc. which are commonly used by plant operators. The *integrity* of an intelligent controller is directly related to the number of such descriptors, bur practical limitations place a limit on this number. The *granularity* of the controller is inversely proportional to the number of linguistic descriptors. Three descriptors are generally sufficient to describe the controller input variables, the *HI* (High), *OK* and *LO* (Low) descriptors all with trapezoidal membership functions. The locations of the centroids of the membership functions can be considered as the modal points of the fuzzy resolution while the number of such modal points corresponds to the number of fuzzy states of the variable. Also, the intermodal spacing of the membership functions is a measure of the resolution of the variable. It is obvious that the overall accuracy of the intelligent control system is directly related to this resolution. In a similar manner, the manipulated variables or controller outputs are allocated by five descriptors *VH* (VeryHigh), *HI* (High), *OK*, *LO* (Low) and *VL* (VeryLow) which provide sufficient fineness of control. For computational simplicity, singletons provide a convenient way to describe the membership functions of the controller outputs where high accuracy is not of paramount importance and also lead to a particularly simple arithmetic procedure for defuzzification.

The inference engine of the intelligent controller manipulates linguistic control rules of the form:

R: if ((*D(BOD)* is Y_1) and (*MLSS* is Y_2) and (*TEMP* is Y_3 and (*DO* is Y_4) and (*N-NH*$_3$ is Y_5) and (*N-NO*$_3$ is Y_6)) then ((*O*$_2$*Feed* is U_1) and (*R_Sludge* is U_2) and (*R_ml* is U_3))

where Y_m and U_n are the linguistic descriptors of the m controller inputs and n outputs respectively, where m $\in$ {1,2,3,4,5,6} and n $\in$ {1, 2, 3}. For the kth linguistic rule, the values of the membership functions corresponding to the process outputs (i.e. the controller inputs) are computed to form the array

$$\mu_1^k(D(BOD)),\ \mu_2^k(MLSS)\ldots\mu_6^k(N-NO_3)$$

the minimum element of which is the *degree of fulfillment* of that rule and is a measure of the contribution of that rule to the final control action, i.e.

$$\sigma_k = \min\left\{\mu_1^k(D(BOD)),\mu_2^k(MLSS)\ldots\mu_6^k(N-NO_3)\right\} \in [0,1]$$

The union of the weighted products of the corresponding membership functions of the controller output fuzzy sets $\{v(\cdot)\}$ is subsequently computed to form the resultant output membership functions. The membership function of the jth controller output is thus the result of the *max* operator:

$$\max\left\{\sigma_1 v_1^j(O_2Feed),\sigma_2 v_2^j(R\_Sludge),\sigma_3 v_3^j(R\_ml)\right\}$$

The engineering values of the controller outputs, necessary to drive the actuators of the plant under control are obtained following the defuzzification controller outputs are described by p singletons the centre of gravity *(COG)* of the jth controller output is simply the inner product procedure. When the membership functions of the:

$$y_j = \langle \varphi_j, z_j \rangle$$

where the coefficients

$$\varphi_j = \frac{\sigma_j}{\sum_{l=1}^{p}\sigma_l} \in [0,1]$$

are the *fractional degrees of fulfillment* while the array $\{z_j\}$ contains the locations of the singletons of the membership functions of the outputs.

A fuzzy system shell (FuzzyControl++ S7 ® by Siemens) was used to develop the intelligent controller of the wastewater treatment plant for a city of 120.000 PE in Greece. The shell uses trapezoidal membership functions, Mamdani *max-min* inference and *COG* defuzzification (Mamdani 1974; Patyra and Mylnak 1996). A Simatic S7-300 programmable logic controller equipped with digital and analogue input-output cards, to which the plant sensors and actuators are directly connected, has

been selected. This actual controller is linked to a host computer necessary not only for the fuzzy system shell implementation but also for the data mining and knowledge discovery procedure which is described in next section.

4 DATA MINING AND KNOWLEDGE DISCOVERY FOR INTELLIGENT CONTROL

With intelligent controller being operated as described above, the controller accepts a stream of data and decides about the actions on the manipulated variables. The generated continuously over time actions, together with the operational conditions constitute a kind of knowledge, remarkably accepted or not, which nobody exploits. Hence, a DMKD system should be introduced in order to acquire and evaluate this knowledge. It is known that the operational data of any industrial process are used by plant operators and supervisors to develop an understanding of a plant operation through interpretation and analysis. In a first level, there are methodologies and tools that automate data interpretation and analysis derived by a large number of measurements. In industrial processes which already operate with an expert-fuzzy rule-based controller the requirements of this first level have been realized. Human operators are more concerned with the current status of the process and possible future behavior rather than the current values of individual variables. Apart from simple measurements, we need in a second level a furthermore analysis of the dataset consisting of the combined output actions and input measurements, which only the expert fuzzy controller can give us during its operation. Therefore, we propose the analysis of the set of the fuzzy control states with the use of DMKD techniques. *Data mining* refers to the extraction of interesting patterns from large amounts of data, while involves the use of techniques from multiple disciplines such as database technology, statistics, machine learning, neural networks, information retrieval, etc. *Knowledge Discovery in Databases* (KDD) concerns a systematic process consisting of a set of well-defined steps. Data mining constitutes a step in the whole knowledge discovery process (Comas et al. 2001; Gibert et al. 2005). The processes controlled by expert-fuzzy controllers are usually slow procedures and hence the DMKD process can operate off-line allowing the human confirmation. The data mining algorithmic procedure will operate as a computational component at the control center and will run periodically to discover knowledge and update the knowledge base.

4.1 Problem Ddefinition and Rrepresentation

Given a wastewater treatment plant along with the corresponding intelligent controller, we define the fuzzy control state as the vector

$$((m_1, m_2, m_3, m_4, m_5, m_6), (u_1, u_2, u_3), t_k)$$

where m_i, $i=1,...,6$ are the measurements of the inputs for which the fuzzy controller decides the actions u_j, $j=1,...,3$ for the corresponding outputs at the time stamp t_k, $k= 1, 2 ..., \infty$. According to this description, a set of fuzzy control states could be stored in a relational database on which data mining could be performed. A relational database is a collection of tables, each of which is properly named. In our case there is one table representing the set of fuzzy states (see Table 1). As an example, the fuzzy state s_1 is defined as

$$s_1: ((HI, OK, OK, LO, OK, OK), (VH, OK, LO), 2)$$

The table of fuzzy control states results after removing the erroneous data and integrating information from various data sources, in particular from the measurements and actions, as well as from the set of fuzzy rules. Particularly, the rule activity and the degree of rule fulfillment are also stored in database. This process constitutes the first step of the overall knowledge discovery process. The second step concerns the data selection and transformation into meaningful representations. In our case, the time stamp may partially eliminated, because we are interesting in the behavior of the system during a period of time rather than a particular time instance. For example, weather conditions for certain periods of time may affect the plant operation. Changes in rainfall, temperature and humidity must be recorded with time stamp and correlated with the rest of mined data.

Table 1: Fragment of records from the database for fuzzy control states

	m_1	m_2	m_3	m_4	m_5	m_6	u_1	u_2	u_3	t
s_1	HI	OK	OK	LO	OK	OK	VH	OK	LO	2
s_2	HI	OK	HI	LO	OK	LO	VH	LO	LO	3
s_3										
…										

4.2 Mining Interesting Patterns

Data mining methods and techniques could be applied into the set of cleaned records, in order to discovering interesting patterns. *Association rule mining*, *clustering*, and *classification* seem to better serve the needs of the particular problem of controlling a plant. Association rule mining aims at finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories (Vazirgiannis et al. 2000). An association rule is a statement of the form $A \Rightarrow B$, where A and B are disjoint subsets of a set of items. The rule is accompanied by two meaningful measures, *confidence* and *support*. Confidence measures the percentage of transactions containing A that also contain B (i.e. *confidence* $(A \Rightarrow B) = P(B \backslash A)$). Similarly, support measures the percentage of transactions that contain A or B (i.e. *support* $(A \Rightarrow B) = P(B \cup A)$) (Han and Kamber 2001).

The application of association rule mining algorithms in the relational database of the control states of the wastewater treatment plant could produce interesting results. If the derived set of association rules contains some rules not already encountered into the input data set, then these new rules could be embedded into the inference mechanism of the expert-fuzzy controller. Techniques of rule induction are also being applied to estimate values of sensors readings based on more easily obtained values, and to determine how reliable the models remain over time. Rules may be generated in forms such as the following,

> *Variable (DO) falls in a particular range of low values if variable D(BOD) falls within a particular range of high values*

accompanied by an indication of the degree of satisfaction of the rule.

Subsequently, the clustering concerns the process of grouping a set of physical or abstract objects into classes of similar objects. So, a cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters (Han and Kamber 2001; Gibert et al. 2005). The application of clustering into a set of data objects requires that the data objects are not class-labeled. In our specific problem a state, consider for example s_1 (see Table 1), constitutes a data object. This data object is labeled by the values of the three output variables, u_1, u_2, and u_3. The unlabeled data object corresponding to s_1 is:

$$s_1\text{: (HI, OK, OK, LO, OK, OK)}$$

The application of clustering algorithms to the set of unlabeled states could conduct to the identification of states of similar behavior and, thus, to the derivation of theoretical and more general rules.

Lastly, classification could be considered as a function that maps (classifies) a data object into one of the several predefined classes (Vazirgiannis et al. 2000). That means (i) a well-defined set of classes, and (ii) a set of pre-classified data objects are required. Consequently, classification is a two-step process: learning and classification. During the learning sub-process, the set of labeled data objects is analyzed using a classification algorithm and a classification model is derived. Next, during the classification sub-process, the model can be applied to the new unlabeled data objects for inferring their classes. To be clearer, in our case, a classification algorithm is firstly applied on the set of cleaned control states in order to find a classification model. Next, whenever a new unlabeled state appears, the model is applied for classifying this new state; in other words for deciding the actions u_1, u_2, and u_3. Classification is, thus, another way for creating new rules, for testing and modifying the existing ones.

4.3 Data Mining Tools

The selection of a commercial data mining tool depends on various similar parameters, such as:

- system issues (like operating system, client-server architecture, etc.)
- support of different types of data sources (ASCII files, relational databases, ODBC connections)
- support of various data mining algorithms
- visualization of the resulted patterns
- price
- ease of learning to use

For the wastewater treatment plant application described above and particularly for a plant of medium size we need a stand-alone PC architecture with windows operating system, to support only flat files with numerical data and various mining algorithms while visualization in not necessary. Some examples of data mining tools are IBM Intelligent Miner, SGI MineSet, Clementine (SPSS) and GESCONDA (Gibert et al. 2005), from which the first one has been selected to implement the current project.

5 EXPERIMENTAL RESULTS

In this work, we used the time series data of four months (April 2006 to July 2006) measured or recorded at the medium-size wastewater treatment plant mentioned above. As one example for all of these data records the measurements in the influent D(BOD), N-NO_3 and DO are shown in Fig.2. Zooming into this plant's monitoring in order to achieve better resolution the corresponding measurements are shown in Fig.3 and Fig.4 in a two-day and one-day section for D(BOD) and N-NO_3 concentrations respectively. The set of acquired data includes also the actions of the fuzzy controller which are recorded in the database by means of rules activity. Figs.5, 6, 7 show the membership functions and the input value appearance frequency for the D(BOD), N-NO_3 and DO respectively. The

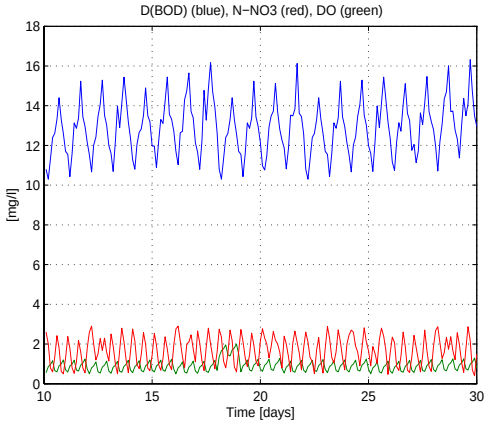


Figure 2: Measured D(BOD), N-NO3 and DO concentrations during the period 10/6 – 30/6 2006.

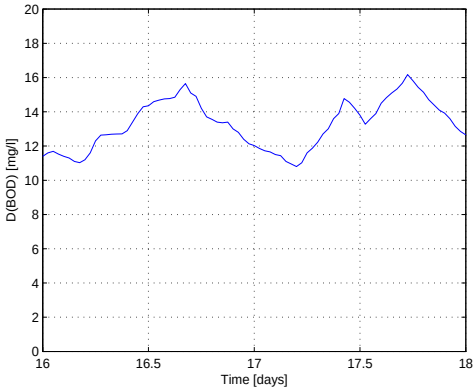


Figure 3: Two-day section of D(BOD) concentration from the four months data (16/6-18/6 2006).

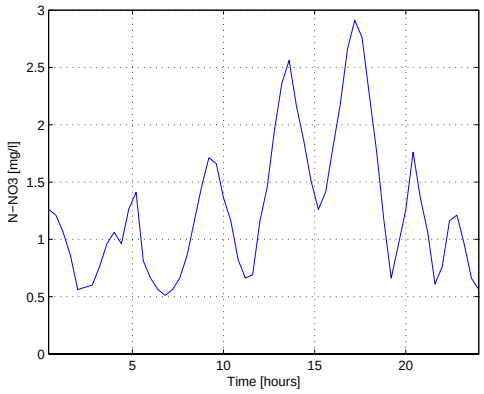


Figure 4: One-day section of N-NO_3 concentration from the four months data.

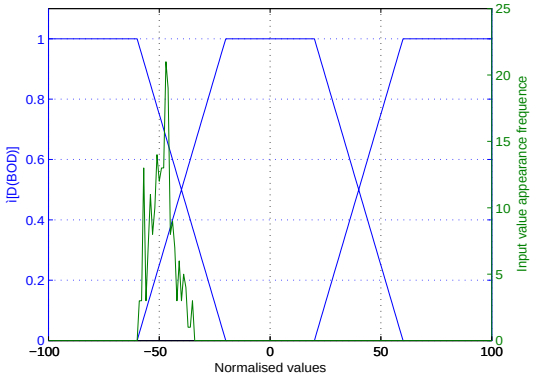


Figure 5: Membership functions and sampled measurements for D(BOD).

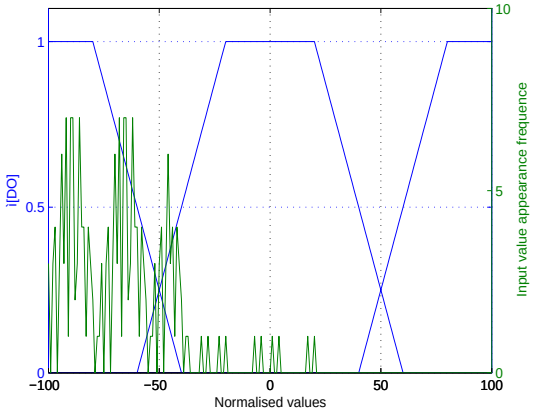


Figure 6: Membership functions and sampled measurements for DO.

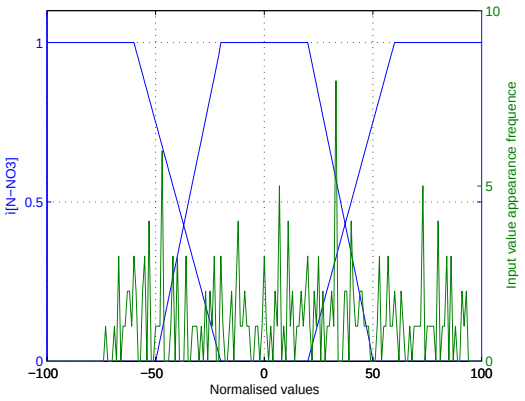


Figure 7: Membership functions and sampled measurements for N-NO_3.

The values in these three figures are normalized into the domain [-100, 100] since the six inputs variables get values in different domains and hence uniformity is needed. By inspection of Fig.5 one can deduce that the most of the reformatted measurements belong partially in both LO and OK membership functions. This means that a reconfiguration of the defined membership functions is required in order to have a set of membership functions which will cover effectively the actual values of D(BOD) in the real process. The narrow distribution of the sampled measurements into the heuristically predefined membership functions indicates furthermore that reconfiguration is necessary. Another conclusion concerns the small variation of D(BOD) measurements in comparison to the corresponding predefined range of D(BOD) values and therefore has to be reconsidered. Fig.6 shows a more uniform distribution of the sampled measurements of DO, while the best distribution of the obtained measurements is depicted in Fig.7 concerning the N-NO_3 variation.

5.1 Rule Activity-Modification

As mentioned above, the second step of the overall knowledge discovery process concerns the data selection based on statistical criteria and the transformation of them into meaningful representations. As a first approach to obtain cleaned data for knowledge creation, the rule activity has been recorded in DMKD off-line module and some results are shown in Fig.8 and Fig.9. Each figure has five subplots for equal number of rules. For each rule, the number of appearances is shown as a function of the degree fulfillment. For example, rule No.5 was satisfied three times with a 20% degree of fulfillment. Fig.8 indicates the least active rules while Fig.9 shows the corresponding most active ones.

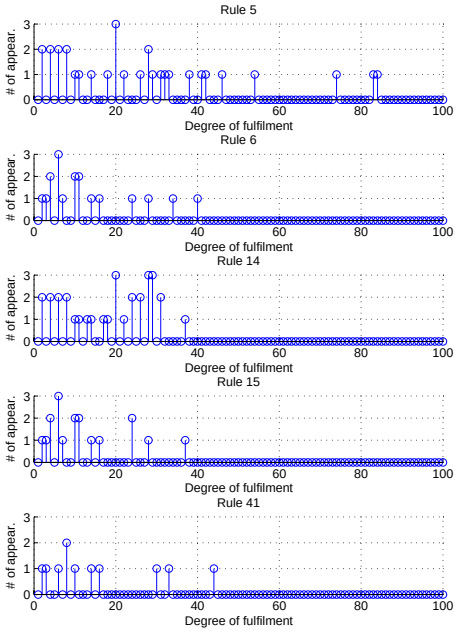


Figure 8: Statistical analysis of the least-active rules behavior.

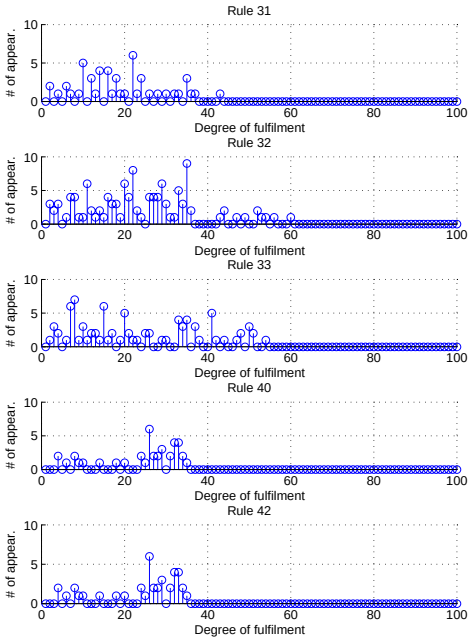


Figure 9: Statistical analysis of the most-active rules behavior.

6 CONCLUSIONS

Although wastewater treatment plants are implemented with properly functioning control loops concerning the biological process, in practice, this type of plant requires a major time investment on the side of the operator, involving many manual operations. These difficulties can be overcome by an intelligent controller which incorporates the human experience. The mined data, characterized as multivariate and interrelated, constitutes a combination of measurements of the process's variables and actions of the controller. The consequences of the mining and knowledge discovery procedure are used to adapt the soft structure of the intelligent controller through a semi-automatic scheme that provides deeper understanding and better operation of the controlled plant. The experimental results give us basic directions to improve the operation of the control system but it is obvious that a longer validation period of data monitoring and processing is needed.

REFERENCES

Boverie, S, Demaya, B and Titli, A, 1991, Fuzzy logic control compared with other automatic control approaches, *Proceedings of 30th CDC Conference*, Brighton.

Comas, J., Dzeroski, S., Gibert, K., Roda, I.R. and Sanchez-Marre M., 2001. Knowledge discovery by means of inductive methods in wastewater treatment plant data", *AI Communications*, Vol.14(1), pp.45-62.

Condras, P, Cook, J and Roehl E, 2002, "Estimation of Tidal Marsh Loading Effects in a Complex Estuary" *American Water Resources Association Annual Conference*, New Orleans.

Condras, P and Roehl, E, 1999, Real-time control for matching wastewater discharges to the assimilative capacity of a complex tidally affected river basin *South Carolina Environmental Conference*, Myrtle Beach.

DeSilva, C W, 1995, *Intelligent Control, Fuzzy Logic Applications*, CRC Press, Boca Raton.

Dixon, M., Gallop, J.R., Lambert, S.C., Lardon L., Steyer P. and Healy J.V., 2004. Data mining to support anaerobic WWT plant monitoring and control, *Proceedings of the IFAC Workshop on Modelling and Control for Participatory Planning and Managing Water Systems*, Venice Italy,.

Dixon, M., Gallop, J.R., Lambert S.C. and Healy, J.V., 2007. Experience with data mining for the anaerobic wastewater treatment process, *Environmental Modelling & Software*, Elsevier, , vl.22, pp.315-322.

Gibert, K., Sanchez-Marre, M. and Flores X., 2005. Cluster discovery in environmental databases using GESCONDA: The added value of comparisons, *AI Communications,* Vol.18(4), pp.319-331.

Han, J and Kamber, M, 2001, *Data Mining: Concepts and Techniques*, San Francisco, Morgan Kaufmann.

Harris, CJ, Moore, CG and Brown, M, 1993, Intelligent control-aspects of fuzzy logic and neural nets, NJ, World Scientific Publishers.

Huang, YC and Wang, XZ, 1999, Application of fuzzy causal networks to wastewater treatment plants, *Chemical Engineering Science* 54:2731-2738.

Jamshidi, M, Vadiee, N and Ross, TJ, 1993, *Fuzzy Logic and Control–Intelligentware and Hardware Applications*, NJ, Prentice Hall.

Katebi, R, Johnson, M and Wilkie, J, 2000, The future of Advanced Control in Wastewater Treatment Plants, *Proceedings of the CIWEM Millennium Conference* Leeds UK.

King, R, 1992, Expert supervision and control of a large-scale plant, *Journal Intelligent and Robotic Systems*, 2(3).

Lambert, S, 2004, (CCLRC-Data mining Resp.) Telemonitoring and advanced tele-control of high yield wastewater treatment plants, TELEMAC, R&D project, No. IST-28156.

Mamdani, EH, 1974, An application of fuzzy algorithms for the control of a dynamic plant, *Proceedings IEE* , 121(12).

Manesis, S, Sapidis DJ and King RE, 1998, Intelligent Control of Wastewater treatment Plants, *Journal Artificial Intelligence in Engineering* (12):275-281.

Patyra, MJ and Mylnak, DM, 1996, *Fuzzy Logic Implementation and Application*, NJ, J. Wiley-Teubner.

Rodriguez-Roda, I, Sanchez-Marre, M, Comas, J, Baeza, J, Colprim, J, Lafuente, J, Cortes, U and Poch M, 2002, A Hybrid Supervisory System to Support WWTP Operation: Implementation and Validation, *Water Science and Technology Journal* (45): 289-297.

Sanguesa, R, Cortés, U and Béjar, J, 1997, Causal dependency discovery with posscause: an application to wastewater treatment plants, *Proceedings of the 1st International Conference on the practical application of knowledge discovery and data mining*, London.

Serra, P, Sànchez, M, Lafuente, J, Cortès, U and Poch, M, 1994, DEPUR: a knowledge-based tool for wastewater treatment plants, *Engineering Applications of Artificial Intelligence*, 7(1):23-30.

Vazirgiannis, M and Halkidi, M, 2000, Data Mining: Concepts and Techniques, Athens University of Economics and Business, TR HELDiNET 10.

Victor Ramos, J, Goncalves, C and Durado, A, 2004, On-line Extraction of Fuzzy Rules in a Wastewater Treatment Plant, *Proceedings of the 1st International Conference on Artificial Intelligence Applications and Innovations*, 1:87-102.

IMPROVEMENTS IN THE FIELD OF DEVICE INTEGRATION INTO AUTOMATION SYSTEMS WITH EMBEDDED WEB INTERFACES

[1]Anton Scheibelmasser, [2]Jürgen Menhart
[1]*Department of Automation Technology, CAMPUS 02, Körblergasse 126, 8021 Graz, Austria*
anton.scheibelmasser@campus02.at
[2]*Test Bed Automation and Control Systems, AVL List GmbH, Hans List Platz 1, 8020 Graz, Austria*
juergen.menhart@avl.com

Bernd Eichberger
Department of Electronics, Technical University Graz, Inffeldgasse 12, 8010 Graz, Austria
bernd.eichberger@tugraz.at

Keywords: Measurement device, automation system, device integration, embedded web interface, automotive test bed.

Abstract: Web-Technologies which came up in many fields of automation seem to be a solution which improves device integration in many ways. On the one hand the used Ethernet improves the installation techniques with reliable and approved network cables and routing devices. On the other hand the used internet protocols provide several services for the application software development. With the introduction of those services, the local controller of the measurement devices has to execute complex communication protocols in addition to the device specific tasks. This fact has serious influences on the measurement device instrumentation and the execution of the device firmware. Concerning new developments and compatible adaptations of existing instruments several ways for the integration of web technologies are available. The following article is intended to explain the architectural aspects of device integrations using Industrial Ethernet by means of an embedded web server. As a practical example to this architecture, concepts and results of a new developed communication module called EWI (embedded web interface) are given to demonstrate the improvements in measurement device integration in the field of automotive test bed automation.

1 INTRODUCTION

Automation systems like test beds in the automotive industry are used for the development and the quality control of combustion engines. In order to evaluate the interesting quantities, a lot of measurement devices have to be controlled to acquire the data during a test run. From the automation system's point of view these measurement devices can be seen as automata incorporating a finite state machine. Integrating such devices in an automation system means to synchronise the states of the measurement device with the states of the automation system. As those devices are physically separated from the test bed, synchronisation has to be performed by means of communication lines. From this point of view device integration can be split into software aspects like device driver development and hardware parts like network design and installation. Concerning the network aspect, web technologies have spread over in many fields of application. As the components have become cheap and reliable, the well defined Ethernet (Metcalfe, Boggs, 1976) was introduced in the field of automation technology. Because of the standardized bus system, network cables and components like plugs, transceiver, switches, routers and gateways, Industrial Ethernet (Hirschmann, 2007) has established in several fields of automation technology. In many applications it was extended to real time derivates (Powerlink, 2003) of this standard. Particulary automation systems in the field of combustion engine development, called test beds, are a potential field for the use of Industrial Ethernet.

As the test bed automation system has to control a lot of measurement devices, there are high requirements concerning the communication. A central task of the system in this context is the synchronisation of the used measurement devices. Synchronisation is needed in order to drive the internal state machine of the devices in the desired state of the automation system (e.g. measurement). On a typical test bed the measurement devices are seperated from the automation system. Therefore synchronisation is only feasable by means of communication lines. So measurement device integration into automation systems can be split up into two parts.

The first one is the network part. A lot of measurement devices have to be connected to the test bed by means of communication lines. Typically existing test bed measurement devices are connected by means of peer to peer connections (e.g. RS232). This causes problems concerning the available connectors at the automation system and problems related to the available data acquisition rate. Based on traditional master/slave communication protocols and low data rate connections (e.g. 9600 Baud), timing requirements for high dynamic combustion engine tests are not met. So improvements in these directions are required.

The second part of device integration is the software part. Based on generic device drivers (Scheibelmasser, Traussnigg, Schindin, Derado, 2004) the protocol layer integration can be managed generically. Based on a Measurement Device Description (MDD) a flexible and efficient way for the device driver development is applicable. Concerning the appropriate graphical user interface (GUI) a similar methode is not available. Typically, a lot of software components have to be developed for different plattforms to provide the user with the appropriate visualisation for every measurement device. This implies high efforts and an economical disadvantage.Therefore new concepts in terms of device visualisation are needed. In both parts of measurement device integration Industrial Ethernet promises an improvement. In case of network problems, Industrial Ethernet provides the user with a high speed bus system. This avoids the connector problems and the data rate restrictions. Concerning the software part, the available Ethernet protocols enables a lot of services (e.g. http, ftp) which are available for implementing economical and high quality user interfaces and additional improvements in the field of service and maintenance.

2 INDUSTRIAL ETHERNET

If we introduce the Ethernet in an automation system, we provide the user with a lot of new capabilities. This chapter is intended to summarize the characteristics of such communication lines.

2.1 Bus Features

In contrast to the traditional peer to peer connections between the automation system and the measurement device, a bus system like Ethernet provides the capability to connect several devices to one line (network). Additionally, Ethernet increases the useable bandwidth in high ranges (e.g. GHz). Therefore existing restrictions concerning the connectors on the host system or the data rate acquisition problems could be easily solved. A possible disadvantage of this solution lies in the fact that the bus feature implies the need of a bus arbitration which is responsible for loosing a deterministic bus response time. Therefore an extension of the classical standard is necessary if hard real-time is required. In most cases the user interfaces require only soft real-time. So the standard Ethernet protocol is sufficient to support these applications.

2.2 Network Capability

Based on the standard and the defined network protocol stack, measurement devices which are connected to the Ethernet are able to communicate not only in the local network with the automation system but worldwide in the web. This feature enables on the one hand improvements in the area of remote control, remote maintenance and updating parameters or software in the field. On the other hand the security aspects will conquer against these features and lead to protected small subnets locally to the automation system. Depending on the required application the security aspect has to be taken into consideration and should be adjusted to the needed level.

2.3 Service Aspects

If we use the Ethernet with the standardized protocols (e.g. TCP/IP, UDP, HTTP) to connect the measurement devices, a lot of services are available to improve the human machine interface. Using these services, the challenges of the measurement device visualisation could be improved in many ways by means of embedded web-server.

Technologies like JAVA or .NET are available to program the appropriate user interface. The graphical user interface (GUI) of such a device is shown within a web-browser window independent from the computer platform. The respective program instructions (e.g. JAVA Applet) for the visualisation and control are stored in this scenario in the measurement device. In addition to this, user manuals, service instructions or electronical data sheets are available inside the device and accessible by means of Ethernet protocols (e.g. ftp). The challenge of implementing this method lies in the appropriate hardware and software design of measurement device.

3 SYSTEM ARCHITECTURE

In contrast to the traditional measurement device visualisation, the development of additional PC-Programs for control and visualisation could be avoided if we are using the Ethernet technology. As mentioned, the required human machine interface is implemented by means of applets which are stored in the respective device and are executed within a web browser. The details of the applets and the methods used for the implementation should not be part of this work. The main focus of this paper concerns the communication aspects of an embedded web server solution. Using the Ethernet protocols for the measurement device communication increases the efforts for the communication task in the device. This fact has two consequences. Existing devices are not able to carry out this additional task because of their limited resources. New device developments have to introduce additional hardware which is capable to execute not only the application program but an operating system with the Ethernet protocol stack. In addition to the protocol stack, a files system and a multitasking support are necessary to support the desired web-services. Concerning the control aspects of the measurement device, we have to take care not to loose the real time aspects. From the economical point of view we have to consider the side effects of introducing an operation system (e.g. licences) and the consequences in terms of additional hardware (e.g. memory). Never the less a compatible solution for existing measurement devices is required additionally. A further aspect lies in the fact that the introduction of a fast bus connection will not automatically yield to a higher data acquisition rate. Typically, existing firmware and protocols limit the performance of the system. So a appropriate hardware architecture is necessary to support the demanded improvements.

Based on a principle which was already implemented in a density meter (Röhrer, 1991) there exists a trade off concerning the improvement of the device communication. If we separate the non real-time tasks like human machine interface from the real-time control tasks of a system with the introduction of two independent execution units (e.g. computers), we can achieve a solution which provides a lot of advantages. Using this concept, we can use even existing small and economical controller solutions without any modifications to carry out the real time control task. In contrast to this, improved communication or human machine interaction is done by an additional device computer which communicates with the real time controller on a proprietary and efficient protocol (e.g. SPI). This architecture provides the system not only with real time advantages but also with economical benefits.

4 TEST BED INTEGRATION

Based on the above mentioned considerations the device integration of measurement devices in an automotive test-bed automation system should be shown. An additional computer called EWI is introduced which works as a gateway between the traditional RS232 lines of the measurement device and the Ethernet. Based on this component, a new integration concept was developed.

4.1 EWI Definition

The main part of the integration concept is a standalone working EWI (embedded web interface). According to the different aspects of the measurement devices (e.g. version, protocol, hardware, technology), this system offers an overall and compact solution for a new and standardized integration concept for measurement devices in automotive automation systems. The EWI was development on the basis of a few important concepts:

- An increase of the data communication rate
- A multi user access from the client systems
- Introduction of a new visualisation concept

The EWI offers different solutions for data communication and exchange and is implemented as an additional hardware component. Therefore it is even applicable for existing devices. The highest priority is set on the communication between the different users. As shown in Figure 1, it takes care of

the direct communication between the measurement devices (MDS) and different client systems (e.g. test bed system, personal computer, handheld system). The web interface provides the user on the one hand with a gateway function (Ethernet to RS232), on the other hand it offers a data storage for a device specific user interface (GUI), directly stored in an extended memory.

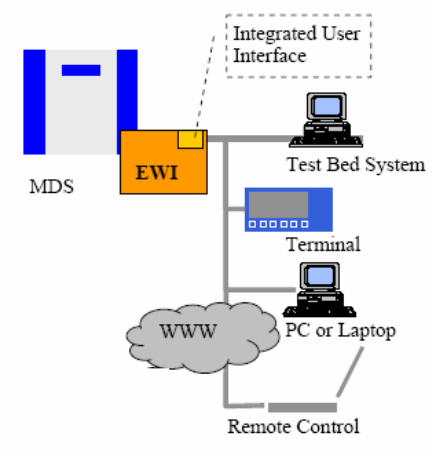


Figure 1: EWI integration concept (AVL, 2007).

Depending on the hardware version of the measurement device, the EWI hardware supports two integration concepts. The first scenario supports existing devices and provides the user with an external component, equipped with a serial connection to the device and a power supply. In contrast to this, the EWI-integration in current measurement devices is done in the system. In this case the communication will be done directly between the microcontroller and the EWI using a high speed serial connection (e.g. SPI). So there is no need for an additional power-supply.

4.2 Economical Aspects

One of the first steps in the EWI-development was the consideration of economical aspects. In order to make a decision, available technologies on the market were checked against aspects like initial costs, time to market or training efforts for the development team. To reach a high reuse of the module, the EWI should be customized by means of text-based configuration files. The following characteristics are configurable:

- Communication protocol definition/settings
- Multi-user access handling
- Automatic device detection
- Boost data handling

Based on this text file and the generic structure the EWI firmware allows the use of the system even with different kinds of ASCII based protocols on the RS232-line. A further economical aspect of the EWI development was the intention to improve the device user interface including the visualization. Currently a lot of different device user interfaces have to be offered to the customer. Depending on the supplier of the test bed system, the operating system on the client, software versions or languages appropriate software have to be provided. Every component of such an interface has to be maintained, bug fixed and released separately. The user has to be trained on each of the user interfaces because the integration into the customer's test bed has to be done always in a different way. So a main goal was to implement the user device interface as a web-page inside the EWI and to use a standard web-browser instead of proprietary PC-programs.

4.3 EWI Hardware Platform

In consideration of the above mentioned aspects the OEM-module solution was chosen. The EWI hardware acts as a communication co-processor solution. It is implemented as an additional hardware component and could be used without any influence on existing controller solution. Therefore it is applicable even for existing devices in the field. Concerning initial- and development time and costs, easy integration of software modules, functionality, licenses and RTOS features the OEM-module was the best trade-off. So the EWI was build up on the IPC@CHIP SC13. The system is based on a 80186 single chip computer and offers a maximum flexibility and availability for individual applications. The module incorporates a real time operating system, RAM, FLASH-Disk, serial and Ethernet controller and a large number of I/Os. The EWI software and architecture was built up on the preinstalled real time operating system. By means of the generic application software it is now possible to integrate the EWI in the different measurement systems, to store and manage the device user interface and to communicate with the measurement devices via multiple access.

4.4 Performance Considerations

Members of the automotive industry and their customers use a common known communication protocol called AK-Protocol (Arbeitskreis, 1991). This protocol is based on ASCII and provides an easy way to communicate with the measurement

devices on the test bed. It has been used for a long time and is described in user manuals of test bed devices. The communication is typically done via a point to point serial line connection in a master/slave mode. As the test bed devices are typically equipped with two serial lines, a connection for a maximum of two users is feasible. The AK protocol offers the user several commands, to switch the device into another mode (set), to acquire data (acquire) or setup new parameters (store) on the system. The interpretation of the AK protocol for a customer is very complex and varies between several measurement devices. To present an easier control mechanism for the user, the supplier of such a measurement system includes device user interfaces on his test bed systems or PCs. These software components are in communication with the measurement devices and offer the user a simplified view of the system. The communication rate between the participants depends on the client system. It varies between 1 Hz up to 10 Hz on current systems in the field. The communication rate, the operating systems and the software platforms are totally different. According to these differences, the standardized EWI integration concept has to consider the different communication rates in order o offer an overall solution in terms of test bed integration. A closer look to the protocol timing shows that the bottleneck of the communication can be found in the measurement devices itself. In current measurement devices the processing time takes about 100 ms. Due to the used hardware this time is necessary to receive the serial line string of the ASCII protocol, to interpret the frame, to calculate the data and to send it back to the client on the serial line. In comparison to this time, the communication of a command on the serial line between the client and the measurement device takes about 25 ms. It must be considered that the EWI is now a third partner in the communication line between the host and the client, and should not decrease the data rate between these communication partners. To improve the communication between the client and the host system, the EWI concept is split up into two parts.

The first part is the increase of the communication data rate between the client and host system by means of changing the physical layer and the protocol to Ethernet and TCP/IP.

The second part of the concept increases the communication data rate between the EWI module and the measurement device system. Instead of 9600 Baud the communication was increased to a data rate up to 115.2 kBaud. In this case the elapsed time can be decreased to 5.5 ms instead of 25 ms (Figure 2). The new communication schema offers now an overall communication time of about 105.5ms between the client and the host system.

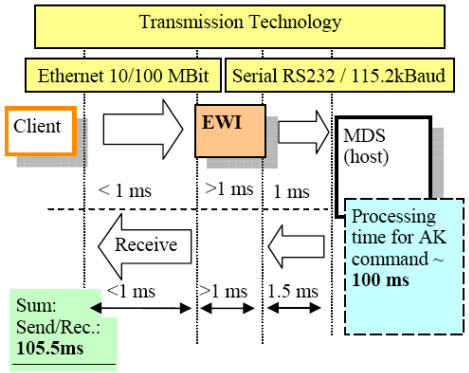


Figure 2: Transmission rates (Menhart, 2007).

Without changing the existing controller hardware in the measurement device, no further improvement of the transmission time is possible. Due to this fact the data rate for existing devices is limited up to 10Hz. But a main demand of the integration concept was the ability to process incoming data from the measurement device with a data rate of 50 Hz. In order to achieve this, the communication schema was extended with a so called Boost mode which was implemented in parallel to the ASCII based communication between the host and the client.

4.5 Data Acquisition/Boost Mode

As mentioned, it will be hardly possible to reach a data rate up to 50 Hz based on an ASCII based AK protocol without a change directly in the measurement device. The bottleneck between the communication structure – client, EWI, host - is the processing and interpretation time of the AK strings in the measurement devices itself.

The solution for this problem is a concept based on a mixed mode of communication. A transparent standardized ASCII mode and a fast binary data mode called "*boost mode*". To achieve the requirement of a measurement data rate up to 50 Hz the communication will be done via this boost data mode in form of binary data sent from the measurement devices to the EWI. The data communication will be started and controlled from the measurement device, independent which client is in communication. The goal of the solution is to relieve the device from processing ASCII based AK data acquisition frames. Therefore the most important data of the measurement devices – the measurement data – will be sent to the EWI via binary data. In case of a client data acquisition the

data are taken from the EWI and no AK frame to the device is invoked.

In parallel the standard communication like parameter settings or service/maintenance commands will be done via the standard AK protocol. By means of this concept a dual communication is established. The processing time of the measurement device is not influenced and the client receives the AK data with a transmission rate up to 50 Hz. Due to the fact that the communication between the clients and EWI is now done via Ethernet, a multiple access from the clients to this boost data and the standard AK commands is possible. (Figure 3).

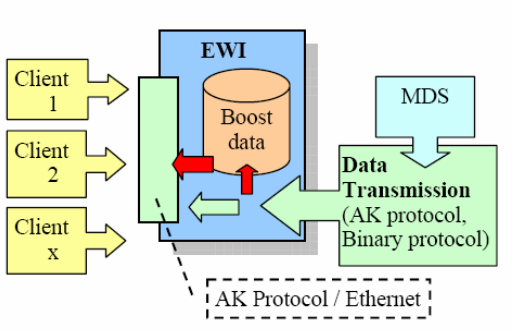


Figure 3: Multiple client access via EWI (Menhart, 2007).

5 CONCLUSIONS

In general the embedded web interface (EWI) represents a uniform and excellent solution for the integration of measurement devices into automotive test bed systems. It offers a complete support of the Ethernet technology as well as a standardized solution to overcome the problems of existing data transmission and communication modes. Providing the user with an Ethernet connection on the measurement device solves a lot of problems. As the peer to peer connections of RS232 lines are replaced with high speed, low cost and high reliable bus connections, the network problems are sufficiently solved. The used TCP/IP protocol offers the application software a lot of services. These services are able to improve not only the communication rate but even the user interface by means of embedded web-server technologies. Implementing such server in a measurement device enables the devices to provide their data not in traditional protocol frames but as web-service on a graphical web-page accessible with a standard web-browser.

The EWI concept, implemented as a communication co-processor has shown excellent results in the field tests. The used EWI principles and methods are able to establish a standardized way of measurement device integration into test bed automation systems.

REFERENCES

AVL List GmbH, 2007, *Embedded Web Interface,* Test Bed Instrumentation Systems, www.avl.com

Röhrer, R., 1991. *Intelligente Instrumentierung von Messgeräten,* Berichte der Informationstagung ME91, pp.46-50, Fric-Verlag

Scheibelmasser A., Traussnigg U., Schindin G., Derado I., 2004, *Device Integration into automation systems with Configurable Device Handler,* ICINCO 04 Conference, Setubal

Metcalfe R.,Boggs R., 1976, *Ethernet: Distributed Packet, Switching for Local Computer Networks,* Association for Computing Machinery, Vol19/No5

Hirschmann, 2007, *Industrial Ethernet,* http:// www.hirschmann-ac.de/Deutsch/industrial-ethernet/ index.phtmlww.hirschmann.com

Menhart J., 2007, *Measurement device integration via Embedded Web Interface,* Diploma Thesis CAMPUS02 University of Applied Sciences, Graz

Arbeitskreis der deutschen Automobilindustrie, 1991, *V24/RS232 Schnittstelle – Grundsätzliches,* UA Schnittstelle und Leitrechner

Powerlink, 2003, *Ethernet Powerlink Standardisation Group,* www.ethernet-powerlink.org/

A GENETIC ALGORITHM APPLIED TO THE POWER SYSTEM RESTORATION PLANNING PROBLEM

A Metaheuristic Approach for a Large Combinatorial Problem

Adelmo Cechin, José Vicente Canto dos Santos, Arthur Tórgo Gómez and Carlos Mendel
Pipca - UnisinoS, Av. Unisinos 950, São Leopoldo, Rio Grande do Sul, Brazil
jvcanto@unisinos.br

Keywords: Genetic Algorithms, Electric Power Systems, Power System Restoration Planning Problem.

Abstract: This work reports the use of a Genetic Algorithm (GA) to solve the Power System Restoration Planning Problem (PSRP). The solution to the PSRP is described by a series of operations or a plan to be used by the Power System operator immediately on the occurrence of a blackout in the electrical power supply. Our GA uses new initialization and crossover operators based on the electrical power network, which are able to generate and maintain the plans feasible along GA runs. This releases the Power Flow program, which represents the most computer demanding component, from computing the fitness function of unfeasible individuals. Results for three different electrical power networks are shown: IEEE 14-Bus, IEEE 30-Bus and a large realistic system.

1 INTRODUCTION

The Power System Restoration Planning Problem (PSRP) can be defined as the search for an optimal sequence of control actions leading a faulty electrical power system from a restoration state of operation to a secure state. The secure state is defined by a normal energy supply with all operational limits observed. The transition of one state to the other is performed through a series of commands sent by the power system operator to the power system. Each faulty situation demands a specific sequence of commands to bring, as fast as possible, the power system back to the secure state. These operations include the connection and disconnection of line sections without overloading the electrical system components.

Further, the PSRP is a multistage problem, being the objective of each stage the reestablishment of the service to a group of priority loads. The solution must obey additional constraints such as those placed by a priority chain in the energy supply. For instance, first hospitals must be attended, then public services, and so on, if distribution context is considered.

However, the main constraint is the time gap while consumers are without energy, which must be kept as small as possible.

The Power System Restoration Planning may be carried out *off* or *on-line*. *Off-line* plans are typically based on previous operator experiences in restoring a faulty system. The efficiency of this process rests on the ability of the restoration program (or operator) in finding a similar network state and in applying the corresponding plan. However, if a new contingency occurs, a new plan (*on-line*) has to be generated. In the simplest case, this can be achieved by shooting down some part of the network and bringing it back to a known state, for which a plan exists. The other solution, certainly better, starts directly from the actual network state.

Two main classes of algorithms have been used to solve this problem: deterministic and stochastic ones. One of the first efforts for treatment of the PSRP with deterministic techniques was the work of (Sakaguchi and Matsumoto, 1983), who had created a Knowledge Based System (KBS) based on the knowledge of the power system operator. This was followed by other works using an expert system approach, such as (Komai *et al*, 1988) and Kojima *et al* (1989). As the size and complexity of electrical power systems surpassed the capacity of human control and therefore of rules based on human knowledge, other methods were developed, such as (Aoki *et al*, 1987) using the classical integer programming approach, (Huang *et al*, 1991) using optimization techniques (Nagata *et al*, 1995) using a

hybrid system of rules and mathematical programming. For a review on these methods, see (Curcic *et al*, 1997). Even for these methods, the determination of hundreds of discrete variables in time is a complex task. Therefore, stochastic methods combined with power flow simulation tools became an interesting alternative to the deterministic approaches (Matos et al, 2004).

Stochastic approaches such as Neural Networks (Hsu and Huang, 1995) (Bretas, 2003) and Genetic Algorithms (Bretas, 1998) (Luan *et al*, 2002) are relatively new and have received some criticisms such as that they typically use a fixed length string to represent the solution plan, high computing times and the low confidence of the power system operator on the generated plans (Susheela, 2000). In order to solve these problems, we propose a variable length solution representation (chromosome) with new mutation and crossover operators. In relation to the computing time, first, only feasible solutions are generated and second, our crossover operator maintains the feasibility of the plan. This spares time because the computation of the power flow along the plan represented in the chromosome does not end in an unfeasible solution and therefore must not be immediately discarded. A plan in the population may have a low score but is rarely discarded because of unfeasibility. The crossover cut occurs only at network states in each plan submitted to the crossover operator.

The third criticism, the confidence of the power system operator on the obtained solution, can be partially solved by exhaustively testing and by carrying out demonstrations on previous cases with the planner and power flow simulator. However, we propose in this work that sound solutions may be obtained if this aspect is considered in the fitness function already, for instance, by requiring a good quality solution along the plan and not just at the last stage of the restoration. Finally, the increase in computer performance will certainly turn those methods now just used for the composition of a restoration plan into *on-line* solutions with a response time from seconds to minutes.

This paper is organized as follows: after the introduction, section 2 presents the restoration system and its components. Section 3 shows the Genetic Algorithm component with a description of the GA operators and section 4 presents results for three different power systems, the IEEE 14-Bus system and IEEE 30-Bus (Freris and Sasson, 1968) and finally the CEEE 146-Bus. Section 5 presents our conclusions.

2 THE RESTORATION SYSTEM

The main components of the restoration system are shown in Figure 1. The GA component obtains information about the network topology by reading the Electrical Network Topology file. This file describes the buses with their connected generators and loads as well the transmission lines among buses. As an illustration, the upper part of Figure 3 shows a schematic view of the IEEE 14-Bus topology file. The topology and the actual network state are used to create an initial population of feasible solutions. The actual state of the electrical network is defined by the actual pattern of connections and activity/inactivity of buses and must be read from the real network through sensors and state observers.

Then, during a GA generation, each chromosome (restoration plan) is used to drive the whole power system through a series of states, from the actual state to an end state, hopefully with all loads restored. Further, the intermediate states of the network must obey the component limits and available power for the loads. Each such state is computed by the Power Flow Program (PFP), whose results are turn back to the GA. Then, the fitness function is used to evaluate the restoration plan.

The PFP is called many times for the evaluation of a chromosome. If the plan in the chromosome is composed of N stages, then the PFP is called N times. For example, if a GA needs a population of size 25 and 25 generations to find a good plan, then the PFP will be executed $25 \times 25 \times N$ times. Our work spares this time by generating and processing feasible solutions and leaving a small margin for unfeasible solutions only in the case of mutated chromosomes, and only for intermediate solutions (before the last generation).

The PFP implements the classical fast decoupled Power Flow method of Stott and (Alsaç, 1974) and was developed by (Canto dos Santos *et al*, 2006) In that work, another approach to the .problem, based on Linear Programming, was presented.

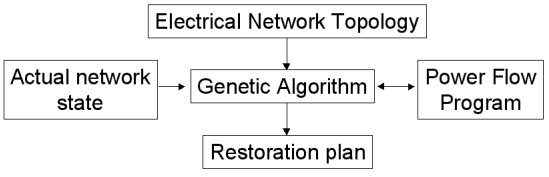


Figure 1: Restoration system components.

3 GENETIC ALGORITHM COMPONENT

The design of the chromosome format is crucial in quickly finding an optimal solution. First, the mutation operator must have a small impact on the chromosome performance (big random jumps should be avoided) and second, the crossover operator should transfer whole functional blocks avoiding any rough breaking of these functional blocks. In this work, the time axis is mapped on the linear chromosome structure.

3.1 Chromosome Representation

Different of the classical GA approach (Goldberg, 1989), the new chromosome representation has a variable size, consisting of a sequence of stages, each containing a group of commands sent to the electrical power system. As the commands are sent, lines and buses change their state. As will be shown in the next section, these states are used to define the crossover points.

For this new proposed representation, each chromosome codes a sequence of operations or stages and has a variable size. Not only the whole sequence has a variable size, but also each stage is composed of a variable number of operations, which are executed simultaneously during the restoration process. Further, the different chromosome sizes and the requirement for feasibility of the restoration plans demanded a new crossover operator different from the classical ones.

Figure 2 shows two different illustrative representations of the chromosomes. The stages are shown as discrete elements (white boxes) inside each chromosome and they contain a group of operations. Each operation in turn is represented by a pair of identifiers using the format "<address>: <operation>". The operations are divided into two groups: operations on lines and operations on buses. The <address> identifies the bus or line on which the <operation> will be executed. If the operation is executed on a bus *n*, its <address> is expressed as B*n*. If it is executed on a transmission line *n*, its <address> is L*n*. The other identifier represents the operation. Operations on transmission lines include the connection of one of its end points (the origin point or operation O+ and destiny point or operation D+). Operations on buses may be a load connection to the bus (operation L+), connection of the power supply (operation Ger) and the connection of synchronous compensators to the bus (operation C+). The system was designed so as to enable simple additions of other operations as well new equipment types.

For instance, the second stage in Figure 2 represents the connection of the origins of transmission line L1 and L2 at the same time.

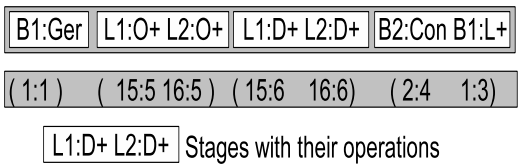


Figure 2: Chromosome representation as a vector of stages with discrete operations. The operations are indicated by mnemonic and numeric codes. The first line shows a chromosome in a mnemonic format and the second line, in a numeric format.

3.2 Genome Initialization

The highly flexible representation of the chromosome demanded the inclusion of specific program modules for the generation of feasible chromosome initializations.

In order to obtain feasible initializations, first a graph is generated in which buses are represented by nodes and transmission lines by edges. Also, the buses associated with black start generators (capable of restarting with no external power source) are at the root of the graph and are connected with other buses in a tree-like way. The graph represents the topology of the power system.

Then, a Random Search (RS) is carried out on the graph from the root nodes where components attached to buses are randomly selected. For each selected component an operation is chosen. Since many components may be chosen at the same stage, not only an operation is created, but a whole operation set may be created and added to the current chromosome. This way, different operation sequences are placed in the initial population.

An example of the genome initialization process is shown in Figure 3 for the IEEE 14-Bus power system topology composed of 14 buses, 18 transmission lines, 2 generators and 11 loads (arrows). In this example, the objective is to find a sequence of operation sets, which should lead the power system from an initial state characterized by total blackout to a final state with all loads supplied. The nodes in the topology graph represent the buses and the links the transmission lines. In the real operation, these data are obtained from a supervisory system. In the first step (topology graph (1)), only the bus 1 (node 1) is activated, fed by the generator

(black start capable generator) connected to the bus. Then (graph (2)), the algorithm computes which are the nodes linked to already activated nodes such as in a Random Search. From the nodes allowed to be activated, the algorithm chooses randomly if they shall be activated or not generating different individuals. In the next steps, the algorithm proceeds considering all the nodes directly linked to the already activated nodes and determines if and which ones shall be activated. Note that the algorithm may activate more than one bus at the same time. For instance, from graph (4) to graph (5), both buses 3 and 4 (nodes 3 and 4) were activated simultaneously.

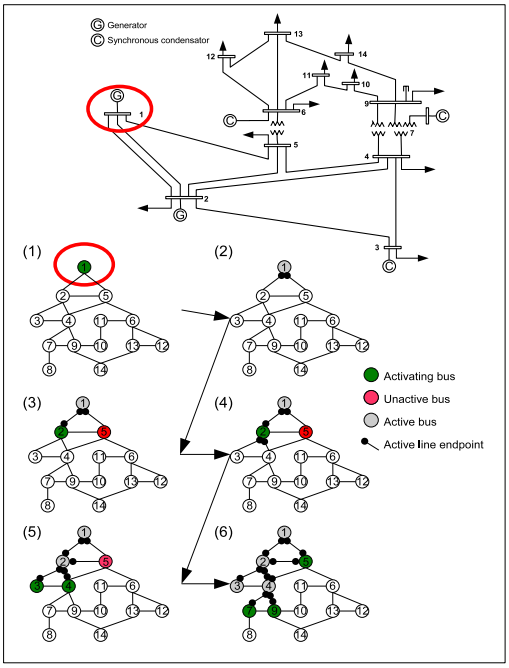


Figure 3: Demonstration of the genome initialization process for the IEEE 14-Bus system. The graph is searched and the corresponding buses are randomly activated. At each step, inactive buses have a renewed opportunity to be activated by the algorithm. Buses are indicated by numbers, transmission lines by solid lines and loads by arrows.

The resulting activations during these steps are accordingly indexed in the genome in the form of operations, and the initialization process continues until all the nodes are connected to active buses and the number of inactive nodes reaches a previously chosen random limit. Therefore, different solutions are generated, all of them considering possible pathways described in the topology graph. For instance, there is no sense in connecting bus 14 before bus 9 or bus 13 is activated. The randomness of the bus activation creates chromosomes with diverse solution strategies ensuring a good coverage of the solution space.

3.3 Crossover Operator

Due to the more flexible representation of the chromosome, a new crossover operator was developed, which avoids improper operations, such as the connection of a load or generator (not the black start capable generator) to a bus without power.

The classical GA crossover cuts two chromosomes at the same position because there is an exact correspondence between both left and right sides of the chromosomes. However, the flexibility of the operation positioning inside the variable size chromosome demands the computation of synchronization states in the parents' chromosomes. Each stage in a chromosome is associated with a network state, which is used by the crossover operator. This state is the result of all operations of all stages to the left side (before) of the respective chromosome stage and of the operations in the stage.

Lines may be in one of four states:

- not connected;
- origin point connected;
- end point connected;
- totally connected.

Each end of a transmission line was modeled by a different connection for two reasons: first, this is more realistic, resulting in solutions that can be readily understood by the system operator and second, if the line model had included just one connection, then the activation of the line would automatically activate both connected buses at the same time.

Figure 4 shows an example how a line is activated, depending on the sequence of bus activation and end connections.

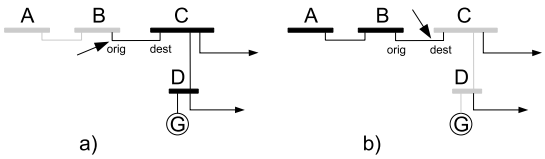


Figure 4: Example of how a line may be activated from different sides. In case a) the line is activated by the bus B if it is activated and the "orig" end of the line is connected. In the case b), the line is activated if the bus C is activated and the "dest" end is connected.

For the buses, the allowed states are:

- inactive bus (disconnected);
- bus activated - when the bus has power from a transmission line or generator;
- generator connected;
- load connected;
- synchronous compensator connected.

The states of the buses in both chromosomes are compared, and if the network state is the same, they are equivalent.

All the equivalent state pairs are stored in a list, and a position in the list is randomly chosen, which represents the stage at which the two sequences will be cut and where the crossover occurs.

A consequence of the use of this operator is that the resulting sequences (children) have a different number of operations reaching the same state, which means that some of them will be smaller and, possibly, more efficient than the others.

This process is illustrated in Figure 5. In this case, a new chromosome generated by crossover of the left side of the upper chromosome and the right side of the lower chromosome may be better than both parent chromosomes in terms of number of stages. The left side of the upper chromosome takes a smaller number of stages to reach the same state than the lower chromosome and it will be transmitted to the children chromosome.

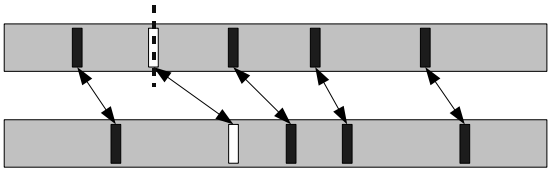


Figure 5: Determination of equivalent stages in two chromosomes of the genome. Different pairs of equivalent stages are stored in a list and one of them (the white one, for instance) is randomly chosen. Dotted lines show the stages in the chromosomes where the two sequences are divided for the crossover operation.

3.4 Mutation Operator

There are three types of mutations: inclusion and exclusion of an operation in one stage and permutation of operations in two stages. As an illustration of the steps performed by the mutation operator, see Figure 6. Because these are unrestricted operations on the chromosome, these may generate unfeasible solutions with respect to the delivered and consumed power. Unfeasible solutions are tolerated only at intermediate generations of the GA and intermediate stages of the chromosome because this increases the diversity of solutions and helps the GA to escape local minima. For instance, a certain load requiring more power than available may be connected during intermediate generations. However, only feasible plans are considered in the end generation.

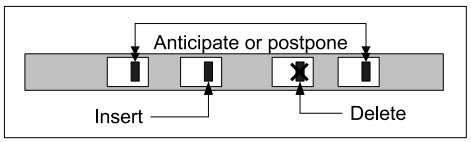


Figure 6: Representation of possible mutations. The white rectangles represent stages and the black ones, possible operations in the stages. The operations can be included, excluded or transferred from one stage to another by the mutation operator.

3.5 Fitness Function

In order to compute the fitness function, the operations described in the chromosome are used by the PFP to compute the power flow at each stage.

In this work, the flexibility of the fitness function was maintained by the use of parameters (*A*, *B* and *C* in the following equations) allowing a detailed analysis of the influence of each system variable on the solution obtained by the GA. The fitness function *FF* is defined by:

$$FF = \frac{\overline{L_{At}}(L_{At_N} - A \cdot \overline{Fl_{At}}) + \overline{L_{Re}}(L_{Re_N} - B \cdot \overline{Fl_{Re}})}{1 + C \cdot N}$$

$$\overline{L_{At}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} L_{At_{ij}}$$

$$\overline{L_{Re}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} L_{Re_{ij}}$$

$$\overline{Fl_{At}} = \frac{1}{N} \sum_{i=1}^{N} R(\sum_{j=1}^{M} L_{At_{ij}} - \sum_{j=1}^{M} Ge_{At_j})$$

$$\overline{Fl_{Re}} = \frac{1}{N} \sum_{i=1}^{N} R(\sum_{j=1}^{M} L_{Re_{ij}} - \sum_{j=1}^{M} Ge_{Re_j})$$

$$R(x) = \begin{cases} 0, & if x < 0 \\ x, & \text{otherwise} \end{cases}$$

where *M* is the number of buses in the power system, *N* is the number of stages, L_{At} and L_{Re} are the active and reactive loads respectively, Ge_{At} and Ge_{Re} are the generated active and reactive power respectively, and *A*, *B* and *C* are trimming constants. *i* and *j* denote the stage and bus respectively and *R(x)* is the ramp function.

The main component of the fitness function is the supplied active and reactive loads at the last stage N, L_{At_N} and L_{Re_N}. However, tests with just these variables presented solutions which were not balanced along the stages and are difficult to be accepted by the system operator. The inclusion of the mean of the loads along the stages intends to meet, at each stage, the highest possible number of active and reactive loads, a good distribution of supplied loads along the stages and a sound solution to the system operator.

Parameters A and B decrease the fitness function of individuals in case the generated power is lower than the total load of the system and therefore help to discard any unfeasible solution generated during the GA runs by the mutation operator. The inclusion of the supply power failure in the fitness function intends to obtain solutions whose power generation is sufficient to attend all the connected loads. The parameters A and B control how tolerant is the GA to unfeasible solutions during runs. This allows suboptimal solutions to remain in the population and thus contributing to maintain the diversity of the population. Both parameters may be increased in the last few generations leaving only feasible solutions in the population.

Since the quick evaluation of the PFP is the main concern for a practical use of the GA, the PFP needs only to compute the state of the connected part of the network. Then, as the restoration program goes through each stage in the chromosome, only the buses and lines necessary to the execution of the current stage are transferred to the PFP, reducing the number of nodes and edges the PFP has to consider during the calculations. As the electrical network gets more and more connected with each new stage, its size increases and the computations in the PFP become harder to carry out.

Other fitness functions were designed. However, the previous fitness function resulted in the best solution plans, with a quick attendance of the needed power while maintaining the power generation restrictions.

4 RESULTS

In this section, results of experiments performed with three different power systems, the IEEE 14-Bus system, IEEE 30-Bus and finally the CEEE 146-Bus, three power systems with increasing degree of complexity are shown. Each experiment used a different population initialization and all electrical parameters of the systems were monitored, such as the active and reactive powers of the best genome in each generation. After running each experiment, the best genome in the population is shown in a graphic representation. As each stage corresponds to a group of operations executed on the power system and the execution of each such operation spends time, a correspondence can be made between the consecutive execution of those stages and the restoration elapsed time. It will be assumed that each stage takes one unit of time to be executed.

It was considered the occurrence of a general blackout in the IEEE 14-Bus (Freris and Sasson, 1968) system and that only the generator at bus 1 is black start able. Also, all the 20 branches are available for the restoration.

In Figure 7, the evolution of the GA with three different initializations is presented. Typically, the GA starts with 40-50 stages (dashed lines) and a fitness function of 0.75 (solid lines) of the maximum value, showing that many solutions generated by our method have a high fitness function (and feasible) at the GA initialization already. As the GA computes new solutions, the fitness function increases and the number of stages needed to reach it decreases.

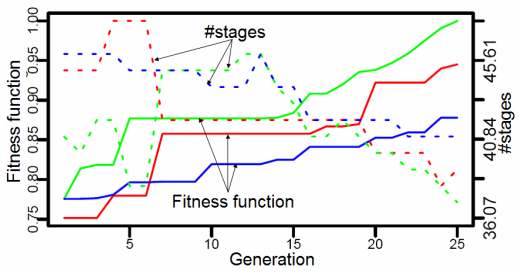


Figure 7: Three GA runs for the IEEE 14-Bus System, with a population size=25 chromosomes and fitness function FF with parameters A=1, B=5 and C=1. Solid lines represent the value of the normalized fitness function for the three runs (scale at the left side of the graph) while dashed lines represent the number of stages (scale at the right side of the graph) of the best solution in the population.

Figure 8 shows the corresponding plan obtained. The generated and delivered power of the best individual in the population is shown along the restoration stages. Results for the power system IEEE 30-Bus using the same GA parameters are shown in Figure 9. It can be observed that the GA obtained an efficient solution in terms of used power. For instance, once the difference between generated and consumed reactive power reaches zero, there is an increase in the power generated so as to supply the new loads. Afterwards, the loads are connected to the system until that margin reaches

zero again, and so on. Further, as expected, the plan for a larger power system such as the IEEE 30-Bus demands more stages, for instance, 37 for the IEEE 14-Bus and 78 for the IEEE 30-Bus .

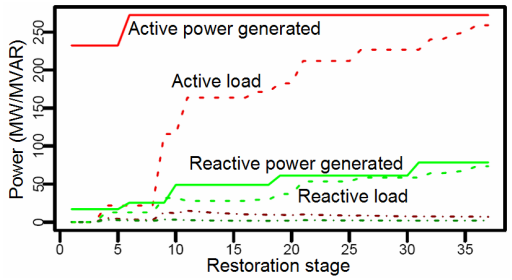


Figure 8: Restoration of the IEEE 14-Bus. Solid lines represent the active and reactive power generated while the dashed ones represent the power consumed or the total loads.

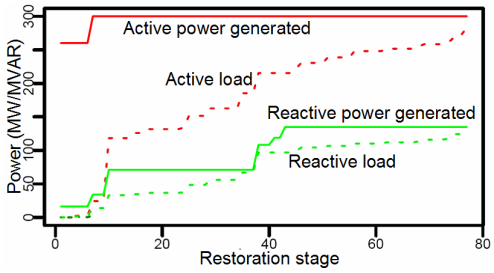


Figure 9: Restoration of the IEEE 30-Bus power system. Solid lines represent the active and reactive power generated while the dashed ones represent total loads.

Finally, we tested the GA algorithm with the CEEE (a real life Brazilian system) power system composed of 146 buses and 196 transmission lines in south Brazil. Figure 10 shows one of the obtained plans. Approximately 130 stages are needed to restore the full system from a complete blackout.

Certainly, the set of experiments presented in this section represents only part of all test performed with our approach. Other tests varying the number of generations, population size, fitness function and parameters were performed too.

5 CONCLUSIONS

The objective of this work was to show a restoration system based on a GA that presents efficient plans to the power system operator when a blackout or any other serious lacks occurs. The use of GA for the generation of operation sequences, or strategies is only possible with the careful codification of the operation sequences in the chromosome taking into account the electrical power network topology. If the task of determining the feasibility of the solutions is entirely left to the PFP, then the GA generates many unfeasible solutions with a direct impact on the performance of the GA. Also, there must be a limit on the number of allowed generations if the system is to be used in an *on-line* modus. For example, in this work, this number was set to 25 generations. All this enable the GA to quickly find a solution without wasting of time computing the power flow for unfeasible individuals.

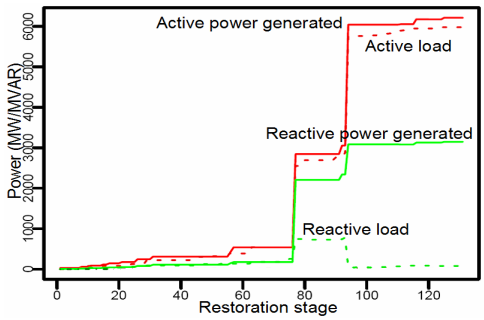


Figure 10: Power system restoration plan for the CEEE System, obtained with a GA with a population size=25, number of generations=25, and parameters A=1, B=5 e C=1.

The choice of the fitness function is important to define the quality of the obtained solution. In this work, a parameterized function was chosen as a good compromise among user priorities, flexibility and complexity of the function. Also, a good distribution of connection operations along the plan represents a sound solution for the operator. In this respect, we observe that there are no works considering the quality of operation stages along the plan in terms of soundness for the system operator, as proposed in this article. Certainly, this is one of the causes for the low confidence level of system operators on solutions obtained by stochastic algorithms.

However, some care must be kept in relation to a solution found by stochastic algorithms. A GA may find a good solution in terms of fitness function but it may be unfeasible. The stochastic freedom of GA solutions may be kept at the intermediate generations with the risk of reaching the last generation with unfeasible plans in the whole population. For example, if the best individual is good but unfeasible, its genes may propagate along the population making all of them unfeasible. This is called *lost of diversity* in the GA area. Therefore, there must be a compromise between keeping all the time all the solutions feasible (for example, making

the parameters *A* and *B* very high) and maybe loosing a good solution, and leaving the GA completely free to generate and process any solution plan. We think that in this respect we could attain a good compromise by allowing only the mutation operator to generate unfeasible solutions at intermediate generations and also by considering only feasible end solutions in the last generation.

Finally, researches in progress point out to the need of inclusion of more complex and realistic simulations, such as the dynamical behavior of the electrical power system. For example, the simulation of voltage and current overshooting during switching operations, which can cause the breaking of lines by limiting current and voltage devices, could preview an instable system reaction leading the power system to a complete blackout. Since this analysis increases the processing time of the PFP, any such addition makes the feasibility analysis presented in this work even more important. Currently, the use of other metaheuristics, specifically Tabu Search, is being studied and will be published in the future.

REFERENCES

Aoki, K., Kuwabara, H., Satoh, T., Kanezashi, M. (1987) Outage state optimal load allocation by automatic sectionalizing switches operation in distribution systems. *IEEE Trans. Power Delivery* PWRD-2: pp.1177-1185.

Bretas, N., Delben, A. and Carvalho, A. (1998) Optimal energy restoration for general distribution systems by genetic algorithm. In *Power System Technology, 1998 International Conference on*, volume 1, pages 43–47.

Bretas, A. and Phadke, G. (2003) Artificial neural networks in Power System Restoration. *IEEE Transactions on Power Delivery*, 18(4):1181–1186 .

Canto dos Santos, J., Gómez, A., Rodrigues, A. (2006) An optimization algorithm to improve security of electrical energy systems. In ICINCO 2006 - 3rd Int. Conf. on Informatics in Control, Automation & Robotics. Setubal - Portugal. INSTICC Press.

Curcic, S., Ozveren, C.S., Lo, K.L. (1997) Computer-based strategy for the restoration problem in electric power distribution systems. *Inst. Electr. Eng. Proc.* 144: 389-398.

Freris, L., Sasson, A. (1968) Investigation of the Load Flow Problem, Proceedings of IEE, 115(10): 1459-1470.

Goldberg, David E. (1989) Genetic Algorithms in search, optimization, and machine learning**.** Addison Wesley Longman Inc.

Hsu, Y. and Huang, H. (1995) Distribution systems service restoration using the artificial neural network approach and pattern recognition method. In *IEE Proceeding on Generation, Transmission and Distribution*, volume 142, pages 251–256.

Huang, J.. Galiana, F. and Vuong, G. (1991) Power system restoration incorporating interactive graphics and optimization. In *Power Industry Computer Application Conference, 1991. Conference Proceedings*, pages 216–222, Baltimore.

Komai, K., Matsumoto, K. and Sakaguchi, T. (1988) Analysis and evaluation of expert's knowledge for Power System Restoration by mathematical programming method. *IEEE International Symposium on Circuits and Systems*, 2(1):1895–1898.

Kojima, Y., Warashina, Nakamura, S. and K. Matsumoto, K. (1989) The development of power system restoration method for a bulk power system by applying knowledge engineering techniques. *IEEE Transactions on Power Systems*, 4(2):1228–1235.

Luan, W., Irving, M. and Daniel, J. (2002) Genetic algorithm for supply restoration and optimal load shedding in power system distribution networks. In *IEE Proceeding on Generation, Transmission and Distribution*, volume 149, pages 145–151.

Matos, M., Ponce de Leão, M., Saraiva, T., Fidalgo, J.,
Miranda, V., Lopes, J., Ferreira, R., Pereira, J., Proença, M., Pinto, J., (2004). *Metaheuristics applied to power systems* in *Metaheuristics: computer decision-making*, Kluwer Academic Publishers, Norwell, MA, USA

Nagata, T., Sasaki, H. and Yokoyama, R. (1995) Power System Restoration by joint usage of expert system and mathematical programming approach. *IEEE Transactions on Power Systems*, 10(3):1473–1479.

Sakaguchi, T. and Matsumoto, K. (1983) Development of a knowledge based system for Power System Restoration. *IEEE Transactions on Power Apparatus and Systems*, PAS-102(2):320–329.

Stott, B., Alsac, O. (1974) Fast Decoupled Load Flow. *IEEE Transactions Power Apparatus and Systems*, Vol 93, May / June 1974, pp. 859-869.

Susheela, D.V. and Murty M.N. (2000) Stochastic search techniques for post-fault restoration of electrical distribution systems. *Sadhana*, vol. 25, Part 1, February 2000, pp. 45-56.

Wu, F. and Monticelli, (1988) A. Analytical tools for Power System Restoration - conceptual design. *IEEE Transactions on Power Systems*, 3(1):10–16.

COGNITIVE TECHNICAL SYSTEMS IN A PRODUCTION ENVIRONMENT

Outline of a Possible Approach

Eckart Hauck, Arno Gramatke and Klaus Henning
Centre for Learning and Knowledge Management and Department of Computer Science in Mechanical Engineering
RWTH Aachen University, Dennewartstr. 27, Aachen, Germany
{hauck, gramatke, henning}@zlw-ima.rwth-aachen.de

Keywords: Cognitive technical system, cognitive architectures, production, knowledge representation.

Abstract: High-Wage countries face the dilemmas of value- vs. planning orientation and the dilemma of economies of scale vs. economies of scope summed up in the term polylemma. To reduce the dilemma of planning vs. value orientation cognitive technical systems seem to be a promising approach. In this paper the requirements of such a cognitive system in a production environment is presented. Furthermore a first concept of a software architecture is given. To implement a knowledge base for a cognitive technical system certain formalism were scrutinized for their suitability in this approach and a possible use case for such a cognitive technical system is presented.

1 INTRODUCTION

Today's production industry in high-wage countries is confronted with two dichotomies value orientation vs. planning orientation as well as economies of scale vs. economies of scope. In the last years, production in low-wage countries became popular with many companies by reason of low production costs. To slow down the development of shifting production to low-wage countries, new concepts for the production in high-wage countries have to be created.

The question of developing these concepts is connected to the polylemma of production, shown in Figure 1, which summarizes the two dilemmas mentioned above. Production systems of the future have to accomplish the apparent incompatibility of the two dichotomies. To improve the competitiveness compared to production in low-cost countries, it is not sufficient for production in high-wage countries to achieve a better position within one of the dichotomies, it will have to resolve the polylemma of production (Schuh, 2007). The research questions of the Cluster of Excellence "Integrative Production Technology for High-Wage Countries" aims at the resolution of this polylemma.

A reduction of the polylemma would widen the operational range of production systems over the batch size resulting in a smoothed unit cost curve as shown in Figure 2.

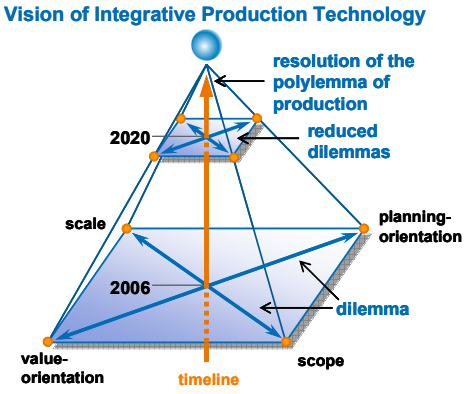


Figure 1: Polylemma of production.

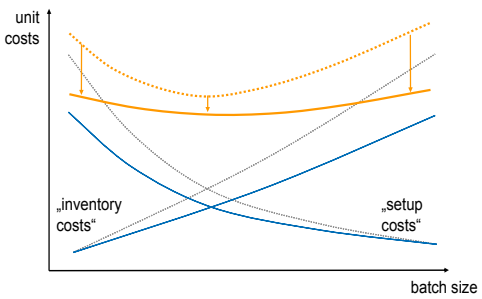


Figure 2: Unit costs above batch size.

Our approach to reduce the dilemma of value orientation vs. planning orientation leads towards an intelligent manufacturing environment realised by the use of artificial cognition. A cognitive architecture is one of the central parts for an intelligent production system to become reality.

Using cognitive mechanisms like learning, planning and problem solving in connection with interaction with a real environment is not a new consideration anymore. Over the years, cognitive science has become an important part of research in psychology as well as in robotics. It is our ambition to study the different theories of cognitive architectures and finally to conceptualize a control unit suitable for a production system. Before we will present a possible concept of a cognitive control unit, we will give a brief overview of the state of the art of cogntion in technical systems and then focus on the requirements and an architecture of a cognitive system for the production industry.

2 COGNITVE TECHNICAL SYSTEMS

Cognition is defined as the acquisition, storage, transformation and usage of knowledge (Matlin, 2005). A cognitive system could imply following mental processes:

- Perception and action
- Learning
- Problem solving
- Reasoning
- Decision making

To create technical systems using some of the mentioned abilities, cognition started to play a major role in more and more fields of technical environments in the last years. Many variations of (partial) autonomous systems have been developed (Putzer, 2004), e.g. service robots or robots for sports competitions which mostly strive to copy human behaviour. Also in numerous areas of our daily life intelligent technical systems become more and more common, applications like driving assistance systems (Heide, 2006) or assistance robots in the kitchen (Burghart, 2005) try to improve the daily routines of our society in the future.

In today's production industry cognition is beginning to enter the fields of sophisticated production systems, which so far are mostly automated systems. A disadvantage of these systems is the lack of flexibility. Changing the characteristic of a product leads to a great effort to reprogram whole process steps or even requires a partial change of the used modules. For a large process chain these changes can be the most cost intensive part and could cause a loss of efficiency. In conjunction to our research to conceptualize a cognitive unit for a production environment, an associated research group of the Cluster of Excellence is focussing on technology enablers for embedded cognition. These enablers should also be capable of self-optimisation.

2.1 Requirements of a Cognitive Technical System

A technical system including cognitive abilities could possibly circumvent the aforementioned problematic situation in current automation. To be suitable for a production environment such a system has to meet at least the following requirements. First multimodal Interaction with the environment and with human controllers should be possible. Also Information processing (mental processes) in addition with the availability of planning and coordination modules is required. To ensure a flawless interaction with human controllers transparent machine behaviour is essential.

A cognitive technical system must be able to perceive and to influence its environment, which is realised through a perceptional and an actoric component. Figure 3 shows the different communication levels acting upon a cognitive control unit. Aside from the communication with a human operator it has to interact with other production systems from shop-floor level to whole production networks. To ensure a flawless information and knowledge flow a well-balanced multimodal interaction between operator and machine is indispensable. This is also especially relevant for providing embedded training (Nolden, 1999) of human operators on these systems which leads to technological and methodological competence of a joint cognitive system of human and machine.

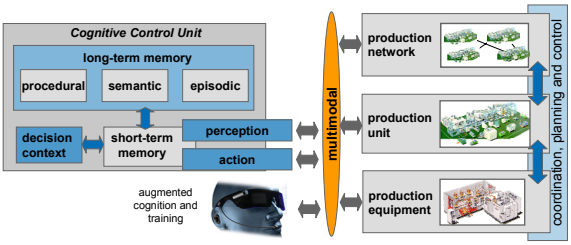


Figure 3: Multimodal interaction of a cognitive technical system.

In addition to that the gained information has to be processed – comprehending knowledge storage, learning and problem solving. This requires an explicit knowledge representation within the system and the possibility to reason about the given problems. The knowledge has to be stored in an inferable way that deterministic algorithms can be used to find possible ways through the problem space to the desired goal.

For more complex processes the system has to arrange the different tasks in a useful combination to accomplish the job. This requires a sophisticated planning module, which is one of our research focuses. A coordination module is responsible for the implementation of the scheduled tasks with the action module.

Transparency of machine behaviour to a human user will be one of the crucial aspects of the cognitive technical system. The system itself and human operators should be able to comprehend the decisions the technical system takes and the subsequent actions it executes. That is necessary to prevent handling errors by the human operator and increases the chance to discover and correct malfunctions. Also the mental models of the operator and the technical cognitive systems have to be compatible. This leads to an increasing acceptance of the system by the human operator (Hartmann, 1995).

2.2 Cognitive Architectures

A possible approach to fulfil the discussed requirements is the use of a cognitive architecture. In 1987 Newell defined the Unified Theory of Cognition (UTC) (Newell, 1990). An approach conforming to the UTC has to be composed of a set of mechanisms which accounts for all forms (processes) of cognition. In robotics and cognitive science research aimed for developing architectures sufficient to the UTC. Two popular representatives are ACT-R (Adaptive Control of Thought—Rational; Anderson, 2004) and Soar (originally SOAR - State, Operator And Result; Laird, 2006). Soar and ACT-R are both rule-based and goal-oriented architectures, which can be used for creating artificial intelligence.

The structure of Soar is characterized by different models of the memory (Figure 4). Production Rules entered by a user are included in the long-term memory. With the help of the perception module, the actual state of the environment is modelled in the working memory. Depending on this state and the preference memory Soar elaborates and fires within a decision cycle dertermined production rules and modifies entries in the short-term memory.

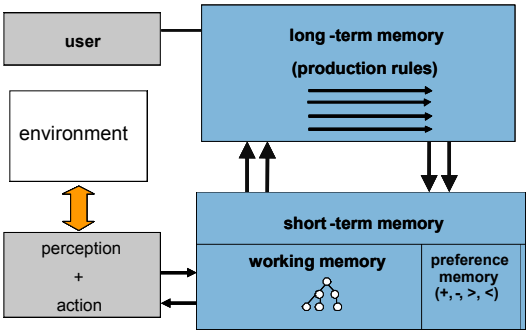


Figure 4: Structure of the memory in Soar.

The basic architecture of ACT-R consists of a set of modules for processing different forms of information (Anderson, 2004). In comparison to Soar, ACT-R differs between declarative and procedural knowledge. The basic idea of modelling cognitive abilities like learning and problem solving by using production rules is similar to the Soar architecture. However, the functional aspects of the different modules are deduced from psychological theories. Since ACT-R is a theory focussing on modelling human cognition, it also simulates inefficient human behaviour which is not consistent with industrial applications. Thus, there is only a minor presence of the theory in the field of robotics or automation so far.

Unlike more specialized approaches in cognitive science and robotics, Soar and ACT-R provide a generic concept for developing artificial intelligence. We studied the two architectures in the face of their pros and cons for their use in production environments by examining criterias like persistence, expandability and autonomy.

Soar is a suitable approach for modelling cognitive systems for production environments. It provides a wide field of the required capabilities like learning, planning and problem solving within a complex production rule system. However, our research so far has shown that not all components of Soar are adequate for a production environment. Due to complexity of the application area, the real time capability of Soar-architectures decreases with the higher amount of knowledge stored in the procedural memory, provoked by an increase in possible matches for the reasoning algorithm (Doorenbos, 1995). This is a common problem of deliberative rule-based architectures. Architectures which are used for mobile robots claiming improved

real time capability often don't include an explicit representation of a knowledge base. The main problem of these systems is that it cannot be ascertained that the implicit knowledge is sufficient for a given problem.

A hybrid approach to this problem seems to be the most feasible. A possible software architecture for such an approach is presented in the next section. The above mentioned requirements for a cognitive system like problem solving and planning is satisfied by Soar. In addition to that the interaction between a cognitive unit with other systems or human workers has to be researched.

2.3 The Cognitive Control Unit (CCU)

The challenge of developing a system sufficient to the aforementioned requirements lies in combining dynamical system adaption to mutable goals with a real-time capability regarding operations in the production environment. Concerning our chosen handling operation, which will be presented in the following, we developed a first concept of a cognitive control unit (CCU) usable in the field of production technology.

The CCU (see Figure 5) underlies a concept from a planning level down to the operating level represented through components for perception and action. This concept is derived from the multi-level approach for cognitive technical systems proposed by Paetzold (Paetzold, 2006) The modules for planning and coordination take over the aspired cognitive abilities. Due to the symbolic representation from objects in Soar the perception module has to connect the perceived information to a symbol in the knowledge base to allow proper reasoning in the planning module. The coordination module is responsible for the correct execution of the planned tasks on the hardware level. The actual execution of the tasks is then done via the action module. This allows a separation of the deliberative and reactive parts of the system and ensures a real time capability. Sensor information which needs an immediate response won't reach the deliberative level of the system and will be executed without reasoning. This is important for the safety of human operators, who work in a human machine cooperation.

To control external modules, the CCU has a generic interface which enables a direct communication on machine level. Additionally knowledge engineering processed by multimodal human machine interaction should be possible.

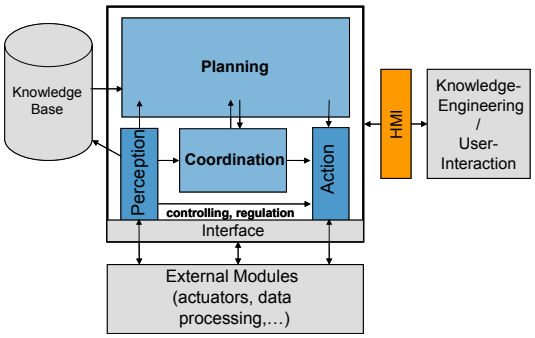


Figure 5: Possible concept of a CCU.

The cognitive mechanisms on the planning and organisational level combined with additional modules like a Human-Machine Interface and an external knowledge base will complete our approach to a multi-level architecture used in today's robotics. Due to the important role of internal and external knowledge bases of the aspired system different formalisms of knowledge representation have to be evaluated.

3 KNOWLEDGE BASE

Besides an internal representation of knowledge within the cognitive architecture, e.g. production rules in Soar, it is a subgoal to develop an external knowledge base, which should contain an explicit declaration of knowledge. Generally a representation of knowledge has to accomplish at least the ability to recover the stored knowledge. Furthermore the formalism of knowledge should enable the system to process the stored data (Haun, 2000). Several formalisms of knowledge representations for different purposes are available. Within our project we try to determine which formalism(s) are suitable for the knowledge base of the CCU and could improve the associated cognitive abilities. It has to be researched which additional data has to be stored in the external knowledge base. A knowledge base for our purpose could contain next to a representation of the real environment data for internal computations as well as episodic knowledge which memorizes all past events.
Formalisms of knowledge representations ranging from declarative to procedural forms are:

- Semantic Nets
- First Order Logic
- Frames
- Production Rules
- Object-Oriented-Representations

An explicit representation model of the environment of the production unit could be summarized by the term ontology – which we will refer to as an explicit specification of a conceptualization (Gruber, 1993). In the last years, the use of ontologies became popular and got more important in computer science and artificial intelligence. The possibility of reasoning makes an ontology to an adequate modeling structure of representing knowledge. Not all of the aforementioned formalisms are suitable for an ontological representation of the relevant environment. Also it has to be evaluated which amount and level of detail of knowledge is essential for a proper description. To generate new knowledge and possibly new production rules, the formalisms have to provide the ability of inference.

To realise reasoning it is essential that the knowledge has a semantic structure. Consistency and completeness are also requirements for the process of reasoning. An Ontology defined in OWL (Web Ontology Language) (Smith, 2004) could fulfil our demands of a knowledge base. Because of OWL-DL features computational completeness and decidability, such an ontology would be suitable for a model of the real environment as well as other inferable semantic connected data.

The translation of the knowledge representation form (Figure 6) between the knowledge base and the cognitive unit has also to be realised. The CCU should be capable to generate production rules out of the external knowledge and to extend the external knowledge base with collected and elaborated data within the working memory. For this operation a compiler for both directions is required. This compiler should be able to translate complex and big sized ontologies but also be generic enough to be adaptable to other representation formalisms.

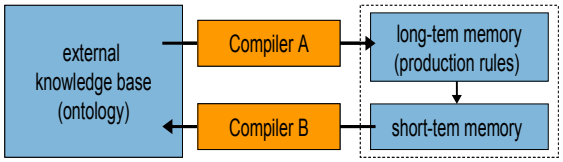


Figure 6: Knowledge Translator.

4 HANDLING OPERATION (USE CASE)

As mentioned before we want to realise a handling operation with a cognitive technical system. The focus of this use case is the realisation of an intelligent grasp behaviour by cognitive means. What actually needs to be described is driven by the process itself which means that here one has to care about center of gravity, material and surface attributes but not for the inner structural composition. Figure 7 shows the layout of the test-setup. The aim is the assembly of a pile out of different coloured bricks. This involves the identification and position of a needed part, the picking operation and the transfer from the belt conveyor to the assembly area. In the assembly area the cognitive control unit has to choose the right grasp strategy depending on the current state of the to be assembled parts. To do this the cognitive unit needs multi sensorical input. For the identification of colour and position an image recognition is required. The transfer operation will be realised by integrating already known collision-free transfer moves whereas the fine movements for the gripper have to be planned cognitively by the system itself.

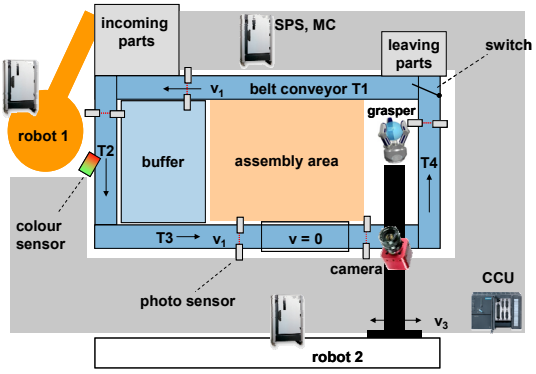


Figure 7: Layout of use case.

5 CONCLUSIONS

In the scope of this research project we hope to achieve a complete assembly operation by cognitive means and therefore reducing the dilemma of planning and value orientation by means of self organisational systems. In the Cluster of Excellence this is one of the researched approaches. In this paper requirements for a cognitive technical system applicable in production environments and a first concept of a software architecture have been presented. Furthermore possible knowledge representation forms which could be suitable for a deployment in production environments were shown. The future work will focus on the implementation of the needed domain knowledge for a handling operation and the interaction of human controllers with the system. Furthermore we will

develop a knowledge translator which satisfies the requirements given in Chapter 3.

ACKNOWLEDGEMENTS

The authors would like to thank the German Research Foundation DFG for the support of the depicted research within the Cluster of Excellence "Integrative Production Technology for High-Wage Countries".

REFERENCES

Anderson, J. R. et al. 2004. *An integrated theory of the mind*. Psychological Review 111.

Burghart, C., et al. 2005. A Cognitive Architecture for a Humanoid Robot: A First Approach. *Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots, Tsukuba.* IEEE Service Center.

Doorenbos, R., 1995. *Production Matching for Large Learning Systems.* Carnegie Mellon University, pp. 2-6.

Gruber, T., 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, Vol. 5, No. 2. (June 1993), pp. 199-220.

Hartmann, E. 1995. *Eine Methodik zur Gestaltung kognitiv kompatibler Mensch-Maschine-Schnittstellen, angewandt am Beispiel der Steuerung einer CNC-Drehmaschine,* Verlag der Augustinus Buchhandlung, Aachen.

Haun, M., 2000. *Wissensbasierte Systeme*. Expert Verlag.

Heide, A., Henning, K., 2006. The cognitive car - A roadmap for research issues in the automotive sector. *Proceedings of the 9th IFAC Symposium on Automated Systems Based on Human Skill And Knowledge*. IFAC.

Laird, J., 2006. *A Gentle Introduction to SOAR*, *an architecture for human cognition*, Michigan.

Matlin, M. W., 2005. *Cognition*., Wiley, New York.

Newell, A., 1990. *Unified Theories of Cognition*., Harvard University press, Cambridge.

Nolden, A., 1999. *Handeln in Virtualität und Realität – ein integriertes Lernsystem für das Laserschweissen.* VDI Verlag, Düsseldorf. pp. 22-27

Paetzold, K., 2006. On the importance of a functional description for the development of cognitive technical systems*, International Design Conference – Design 2006, Dubrovnik*

Putzer, H. J., 2004. *Ein uniformer Architekturansatz für Kognitive Systeme und seine Umsetzung in ein operatives Framework,* Verlag Dr. Köster, Berlin, pp. 4-10.

Schuh, G., 2007. Roadmapping for Competitiveness of High Wage Countries, *18th ISPIM Conference, Warsaw*

Smith, M., et al. 2004. OWL Web Ontology Language Guide*, http://www.w3.org/TR/2004/REC-owl-guide-20040210/*

OBTAINING MINIMUM VARIABILITY OWA OPERATORS UNDER A FUZZY LEVEL OF ORNESS

Kaj-Mikael Björk
Department of Technology, IAMSR, Åbo Akademi University
Joukahaisenkatu 4-6A, 20520 Turku, Finland
kaj-mikael.bjork@abo.fi

Keywords: OWA operators, Fuzzy numbers, Optimization, Signed distance-defuzzification.

Abstract: Finding the optimal OWA (ordered weighted averaging) operators is important in many decision support problems. The OWA-operators enables the decision maker to model very different kinds of aggregator operators. The weights need to be, however, determined under some criteria, and can be found through the solution of some optimization problems. The important parameter called the level of orness may, in many cases, be uncertain to some degree. Decision makers are often able to estimate the level using fuzzy numbers. Therefore, this paper contributes to the current state of the art in OWA operators with a model that can determine the optimal (minimum variability) OWA operators under a (unsymmetrical triangular) fuzzy level of orness.

1 INTRODUCTION

Information aggregation is used in many applications. Some fields of research that takes advantage of aggregation may be found in Neural Networks, fuzzy logic controllers, multi-criteria optimisation and more. Aggregation is necessary to logically split up entities onto several units. A very eminent way of doing aggregators is the OWA operators, originally described by (Yager, 1988). He defined a weight, w_i, to be associated with an ordered position of the aggregate. The weights are often ordered such that the best criterion is associated with the first weight and so on. Given the weights for each object, Yager defined a level of orness, which will represent a major characteristic of the weighting structure. An orness-value of zero represents a situation that the weakest criterion has the full weight, whereas an orness-value of one represents the opposite, i.e. the strongest criterion has the full weight.

Finding the optimal distribution of the weights under a certain level of orness has obtained some interest during the last decade. The weights can be optimal in many ways; O'Hagan, for instance (1988), presented a numerical method to find the maximum entropy OWA operators under a crisp level of orness. Quite recently (Fuller and Majlender, 2001), (Fuller and Majlender, 2003) and (Carlsson et al., 2003) extended those results with both a analytical model for the maximum entropy problem as well as an analytical solution to the minimum variability problem. These contributions are interesting and sound theoretical findings. They did not, however, consider a fuzzy level of orness. The level of orness is often estimated from expert opinions and can be inherent fuzzy. Therefore, this paper contributes with a fuzzy orness level, minimum variability, OWA operator model. This paper does not use the Lagrange multiplier method, used by (Fuller and Majlender, 2001, 2003), but instead the constraints for the minimum variability problem are substituted in the objective function. Afterwards, the objective function is assumed to have a triangular fuzzy level of orness. This paper uses the signed distance method (Yao and Wu, 2000) to defuzzify the objective function, where after the optimisation problem is checked for convexity and solved numerically to the optimal solution. Other contributions using the signed distance method to defuzzify fuzzy numbers are (Salameh and Jaber, 2000) and (Yao and Chiang, 2003), for instance.

The paper is organised as follows: first the minimum variability OWA operator problem is formulated. Then the problem is altered to contain only an objective function, where after the level of orness is allowed to be fuzzy, but defuzzified in

order to obtain a crisp optimal solution to the problem. Finally, a small problem is solved and compared with the solution obtained by (Fuller and Majlender, 2003).

2 THE MINIMUM VARIABILITY OWA OPERATOR PROBLEM

According to (Fuller and Majlender, 2003), the minimum variability problem is the following:

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=1}^{n} w_i^2 - \frac{1}{n^2} \\ s.t. \quad & orness(w) = \sum_{i=1}^{n} \frac{n-i}{n-1} w_i = \alpha, \quad 0 \le \alpha \le 1 \\ & \sum_{i=1}^{n} w_i = 1 \end{aligned} \tag{1}$$

where w_i is the positive weigths (the variables in the optimisation problem) and n is the total number of weights. α is the level of orness (parameter in the optimisation problem). This model can be solved analytically to optimum using a Lagrange multiplier method as in Fuller and Majlender (2003).

The model in eq. (1) can be reformulated by substituting each of the two constraints into the objective function. The second constraint in (1) will give us the following relationship:

$$w_1 = 1 - \sum_{i=2}^{n} w_i \tag{2}$$

Subsituting (2) into (1) yields

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=2}^{n} w_i^2 - \frac{1}{n^2} + \frac{1}{n}\left(1 - \sum_{i=2}^{n} w_i\right)^2 \\ s.t. \quad & \sum_{i=2}^{n} \frac{n-i}{n-1} w_i + \left(1 - \sum_{i=2}^{n} w_i\right) = \alpha, \end{aligned} \tag{3}$$

The constraint in (3) will give us the following relationship:

$$w_2 = -\sum_{i=3}^{n} (i-1) w_i + (n-1)(1-\alpha) \tag{4}$$

Using (4) in (3) will give us the simplified optimisation problem, containing only an objective function as follows (after some simplifications)

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=3}^{n} w_i^2 - \frac{1}{n^2} \\ & + \frac{1}{n}\left(n + \alpha - n\alpha - 1 - \sum_{i=3}^{n} (i-1) w_i\right)^2 \\ & + \frac{1}{n}\left(-n - \alpha + n\alpha + 2 + \sum_{i=3}^{n} (i-2) w_i\right)^2 \end{aligned} \tag{5}$$

First of all it is worth noticing that the optimisation problem in eq. (5) is convex. The convexity can be established by examining the terms and since classical convexity theory states that a function $f(x) = (\sum_i c_i x_i + k)^2$, is always convex, where c_i and k are constants.

The next step is to manipulate (5) to remove the squares (in order to be able to defuzzify it with the signed distance method). This will result (after some simplifications and rearrangements) in the following problem (i.e. to an equivalent problem to the one found in eq. 5):

$$\begin{aligned} \min \quad & -\frac{1}{n^2} + 2n + 2\frac{\alpha^2}{n} + 2n\alpha^2 + \frac{5}{n} \\ & + 10\alpha - 4\alpha^2 - 4n\alpha - 6\frac{\alpha}{n} - 6 \\ & + \frac{1}{n}\sum_{i=3}^{n} (2i^2 - 6i + 6) w_i^2 - \sum_{i=3}^{n} (4i-6) w_i \\ & - \frac{\alpha}{n}\sum_{i=3}^{n} (4i-6) w_i + \frac{2}{n}\sum_{i=3}^{n} (3i-5) w_i \\ & + \alpha \sum_{i=3}^{n} (4i-6) w_i \\ & + \frac{2}{n}\sum_{i=3}^{n-1}\sum_{j=1+i}^{n} (i-1)(j-1) w_i w_j \\ & + \frac{2}{n}\sum_{i=3}^{n-1}\sum_{j=1+i}^{n} (i-2)(j-2) w_i w_j \end{aligned} \tag{6}$$

There are some intermediate steps between eqs. (5) and (6) that are left out since it would require some additional pages of formulas. The reformulation of eq. (5) into the problem in eq. (6) may seem to complicate the problem structure, but in fact it helps the defuzzification step, since α will only be found in separate terms.

3 DEFUZZIFICATION OF THE ORNESS VALUE

If the α value (the orness value) is triangular fuzzy, denoted as $\tilde{\alpha}$, the optimization problem becomes simply the following:

$$\begin{aligned}\min\ & -\frac{1}{n^2}+2n+2\frac{\tilde{\alpha}^2}{n}+2n\tilde{\alpha}^2+\frac{5}{n}\\ &+10\tilde{\alpha}-4\tilde{\alpha}^2-4n\tilde{\alpha}-6\frac{\tilde{\alpha}}{n}-6\\ &+\frac{1}{n}\sum_{i=3}^{n}\left(2i^2-6i+6\right)w_i^2-\sum_{i=3}^{n}\left(4i-6\right)w_i\\ &-\frac{\tilde{\alpha}}{n}\sum_{i=3}^{n}\left(4i-6\right)w_i+\frac{2}{n}\sum_{i=3}^{n}\left(3i-5\right)w_i\\ &+\tilde{\alpha}\sum_{i=3}^{n}\left(4i-6\right)w_i\\ &+\frac{2}{n}\sum_{i=3}^{n-1}\sum_{j=1+i}^{n}(i-1)(j-1)w_iw_j\\ &+\frac{2}{n}\sum_{i=3}^{n-1}\sum_{j=1+i}^{n}(i-2)(j-2)w_iw_j\end{aligned} \tag{7}$$

Some basics from fuzzy set theory need to be introduced in order to make the following model development self-contained.

Definition 1. Consider the fuzzy set $\tilde{A}=(a,b,c)$ where $a<b<c$ and defined on R, which is called a triangular fuzzy number, if the membership function of $\tilde{A}$ is given by

$$\mu_{\tilde{A}}(x)=\begin{cases}(x-a)/(b-a), & a\le x\le b,\\ (c-x)/(c-b), & b\le x\le c\\ 0, & \textit{otherwise.}\end{cases}$$

Definition 2. Let $\tilde{B}$ be a fuzzy set on R and $0\le\alpha_c\le 1$. The α_c-cut of $\tilde{B}$ is all the points x such that $\mu_{\tilde{B}}(x)\ge\alpha_c$, i.e.

$$B(\alpha_c)=\left\{x\middle|\mu_{\tilde{B}}(x)\ge\alpha_c\right\}$$

In order to find non-fuzzy values for the model we need to use some distance measures and we will use the signed distance (Yao and Wu, 2000).

Definition 3. For any a and $0\in R$, the signed distance from a to 0 is $d_0(a,0)=a$. And if $a<0$, the distance from a to 0 is $-a=-d_0(a,0)$.

Let Ω be the family of all fuzzy sets $\tilde{B}$ defined on R for which the α-cut $B(\alpha_c)=\left[B_L(\alpha_c),B_U(\alpha_c)\right]$ exists for every $\alpha_c\in[0,1]$, and both $B_L(\alpha_c)$ and $B_U(\alpha_c)$ are continuous functions on $\alpha_c\in[0,1]$. Then, for any $\tilde{B}\in\Omega$, we have (see Chang, 2004, for instance)

$$\tilde{B}=\bigcup_{0\le\alpha_c\le 1}\left[B_L(\alpha_c)_{\alpha_c},B_U(\alpha_c)_{\alpha_c}\right]$$

From Chang (2004) it can be finally stated (originally by results from Yao and Wu, 2000) how to calculate the signed distances.

Definition 4. For $\tilde{B}\in\Omega$ define the signed distance of $\tilde{B}$ to $\tilde{0}_1$ as

$$d(\tilde{B},\tilde{0}_1)=\frac{1}{2}\int_0^1\left[B_L(\alpha_c)+B_U(\alpha_c)\right]d\alpha_c.$$

The Definition 3 will give us several properties of which the most important is

Property 1. Consider the triangular fuzzy number $\tilde{A}=(a,b,c)$: the α-cut of $\tilde{A}$ is $A(\alpha_c)=\left[A_L(\alpha_c),A_U(\alpha_c)\right]$, for $\alpha_c\in[0,1]$, where $A_L(\alpha_c)=a+(b-a)\alpha_c$ and $A_U(\alpha_c)=c-(c-b)\alpha_c$, the signed distance of $\tilde{A}$ to $\tilde{0}_1$ is

$$d(\tilde{A},\tilde{0}_1)=\frac{1}{4}(a+2b+c).$$

Let us assume that we have a triangular fuzzy orness level, i.e.

$$\tilde{\alpha}=(\alpha-\Delta_l,\alpha,\alpha+\Delta_h) \tag{8}$$

(Note that the orness value, α, should not be mixed up with the α-cut, called α_c.) Then we will defuzzify $\tilde{\alpha}$ in two different ways, depending on whether $\tilde{\alpha}$ is squared or not. From Property 1 we will get directly that the signed distance of $\tilde{\alpha}$ is

$$\begin{aligned}d(\tilde{\alpha},\tilde{0})&=\frac{1}{4}\left[(\alpha-\Delta_l)+2\alpha+(\alpha+\Delta_h)\right]\\ &=\alpha+\frac{1}{4}\Delta_h-\frac{1}{4}\Delta_l\end{aligned} \tag{9}$$

And according to Definition 4 we will get that the signed distance for $\tilde{\alpha}^2$ will be

$$d(\tilde{\alpha}^2,\tilde{0}) = \frac{1}{2}\int_0^1 \left[\left(\alpha^2\right)_L(\alpha_c) + \left(\alpha^2\right)_U(\alpha_c)\right] d\alpha_c$$

$$= \frac{1}{2}\int_0^1 \begin{bmatrix} \left(\alpha - \Delta_l + \Delta_l\alpha_c\right)^2 \\ + \left(\alpha + \Delta_h - \Delta_h\alpha_c\right)^2 \end{bmatrix} d\alpha_c$$

$$= \frac{1}{2}\int_0^1 \begin{bmatrix} \alpha^2 - \alpha\Delta_l + \alpha\Delta_l\alpha_c - \alpha\Delta_l + \Delta_l{}^2 \\ - \Delta_l\alpha_c + \alpha\Delta_l\alpha_c - \Delta_l{}^2\alpha_c + \Delta_l{}^2\alpha_c{}^2 \\ + \alpha^2 + \alpha\Delta_h - \alpha\Delta_h\alpha_c + \alpha\Delta_h \\ + \Delta_h{}^2 - \Delta_h{}^2\alpha_c - \alpha\Delta_h\alpha_c + \Delta_h{}^2\alpha_c \\ + \Delta_h{}^2\alpha_c{}^2 \end{bmatrix} d\alpha_c \quad (10)$$

Finally we will get the signed distance value for $\tilde{\alpha}^2$ as

$$d(\tilde{\alpha}^2,\tilde{0}) = \alpha^2 - \frac{1}{2}\alpha\Delta_l + \frac{1}{2}\alpha\Delta_h + \frac{1}{6}\Delta_l{}^2 + \frac{1}{6}\Delta_h{}^2 \quad (11)$$

The defuzzified objective function will be

$$\begin{aligned} &\min\ -\frac{1}{n^2} + 2n + 2\frac{d(\tilde{\alpha}^2,\tilde{0})}{n} + 2n\cdot d(\tilde{\alpha}^2,\tilde{0}) \\ &+\frac{5}{n} + 10d(\tilde{\alpha},\tilde{0}) - 4d(\tilde{\alpha}^2,\tilde{0}) - 4n\cdot d(\tilde{\alpha},\tilde{0}) \\ &-6\frac{d(\tilde{\alpha},\tilde{0})}{n} - 6 + \frac{1}{n}\sum_{i=3}^{n}\left(2i^2 - 6i + 6\right)w_i^2 \\ &-\sum_{i=3}^{n}(4i-6)w_i - \frac{d(\tilde{\alpha},\tilde{0})}{n}\sum_{i=3}^{n}(4i-6)w_i \\ &+\frac{2}{n}\sum_{i=3}^{n}(3i-5)w_i + d(\tilde{\alpha},\tilde{0})\sum_{i=3}^{n}(4i-6)w_i \\ &+\frac{2}{n}\sum_{i=3}^{n-1}\sum_{j=1+i}^{n}(i-1)(j-1)w_iw_j \\ &+\frac{2}{n}\sum_{i=3}^{n-1}\sum_{j=1+i}^{n}(i-2)(j-2)w_iw_j \end{aligned} \quad (12)$$

And putting the signed distances (to defuzzify) of $\tilde{\alpha}$ and $\tilde{\alpha}^2$ respectively (Equations 9 and 11), into eq. (7) will give us the final defuzzified objective function as

$$\begin{aligned} &\min\ -\frac{1}{n^2} + 2n + \\ &2\frac{\alpha^2 - \frac{1}{2}\alpha\Delta_l + \frac{1}{2}\alpha\Delta_h + \frac{1}{6}\Delta_l{}^2 + \frac{1}{6}\Delta_h{}^2}{n} \\ &+2n\cdot\left(\alpha^2 - \frac{1}{2}\alpha\Delta_l + \frac{1}{2}\alpha\Delta_h + \frac{1}{6}\Delta_l{}^2 + \frac{1}{6}\Delta_h{}^2\right) \\ &+\frac{5}{n} + 10\left(\alpha + \frac{1}{4}\Delta_h - \frac{1}{4}\Delta_l\right) \\ &-4\left(\alpha^2 - \frac{1}{2}\alpha\Delta_l + \frac{1}{2}\alpha\Delta_h + \frac{1}{6}\Delta_l{}^2 + \frac{1}{6}\Delta_h{}^2\right) \\ &-4n\cdot\left(\alpha + \frac{1}{4}\Delta_h - \frac{1}{4}\Delta_l\right) \\ &-6\frac{\alpha + \frac{1}{4}\Delta_h - \frac{1}{4}\Delta_l}{n} - 6 + \frac{1}{n}\sum_{i=3}^{n}\left(2i^2 - 6i + 6\right)w_i^2 \\ &-\sum_{i=3}^{n}(4i-6)w_i - \frac{\alpha + \frac{1}{4}\Delta_h - \frac{1}{4}\Delta_l}{n}\sum_{i=3}^{n}(4i-6)w_i \\ &+\frac{2}{n}\sum_{i=3}^{n}(3i-5)w_i + \left(\alpha + \frac{1}{4}\Delta_h - \frac{1}{4}\Delta_l\right)\sum_{i=3}^{n}(4i-6)w_i \\ &+\frac{2}{n}\sum_{i=3}^{n-1}\sum_{j=1+i}^{n}(i-1)(j-1)w_iw_j \\ &+\frac{2}{n}\sum_{i=3}^{n-1}\sum_{j=1+i}^{n}(i-2)(j-2)w_iw_j \end{aligned} \quad (13)$$

The fuzzy minimum variability OWA operator problem can thus be solved by minimizing eq. (13). The first and second weight can there-after be obtained from eqs. (4) and (2), respectively (with the defuzzified value of $\tilde{\alpha}$, and not the crisp one, c.f. eq. 14).

$$w_2 = -\sum_{i=3}^{n}(i-1)w_i + (n-1)\left(1 - d(\tilde{\alpha},\tilde{0})\right) \quad (14)$$

It is worth noticing that the convexity will remain (from eq. 5) through the operations, since the effect of a fuzzy orness-value (α-value) will only affect the constant in the optimization problem. (I.e. it will only affect the parameter k in the functions of the form $f(x) = (\sum_i c_i x_i + k)^2$ and, thus, not affect the convexity. In addition, the operations in eqs. 6-13 will not change the convexity assumption.) The optimization problem in eq. (13) can be solved numerically with any local nonlinear optimization methods, which can guarantee local optimal convergence. The method need not be able to handle constraints, since there are no constraints involved in eq. (13), except for the non-negativity constraint of the variables. A method that can handle simple

constraints is, however, advisable so that the substituted constraints in eqs. (2) and (4) will always get non-negative values.

4 EXAMPLE

In this section, a test problem is solved and compared to the crisp solution by Fuller and Majlender (2003). This problem contains 5 weights and it is calculated for a level of orness (α-value) of 0.1, 0.2, 0.3, 0.4 and 0.5. First, the problem is compared to the crisp solution for an α-value of 0.3 and different values of the Δ-parameters (i.e. different fuzziness values). The solution is obtained by using a standard local search method on the problem in eq. (13). The problem in this paper is solved with the extended Newton method found in the standard solver available in Microsoft Excel.

Table 1: The optimal OWA-operators for different fuzziness values (α=0.3).

α	Δ_l	Δ_h	w_1	w_2	w_3	w_4	w_5	Obj
0.300	0.000	0.000	0.040	0.120	0.200	0.280	0.360	0.013
0.300	0.050	0.050	0.040	0.120	0.200	0.280	0.360	0.018
0.300	0.100	0.050	0.030	0.115	0.200	0.285	0.370	0.027
0.300	0.050	0.100	0.050	0.125	0.200	0.275	0.350	0.024

In Table 1 it should be noted that the crisp case (i.e. when the Δ's are 0) collapses to the same solution as reported in Fuller and Majlender (2003). It should also be noted that the optimal solution (in this example) remained the same as the crisp solution if $\Delta_l = \Delta_h$. In order to illustrate the behaviour of the weights for different Δ-values (as well as the objective function), Figure 1 and Figure 2 are included. In these figures, the α-value is set to 0.3, but one of the Δ-values is allowed to change. One can see in Figure 1 that if Δ_h is increased from 0 to 0.3 the objective value increases from 0.013 to 0.065 and the weights get more similar to each other. In a similar manner when Δ_l increasing from 0 to 0.3, the objective value will increase from 0.013 to 0.084 and the weights become more diverse.

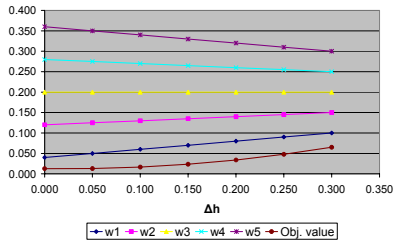


Figure 1: The sensitivity analysis of Δ_h for α=0 and Δ_l=0.

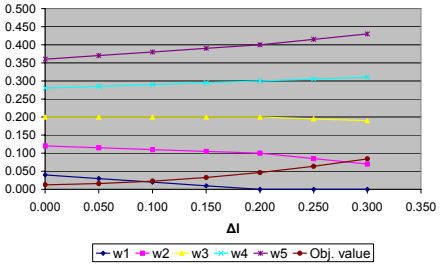


Figure 2: The sensitivity analysis of Δ_l for α=0 and Δ_h=0.

In Table 2, the optimal OWA-operators for several α-values are calculated. When the $\Delta_l = \Delta_h = 0$, (i.e. the crisp case) the operator-values are the same as the one reported by Fuller and Majlender (2003). In the case of Δ-values greater than zero (and unequal) the operator-values are different from the crisp case, except for the case of α=0.1. It is also worth noticing that the objective value for the crisp case is always better than for the fuzzy cases (in this example); when α=0.1 the increase is only about 20 %, but with bigger α-values, the bigger the increase in the objective function when fuzziness is introduced.

Table 2: The optimal OWA-operators for different α-values as well as fuzziness values.

α	Δ_l	Δ_h	w_1	w_2	w_3	w_4	w_5	Obj
0.100	0.000	0.000	0.000	0.000	0.033	0.333	0.633	0.063
0.100	0.050	0.100	0.000	0.000	0.058	0.333	0.608	0.069
0.100	0.100	0.050	0.000	0.000	0.008	0.333	0.658	0.081
0.200	0.000	0.000	0.000	0.040	0.180	0.320	0.460	0.030
0.200	0.050	0.100	0.000	0.055	0.185	0.315	0.445	0.039
0.200	0.100	0.050	0.000	0.025	0.175	0.325	0.475	0.045
0.400	0.000	0.000	0.120	0.160	0.200	0.240	0.280	0.003
0.400	0.050	0.100	0.130	0.165	0.200	0.235	0.270	0.015
0.400	0.100	0.050	0.110	0.155	0.200	0.245	0.290	0.016
0.500	0.000	0.000	0.200	0.200	0.200	0.200	0.200	0.000
0.500	0.050	0.100	0.210	0.205	0.200	0.195	0.190	0.012
0.500	0.100	0.050	0.190	0.195	0.200	0.205	0.210	0.012

5 CONCLUSIONS

This paper presents a new fuzzy minimum variability model for the OWA-operators, originally introduced by Yager (1988). Previous results in this line of research is the elegant results by Fuller and Majlender (2001, 2003), where both the minimum variability problem as well as the maximum entropy problem were solved. These results assumed, however, a crisp level of orness.

This paper added the current research track a model that could account for unsymmetrical (or symmetrical) triangular fuzzy levels of orness. This is important if the decision maker is not certain

about the level of orness, but can estimate it through the proposed fuzzy numbers. The minimum variability model for the fuzzy orness level is obtained through a slightly different approach than the one used in Fuller and Majlender (2001, 2003). This paper substitutes the constraints in the problem (c.f. eq. 1) such that the variables w_1 and w_2 are eliminated out of the problem, and after some rearrangements a convex objective in smaller dimension remains of the original problem. This problem is allowed to have triangular fuzzy α-values, but in order to solve the optimisation problem, the α-values are defuzzified with the signed distance method. The defuzzified optimization problem is then solved with a numerical optimisation method that can guarantee local convergence. The first two weights are then solved from the substituted constraints.

The future research consists of analytical solutions for the optimization problem as well as extending the level of orness to contain other types of fuzzy numbers than the triangular ones. A natural extension could be to investigate the trapezoidal fuzzy numbers as well as other defuzzification methods.

REFERENCES

Carlsson C, Fuller R. & Majlender P., 2003. A note on constrained OWA aggregation. *Fuzzy sets and* systems, 139, pp. 543-546.

Chang H-C., 2004. An application of fuzzy sets theory to the EOQ model with imperfect quality items. *Computers & Operations Research*, 31, pp. 2079-2092.

Fuller R. & Majlender P., 2001. An analytical approach for obtaining OWA-operator weights. *Fuzzy sets and* systems, 124, pp. 53-57.

Fuller R. & Majlender P., 2003. On obtaining minimum variability OWA operator weights. *Fuzzy sets and* systems, 136, pp. 203-215.

O'Hagan M., 1988. Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic. *Proceedings of 22nd annual Asilomal conference on signals, systems and computers*, pp. 681-689.

Salameh M.K. and Jaber M.Y., 2000. Economic production quantity model for items with imperfect quality. *Int. Journal of Production Economics*, 64, pp.59-64.

Yager R.R., 1988. Ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18, pp. 183-190.

Yao J.S. and Wu K., 2000. Ranking fuzzy numbers based on decomposition principle and signed distance, *Fuzzy Sets and Systems*, 116, pp. 275-288.

Yao J-S and Chiang J., 2003. Inventory without backorder with fuzzy total cost and fuzzy storing cost defuzzified by centroid and signed distance. *European Journal of Operational Research*, 148, pp. 401-409.

AN APPROACH FOR A KNOWLEDGE-BASED NC PROGRAMMING SYSTEM

Ulrich Berger, Ralf Kretzschmann and Jan Noack
Brandenburg University of Technology, Siemens-Halske-Ring 14, Cottbus, Germany
Ulrich.Berger@ tu-cottbus.de, Ralf.Kretzschmann@ tu-cottbus.de, Jan.Noack@ tu-cottbus.de

Keywords: NC-programming, Graph theory, Process planning, Planning and Scheduling, Knowledge management.

Abstract: Nowadays, significant deficiencies exist in the information flow and access along the NC (Numerical Control) process chain. These deficiencies are solved insufficiently by introducing CAD/CAM systems and feature-oriented specification languages. In contrast the application of new production and new machining systems requires an intensive information exchange. The introduced approach enables the preparation for feature-based work plans with methods known from the graph theory as a knowledge-based NC programming system. Therefore the work plan will be mapped into a directed graph in a mathematically defined way. Based on that, it is possible to use algorithms to find the shortest path and a Hamiltonian path inside this directed graph under given requirements. Thus, the work plan will be structured and re-ordered. Finally the corresponding NC paths will be generated and distributed to the machinery. Thence in this in-process paper the requirements, the investigation and the selection of suitable knowledge structuring concepts, mathematically basics and the work flow in such a system will be pointed out. Finally a preliminary prototype will be introduced.

1 INTRODUCTION

The current situation for mould and die makers is characterized by increasing requirements from the customer side on quality and individualization of products (e.g. tools) and also uprising pressure on product prices at the same time. Furthermore, new manufacturing technologies like the hybrid combination of milling with dry iced blasting and laser ablation for high precision finishing are already introduced and established. Thus, the NC process chain is getting more complex. Additionally more information has to be handled and exchanged between the different phases of the process change. However, the effective programming of NC programs is still a bottle-neck within this process chain. The reason is the mangled information exchange caused by prevalent organizational and technical obstacles like the use of the DIN 66025 / ISO 6983 as NC programming interface (DIN 66025, 1983). A solution to optimize NC programming will be enabled by introducing feature technology. With the help of knowledge-based machining feature as the elemental part of a NC program the selection and re-ordering of these elements can be executed in a mathematical defined way. With the help of mathematical defined mappings, it is possible to transfer a work plan of a NC program in a directed graph (Hamelmann, 1995) for further processing. As a consequence, the well-known algorithm from the graph theory and combinatorics can be used to process and optimize the graphs under given requirements as objective functions e.g. time reduction, in a traceable way. As a consequence present requirements from the industry (e.g. more transparency and reducing the processing time) can be achieved.

The following article presents an approach for structuring and re-ordering NC programs in a knowledge-based NC programming system based on feature technology and the application of algorithms known from the graph theory.

2 STATE OF THE ART AND RELATED WORK

2.1 NC-Process Chain

The NC process chain consists of three fundamental steps. First of all, the part (workpiece) including manufacturing requirements has to be designed with

powerful CAD (Computer Aided Design) software systems.

Afterwards, the following planning phase consists of further detailed sub process-steps. The main step is the preparation of the work plan which will be used for the NC programming in order to compile the instructions for the NC machining. Thus, a work plan summarizes all information about the machining task including the raw part geometry, the workshop equipment and the sequence of the machining operations. The complexity of the workpiece and the generated work plan determines the used NC programming method. The common used methods are shop-floor-oriented programming procedure (SOP) and powerful CAP (Computer Aided Planning) and/or CAM (Computer Aided Manufacturing) software applications in order to generate the DIN 66025 / ISO 6983 based NC code instruction (DIN 66025, 1983).

Finally, the part will be machined by NC machines executing these generated instructions in the shop-floor as the last step in the process chain (Eversheim, 1996). The NC programs have to be distributed to the dedicated machinery at the right time according to the production, planning, and scheduling results by a DNC (Distributed Numerical Control) system. These shop-floor scheduling tasks are supported by MES (Manufacturing Execution System) based on models like MRP II (Manufacturing Resources Planning). Determining factors for the shop-floor scheduling are the machinery, the tools, the operators and constraints regarding to the capacity and time scheduling (Eversheim, 1996).

2.3 Problems and Challenges

As already mentioned the NC machining process is characterized by permanent enhancements. There are five factors, which influence these enhancement and complexity of the process chain as seen in Figure 1. The first and the second factors are dealing with the manufacturing of complex products with variable lot sizes, the machining of new resources e.g. expensive materials which are difficult to machine like nickel for turbine blades. The consequence is that complex programs have to be generated. So, more potential programming errors has to be avoided in the CAM system. But these systems do not consider all information concerning the machinery. The result is that the programs are not 100% verified (Warnecke, Valous, 1993). The third part deals with the operator on the machine and the programmer in the process planning phase. The fourth factor is the use of multi-axis-machining and hybrid technologies like machining with dry ice blasting and laser ablation. Therefore the NC process chain comprehends knowledge intensive processes. A new knowledge exchange is enabled within these phases with the help of new methods (fifth factor) like the use of integrated design and process planning by joining CAD and CAM systems and introducing the feature technology paradigm.

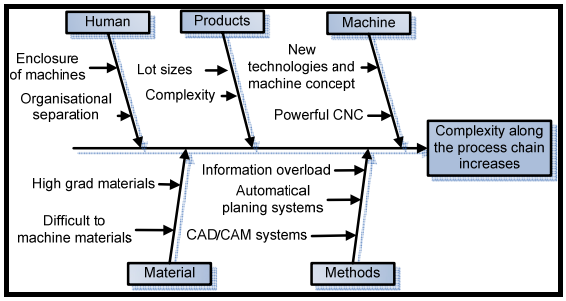


Figure 1: Complexity along the NC process chain.

Several solution approaches are presented in last years in order to reduce the disadvantages of the DIN 66025. The most promising project STEP-NC as a high level and feature oriented programming language demonstrated very good laboratory result (Weck et al., 2001). The main idea was to provide the CNC controller more structural information about the machining process as the DIN 66025 does. Thus, several research projects using STEP-NC were very promising (Pritschow, Heusinger, 2005). Apparently STEP-NC has been aborted because of the complexities in its practical implementation. Further approaches retaining the DIN 66025 were introduced by Gerken (Gerken, 2000) and by Hamelmann (Hamelmann 1996). The concepts of Gerken used Case-based-Reasoning (CBR) for finding a suitable machining operation for a given unknown machining task by comparing already machined tasks with the selected operation. In contrast to that Hamelmann used rules-based structures to store the process planning and operating knowledge in order to reuse it. Nevertheless, the information and knowledge exchange from and to the shop floor is still handicapped by using the DIN 66025 exclusively for generating NC programs for a fixed setup. Thus, the short-term shop-floor scheduling is prevented. As a consequence, it is hard to modify these NC programs, if the allocation will change because of unpredictable states in the machinery. An information feedback of these modified NC programs fails because of missing or insufficient

information feedback possibilities back to the design and process planning. As the result, the knowledge is kept on the one hand in the design and process planning phase and on the other hand in the shop-floor. Due to the missing link to the CAM information, recurrent problems occur (like broken die and collision), while setting up a machine in the shop-floor with a new program (Warnecke, Valous, 1993). New solutions are needed.

3 REQUIREMENTS FOR A KNOWLEDGE-BASED NC PROGRAMMING SYSTEM

Generally, an approach for such a system has to structure and re-order the programs in a traceable and customized way. Regarding to the actual machinery and production state it must be possible to change parts of the NC program (operation scheduling) and the assignments of the machine in the machinery. The fundamental basis will be the use of process knowledge provided by the employees involved in the NC process chain and process monitoring. As a result, the feature technology along the whole NC process chain will be utilized. These features have to be used to describe the physical design of the workpiece (CAD model) and the assigned best-practice machining operations to machine the workpiece (Hamelmann, 1995). Finally, the concrete order of the machining features will be translated into the DIN 66025 instructions. A feedback from the shop-floor must be established to guarantee and validate an optimal assignment of machining operation to design features. An essential aspect is to enable the feedback of the operators back to the process planning about unsuccessful machining operations.
Summarizing, the following requirements have to be maintained (Berger et al, 2007):

- Feature-based description of the process planning to benchmark machine operations,
- Knowledge-based algorithms for structuring and re-ordering the work plan,
- Establishing different user profiles with access authorization to planning documents,
- DIN 66025 as NC programming language,
- "Add-on" to the prevailing IT architecture,
- Modular architecture for roll out of STEP-NC

4 DEVELOPMENT OF A KNOWLEDGE-BASED NC-PROGRAMMING SYSTEM

The development can be divided in four main steps. At first suitable mathematical descriptions for defining a work plan have to be investigated. Secondly, the process planning has to be defined. Thirdly, the workflow of the system has to be described. Thirdly, the NC programming system has to be established as an "Add-on" with access to several databases according to user profiles and the prevailing IT architecture.

4.1 Investigating the Mathematical Description for a Work Plan

Feature-based concepts have been established already in order to define a work plan (Gerken, 2000) (Weck et al., 2001) (VDI 2218, 2003). The process of machining a job can be subdivided hierarchically in several process plan elements (Gerken, 2000). So, the machining task (as machining feature) of drilling a step drilling can be divided in machining operations and machining steps. Berger and Cai defined a machining step (MStep) as a product of selected tools (Tool), technology (Tech), paths and machining features (MFeat) in order to describe the elements in a mathematical way (Berger et al., 2005). Thus, a machining step has to be expanded with the element machine (Mach). Finally, a set of machining steps can be defined as a set of 5-tuple (1).

$$\begin{aligned} &MStep = \{Tool \times Tech \times Path \times Mach \times MFeat\} \\ &SU \in MStep \end{aligned} \quad (1)$$

Each specific 5-tuple set-up (SU) describes the parameter of a machining process. Consequently, each machining operation is an assembly of machining steps. Furthermore, each work plan consists of a sequence of concrete set-up from MStep to machine the machining task. Unlike the STEP-NC data model machining operation, the MSTEP is defined in a more abstracted way.

4.2 Investigation and Selection of the Mathematical Model for Processing a Work Plan Processing

For processing the information of a work plan a suitable concept which supports the operations of re-ordering and structuring has to be found. Therefore

requirements have to be determined and suitable available concepts have to be benchmarked with these requirements. As already mentioned the knowledge-based NC programming system uses benchmarked features like MStep for structuring the work plan. Alternative work plan are usable to machine the same tasks. Therefore the requirements for a concept are handling time dependency, outlining different alternatives in a work plan and benchmarking operations (e.g. association of MStep). Furthermore, the application of traceable and well known algorithms for selecting / structuring and re-ordering alternative operation is required. Suitable and investigated knowledge representation and knowledge processing concepts known from the Artificial Intelligence (AI) are neural networks, formal rules-based languages and finally methods based on the graph theory (Görz, 2003). Neural networks are a set of "neurons", which are organized in the input layer, processing layer and output layer. Neural networks are trained with special training algorithms. The disadvantage of this "black box" concept is that the neural networks follow processing in non traceable way (Görz, 2003). Therefore, the neural networks do not conform the requirements. Formal and rule-based languages are used to formalize a given well structured domain under discussion (Görz, 2003). Because of missing algorithms for processing structures like work plans this concept is not suited, too. The graph theory as last concept fulfils the given requirements. It is possible to model time dependency. Furthermore, the different paths will be utilized to represent the alternatives within the directed graph. By introducing "costs" for passing paths the benchmarking will be enabled. Finally the use of algorithms introduced in the graph theory like Floyd-Warshall (FW) algorithm and the TSP (Travailing Salesman Problem) algorithm enable the processing of the work plan. The methods from the combinatorics are used to transfer the task of generating an optimized work plan as operations research problem. Thus, a complete work plan consisting of a sequence of elements of MStep can be transferred in a directed graph. A directed graph DG is defined as an arrangement of a set of nodes (edges) V and a set of edges or arcs E. These arcs are connecting two nodes out of the set of nodes (Jungnickel, 1990). Regarding the work plan, the set of separate MStep can be transferred in the set of nodes. The arcs between the nodes will represent the ordering of these two MStep. An arc e = (a,b) means, that the MStep a is ahead of MStep b. Therefore time dependency of different elements from MStep can be modeled. At this moment, an arc has no quantifier and emblematizes a connection between two MStep. By introducing a quantifier, the arcs become an additional dimension "cost". This cost is an assembly of two efforts to pass this arc. The first effort is the cost to apply the first machining step a. The second effort represents the cost to pass to the second step b. It is possible to enhance the second cost by adding additional cost as a sum of all costs, which occur by passing to all other nodes in graph. Thus, a target system of different criteria like machining time, machining quality, and machining energy can be defined to calculate the costs. Therefore a need of additional information concerning the machinery is detected. Now, it is possible to change the value of a quantifier corresponding to a given scenario, which will be described by a concrete specification of these criteria. Summarizing a work plan consisting of MStep can be described as a directed graph with edges and arcs in-between the edges. Now, it is possible to use algorithms known from the graph theory to calculate significant paths through the directed graph. Two major algorithms can be identified which are suitable for the application in the field of NC machining (Jungnickel, 1990). At first, the Floyd-Warshall algorithm calculates the shortest distance (as a total sum of arc quantifiers) between two nodes (machining step) with the help of the transitive closure. With the help of this algorithm different alternatives for the same machine feature can be rated. The complexity is $O(n^3)$, where n is the amount of nodes. Secondly, the TSP solving algorithms can be executed in order to re-order all machining steps in a work plan to get lowest cost in total to machine all MStep in a row (as Hamiltonian path). The TSP is NP-complete. Consequently, the optimal path could be calculated by combining all MStep. But, this algorithm is not an acceptable way for a common number of MStep. There are several algorithms to find a heuristic solution for a Hamiltonian path in acceptable effort. The use of these heuristic algorithms shortens the runtime of the combinatoric algorithm (Jungnickel, 1990).

4.3 Workflow for the Knowledge-Based NC Programming System

After defining the mathematical background the workflow for the system has to be specified. Therefore, the architecture of the system is outlined in Figure 2.

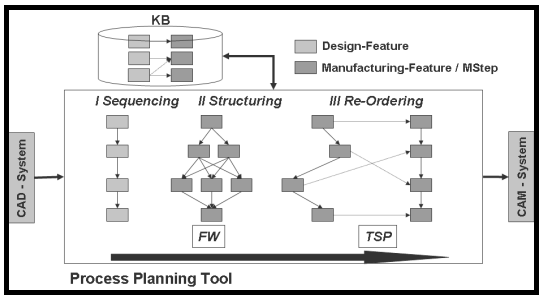


Figure 2: Architecture of the approach.

The knowledge-based NC programming system will be used as a process planning tool. The tool has access to the knowledge-base (KB) and to the CAD and CAM systems. The workflow is divided into three operations (I-III).

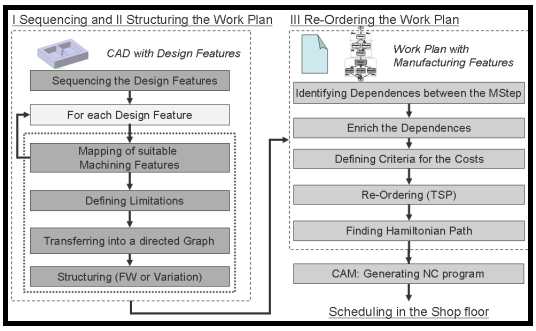


Figure 3: Workflow of the approach.

In Figure 3, the workflow of the approach is presented. The initial task is to generate a work plan for machining a work piece. Therefore, a design feature-based CAD model will be investigated. At first, the sequence of the design feature will be determined (manually or with help of permutation). Afterwards, each design feature will be transferred into suitable alternative machining features which are stored as an assembly of MStep in the KB. Thus, a work plan can described as a sequence of MStep as mentioned before. In Figure 4(a), a specimen work plan is given with three design features, which can be mapped into six machining features (MF). Each of the machining features will be represented by machining steps. The machining features MF1a and MF1b have a shared machining step MStep1 (e.g. roughing) and two alternative steps MStep2a (finishing 3-axis) and MStep2b (finishing 5-axis). The machining feature MF2 will be represented by one machining step. Finally the last design feature can be represented by three machining feature (MF3a … MF3b) with three alternative machining steps (MStep5a … MStep5c). To solve the alternatives, limitations for the machining steps will be given by using scenarios as mentioned before. A scenario describes a special combination of criteria to calculate a quantifier for each element from MStep and each arc. The process planning tool defines a machining scenario from the given scenarios profiles. The selected scenario defines ranges for each criterion. Possible criteria are machining time, tool change count, and surface quality. Afterwards, the process planer will select the right alternative with the Floyd-Warshall (or variation), which calculates the "shortest path" between the additionally added start node "S" and end node "E". A suitable work plan as result is structured and bold marked in Figure 4a.

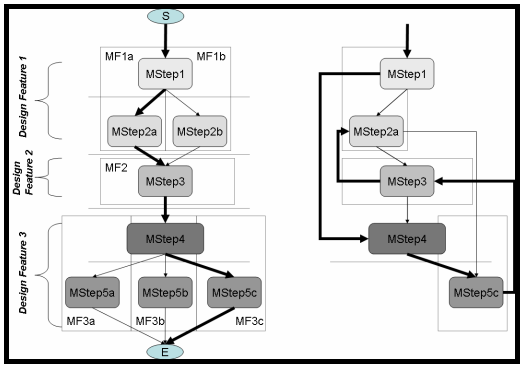


Figure 4: Specimen work plan (a) and directed graph (b) with Hamiltonian path (without quantifiers).

The following task is re-ordering the work plan in order to find a sequence of the MStep with the least sum of quantifiers between the MStep in total. The Hamiltonian path can describe such a path. Therefore the graph has to be enriched with additions of arcs, which represents additional dependencies as seen in Figure 4b. Afterwards calculation criteria will be set, before calculating the Hamiltonian path. These criteria specify the quantifier a second time. Possible quantifiers are the same machine, same clamping, same tool and same type of operation. Finally a suitable Hamiltonian path can be calculated. The grade of the path is depending on the executed algorithm. The best path can be archived by the combination of all possibilities (Brute-Force algorithm). This algorithm is NP-complete. The uses of heuristic algorithms shorten the runtime drastically. But is it not guaranteed, that they will find the best path through the work plan. A possible Hamiltonian path through the example is shown in Figure 4b. Finally a CAM model generates the NC program corresponding to the given re-ordered work plan by translating the

machining steps. The optimized NC program is found. Finally the scheduled NC programs will be dispatched to the corresponding machine. The operator has now the possibility to benchmark the machining steps for re-using them.

4.4 Integration in the NC - Process Chain

The knowledge-based programming system has to be embedded in a guidance system. Therefore it will be structured into two major parts, which are integrated in already existing applications. At first the approach will be integrated in the machines on the shop floor. So the user can access additional planning information with the help of an information system as front-end of the guidance system. Secondly a module called "process planning tool" will be integrated in the guidance system and supports the knowledge-based process planning with the already introduced methodology and functions.

5 TECHNICAL REALIZATION

The technical realization of the concept for the knowledge-based NC programming will be implemented in two steps. At first, the proposed process planning tool will be implemented. Therefore Visual Basic (VB) is used as programming language with access to MS Visio as visualization tool and MS Access as knowledge base. Corresponding to the figure 2, the first (I Sequencing) and the second (II Structuring) part of the architecture are already implemented. The structure of the prototype is outlined in Figure 5. At first the sequenced work plan will be transferred into a VB program and visualized with MS Visio. Afterwards, the FW implementation calculated the optimal structure of the work with the help of additional information about tools, machinery provide by the knowledge base. The next step will be to work on the interface specifications and the investigation of the use of the algorithm for the TSP algorithm. The gist will be the decrease of runtime of this NP-complete algorithm. Thus, several algorithms are investigated which that follow heuristic approaches.

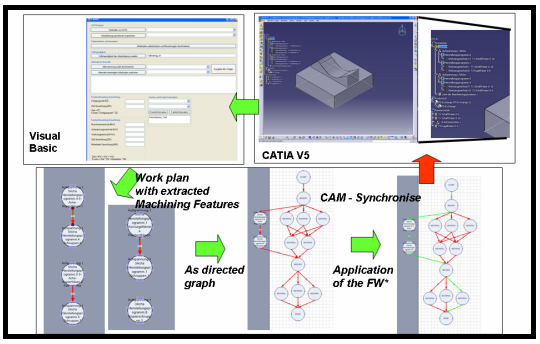


Figure 5: Prototype of the application.

As seen in figure 6 "Brute-Force" algorithms have a high exponential running time. In contrast to that, the heuristic algorithm "All nearest neighbor" (ANN) and the "Minimum spanning tree" has lower running times. The ANN as simple heuristics will be used to carry out the implementation in the way that exclusively time independent MStep will be selected as lowest distance MStep in the ANN algorithm.

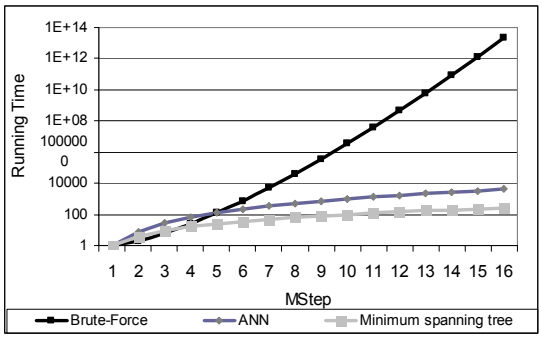


Figure 6: Running time of different TSP algorithms.

Finally, the approach will be realized as an information system as a client-server-application. The mobile client suits industrial need. The server provides in addition an interface to the CAM system CATIA V5. A 5-axis machining centre will be completed the experimental setup. In this current development status of the information system the user has access to the product data of the CAD module and planning data of the CAM module. Thus the first feedback from the shop floor will be enabled. The user can directly modify parameters of the process planning of turning and milling tasks in the actual implementation status.

6 CONCLUSIONS AND OUTLOOK

Nowadays, there are significant deficiencies in the information flow along the NC process chain. The reasons are the used insufficient interfaces within the process chain. Consequently, knowledge is kept in the uncoupled process chain. The deficits could be solved insufficiently with the help of common integrated CAD/CAM software solutions. Modifications in the shop-floor and the handling of alternative machining strategies are not supported in an adequate way. The presented approach for knowledge-based NC program is an enabler to optimize the machining of different tasks under given scenario. The fundamental concept will be the use of feature-based designing and process planning. Therefore, a mathematical description of machining will be introduced. This specification describes a work plan as a sequence of machining features with alternative machining steps. These MStep are benchmarked with the help of different certain criteria. The process planer has now the possibility to select different alternatives parameters for machining different areas of a workpiece regarding a given machining scenario. Furthermore a work plan structuring and re-ordering approach will support the process planer. For processing a work plan, different mathematical descriptions were investigated and assessed. Finally, the MStep-based work plan will be transferred in a directed graph with nodes as MStep and arcs as quantified dependences within them. The application of the Floyd-Warshall algorithm calculates now the best choice of alternative MStep for each machining feature. Furthermore heuristics algorithms for the TSP calculate the optimal sequence of the received MStep in a work plan. Afterwards, the optimal work plan will be transferred to the NC path generator of the CAM system. A first prototype is implemented step by step. First results validate the approach. The next steps will be the enhancement of the prototype in order to add additional databases to store the MStep and the structured machinery. Furthermore, the user profiles have to be defined and implemented instantly. An algorithm has to be defined, which suggests possible set-ups for given structured machining tasks with methods known from the AI (e.g. CBR). The heuristic TSP algorithm has to be implemented. Therefore the ANN will be modified and adapted to the NC process planning. In conclusion the approach supports the process planner to select the suitable process operations corresponding to a given scenario definition.

ACKNOWLEDGEMENTS

The work reported in this paper was partially supported by EC / FP6 Program, “Development of a Hybrid Machine Tool Concept for Manufacturing of Free-form Surface Moulds” (FP6-2004-NMP-NI-4-026621-2).

REFERENCES

Berger, U., Cai, J., Weyrich, M., 2005. Ontological Machining Process Data Modelling for Powertrain Production in Extended Enterprise. *Journal of Advanced Manufacturing System (JAMS)*, Vol. 4, No. 1 (2005), pp. 69-82.

Berger, U., Kretzschmann, R., Aner, M., 2007. Development of a holistic guidance system for the NC process chain for benchmarking machining operations. In: *Proceedings of the 12th IEEE Conference on Emerging Technologies and Factory Automation*, Greece, September 25-28, 2007.

DIN 66025, 1983. *Programmaufbau für numerisch gesteuerte Arbeitsmaschinen*.

Eversheim W., 1996. *Organisation in der Produktionstechnik - Band 1 Grundlagen*, VDI-Verlag, Düsseldorf.

Gerken H., 2000. *Management von Erfahrungen mit einem Assistenzsystem für die Arbeitsplanung*, TU-Berlin.

Görz, G., 2003. *Handbuch der Künstlichen Intelligenz*, Oldenbourg, München.

Hamelmann, S., 1995. *Systementwicklung zur Automatisierung der Arbeitsplanung*, VDI-Verlag Aachen.

Jungnickel, D., 1990. *Graphen, Netzwerke und Algorithmen*, Wissenschaftsverlag, Mannheim.

Pritschow G., Heusinger S., 2005, *STEP-NC-basierter Korrekturkreis für die Schlichtbearbeitung von Freiformflächen*, Jost-Jetter Verlag, Heimsheim.

VDI Richtlinie - VDI 2218, 2003. *Informations-verarbeitung in der Produktentwicklung – Feature-Technologie.*

Warnecke G., Valous A., 1993. *Informationsrückkopplung zwischen NC-Fertigung und Arbeitsplanung*, Schnelldruck Ernst Grässer, Karlsruhe.

Weck M., Wolf J., Kiritsis J, D., 2001. STEP-NC – The STEP compliant NC Programming Interface, *IMS Forum*, Ascona Schweiz.

LIGHT-WEIGHT REINFORCEMENT LEARNING WITH FUNCTION APPROXIMATION FOR REAL-LIFE CONTROL TASKS

Kary Främling
Helsinki University of Technology, PL 5500, FI-02015, Finland
Kary.Framling@hut.fi

Keywords: Reinforcement learning, function approximation, normalised radial basis function network, eligibility trace, mountain-car, cart-pole, pendulum.

Abstract: Despite the impressive achievements of reinforcement learning (RL) in playing Backgammon already in the beginning of the 90's, relatively few successful real-world applications of RL have been reported since then. This could be due to the tendency of RL research to focus on discrete Markov Decision Processes that make it difficult to handle tasks with continuous-valued features. Another reason could be a tendency to develop continuously more complex mathematical RL models that are difficult to implement and operate. Both of these issues are addressed in this paper by using the gradient-descent Sarsa(λ) method together with a Normalised Radial Basis Function neural net. The experimental results on three typical benchmark control tasks show that these methods outperform most previously reported results on these tasks, while remaining computationally feasible to implement even as embedded software. Therefore the presented results can serve as a reference both regarding learning performance and computational applicability of RL for real-life applications.

1 INTRODUCTION

A Reinforcement Learning (RL) agent can sense the state of itself and its environment, take actions that may change that state in some way and improve its behaviour based on a reward signal that expresses if one or several actions (or their result) was good or bad and to what extent. RL is therefore similar to the way that most animals learn to handle new or changing situations, e.g. babies learning how to walk and grab things, children learning how to ride a bicycle or adults learning how to drive a car. Despite this relationship with human and animal learning in the real world, it is surprising how few successful uses of RL for real-world control applications have been reported. The most well-known success of RL so far might be the TD-Gammon system (Tesauro, 1995) that learned to play Backgammon on world-champion level. However, Backgammon is not a control task and the methods and results are not easy to extrapolate to control tasks. A recent success on a real-world RL control task is the helicopter flight control reported in (Abbeel et al., 2007). What is common to both of these successful applications of RL (especially in helicopter flight) is that they are the result of rather complex calculations that involve several different learning phases and extensive hand-tuning by human designers. Even in experiments with standard control tasks such as those used in this paper, the RL methods employed tend to be complex and require extensive hand-tuning by human designers in order to make the learning task feasible. This complexity makes it difficult to re-use successful RL models in new application areas. Such models also tend to be too complex to provide plausible models of animal learning.

Both TD-Gammon and the helicopter controller use function approximation in order to be able to handle the huge state spaces involved. Especially continuous-valued control tasks tend to be problematic for RL because of their infinite state space. Many popular RL methods assume that the task to learn can be represented as a Markov Decision Process (MDP). In tasks with continuous-valued inputs it becomes necessary to discretize the state space in order to handle it as an MDP but the discretisation easily leads to an explosion of the state space. This contradic-

tion between 'provably converging' but 'hard-to-use in real-life' MDP-based methods may also be a reason for the lack of reported successful uses of RL in real-world control tasks.

In this paper we will show how function approximation and one of the 'lightest' RL methods, i.e. model-free action-value learning with gradient-descent Sarsa(λ), can be used for successful learning of three benchmark control tasks. Less hand-tuning is needed and learning is significantly faster than in previously reported experiments. These small memory and computation requirements and the rapid learning should make it possible to use RL even in embedded systems that could adapt to their environment. These characteristics also make the learning biologically plausible. The experimental tasks used are the *Mountain-Car, Swinging up pendulum with limited torque* and *Cart-Pole* tasks.

After this introduction, section 2 describes the background and theory of gradient-descent Sarsa(λ), then section 3 presents the Normalized Radial Basis Function approximation technique used, followed by experimental results in section 4 and conclusions.

2 ACTION-VALUE LEARNING WITH FUNCTION APPROXIMATION

Action-value learning is needed in control tasks where no model of the controlled system is available. In the case of continuous-valued function approximation, *gradient-descent Sarsa(λ)* (Sutton and Barto, 1998, p. 211) can be used:

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \alpha\left[r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)\right]\vec{e}_t, \tag{1}$$

where $\vec{\theta}_t$ is the parameter vector of the function approximator, r_{t+1} is the reward received upon entering a new state, α is a learning rate, γ is the discount rate and $Q_t(s_t, a_t)$ and $Q_t(s_{t+1}, a_{t+1})$ are the action-value estimates for the current and next state, respectively. In the case of an accumulating trace, the trace $\vec{e}_t$ is updated according to:

$$\vec{e}_t = \gamma\lambda\vec{e}_{t-1} + \nabla_{\vec{\theta}_t} Q_t(s_t, a_t) \tag{2}$$

with $\vec{e}_0 = \vec{0}$. $\nabla_{\vec{\theta}_t} f\left(\vec{\theta}_t\right)$, for any function f denotes the vector of partial derivatives (Sutton and Barto, 1998, p. 197):

$$\left(\frac{\partial f\left(\vec{\theta}_t\right)}{\partial\theta_t(1)}, \frac{\partial f\left(\vec{\theta}_t\right)}{\partial\theta_t(2)}, \ldots, \frac{\partial f\left(\vec{\theta}_t\right)}{\partial\theta_t(n)}\right)^T, \tag{3}$$

where $\vec{\phi}$ is *feature vector* that consist of the state variables of the task (continuous-valued or not). Eligibility traces can speed up learning significantly by improving temporal credit assignment. They are inspired from the behaviour of biological neurons that reach maximum eligibility for learning a short time after their activation and were mentioned in the context of Machine Learning at least as early as 1972 and used for action-value learning at least as early as 1983 (Barto et al., 1983), where the accumulating eligibility trace for discrete state tasks (MDP) was proposed:

$$e_t(s,a) = \begin{cases} \gamma\lambda e_{t-1}(s,a) + 1 & \texttt{if } s = s_t \texttt{ and } a = a_t; \\ \gamma\lambda e_{t-1}(s,a) & \texttt{otherwise} \end{cases} \tag{4}$$

for all s, a. λ is a trace decay parameter. s corresponds to the feature vector $\vec{\phi}$ in 3 but is restricted to binary values. Equation 4 is a special case of equation 2 when using binary state representations because then $\nabla_{\vec{\theta}_t} Q_t(s_t, a_t) = 1$ when $s = s_t$ and $a = a_t$ and zero otherwise. In (Singh and Sutton, 1996) it was proposed to use a replacing eligibility trace instead of the accumulating eligibility trace:

$$e_t(s,a) = \begin{cases} 1 & \texttt{if } s = s_t \texttt{ and } a = a_t; \\ 0 & \texttt{if } s = s_t \texttt{ and } a \neq a_t; \\ \gamma\lambda e_{t-1}(s,a) & \texttt{if } s \neq s_t \end{cases} \tag{5}$$

The replacing eligibility trace outperformed the accumulating eligibility trace in the Mountain-Car task as reported in (Singh and Sutton, 1996). This is the main reason why replacing traces are considered to perform better than accumulating traces. In (Främling, 2007b) a generalisation was proposed for the replacing trace that makes it possible to use with continuous-valued function approximation:

$$\vec{e}_t = \max(\gamma\lambda\vec{e}_{t-1}, \nabla_{\vec{\theta}_t} Q_t(s_t, a_t)) \tag{6}$$

As for the accumulating trace, equation 5 is identical to equation 6 in the case of binary features, except for resetting the trace of unused actions of the current state to zero in equation 5. Even though the results in (Främling, 2007b) seem to confirm the advantage of replacing traces, especially when not resetting the traces of unused actions, there is still a lack of evidence on whether this is also true for other tasks. The experiments in this paper are performed using both kinds of eligibility traces as an attempt to provide increased insight into this issue.

3 NORMALISED RADIAL BASIS FUNCTION NETWORK APPLIED TO ACTION-VALUE LEARNING

When function approximation with neural nets is used for learning an action-value function, an eligibility trace value is associated with every weight of the neural net (Barto et al., 1983). Then the eligibility trace value reflects to what extent and how long ago the neurons connected by the weight have been activated. The use of function approximation rather than binary lookup-tables signifies that more than one feature value ϕ can be different from zero. This is also the case with the binary CMAC (Cerebellar Model Articulation Controller) neural network (Albus, 1975) used in (Singh and Sutton, 1996). When using CMAC, there are as many active features as there are overlapping layers in the CMAC, which could lead to excessive weight changes in Equation 1. In order to avoid this, (Singh and Sutton, 1996) divide every value $e_t(s,a)$ by the sum of all feature values in order to avoid learning divergence as a result of excessive weight changes. No justification was given for this operation in (Singh and Sutton, 1996) but there seems to be a connection with the well-known Normalised Least Mean Squares (NLMS) method, where α in Equation 1 is replaced by α_{norm}:

$$\alpha_{norm} = \alpha / \sum \vec{e}_t. \tag{7}$$

The use of NLMS for RL in a real-world robotics task has been studied e.g. in (Främling, 2004) and (Främling, 2005). When using continuous-valued function approximation, most terms $\nabla_{\vec{\theta}_t} Q_t(s_t,a_t)$ in equation 2 will be different both from zero and one, which is the main source of incompatibility between the discrete eligibility traces (Equations 4 and 5) and their continuous-valued counterparts (Equations 2 and 6). In order to avoid weight divergence in equation 1, it can be assumed that an NLMS-type normalisation would be useful also with continuous-valued features. Because NLMS does not seem to have been used in neural nets, another kind of normalisation than NLMS is used here, i.e. the Normalised Radial Basis Function network (NRBF).

The continuous feature values are calculated by an RBF network as:

$$\phi_i(\vec{s}_t) = \exp\left(-\frac{(\vec{s}_t - \vec{c})^2}{r^2}\right), \tag{8}$$

where $\vec{c}$ is the *centroid* vector of the RBF neuron and r^2 is the *spread* parameter. The action-value estimate of action a is:

$$Q_t(\vec{s}_t, a) = \sum_{i=1}^{N} w_{ia} \left(\phi_i / \sum_{k=1}^{N} \phi_k \right), \tag{9}$$

where w_{ia} is the weight between the 'action' neuron a and the RBF neuron i. The division by $\sum_{k=1}^{N} \phi_k$, where N is the number of RBF neurons, is the normalisation step that ensures that the sum of all ϕ will be exactly one. Every parameter w_{ia}, has its corresponding eligibility trace value. For the accumulating trace, equation 2 was used in all experiments, while equation 6 was used for the replacing trace. In all experiments, an affine transformation mapped the actual state values from the respective value ranges into the range [0,1] that was used as $\vec{s}$. This transformation makes it easier to adjust the spread parameter ($r^2 = 0.01$ is used in all experiments).

4 EXPERIMENTS

The Mountain-Car task (Figure 1) was used in (Singh and Sutton, 1996) and the results presented there seem to be the main reason why replacing eligibility traces are considered to perform better than accumulating traces. The second task consists in swinging up a pendulum with limited torque and keeping it in upright position (Figure 4). The pendulum task is surprisingly difficult for ordinary learning methods such as Sarsa(λ) with binary representations (lookup-table, CMAC) due to the non-linear control function required for passing from swing-up behaviour to balance-keeping behaviour. For the Cart-Pole task (Figure 7), successful results were reported using actor-critic and lookup-tables already in (Barto et al., 1983).

For each task, results are first shown by graphs that allow comparison between results shown here and those reported in existing literature. A supplementary graph may be used to show the results in a homogeneous way between the three tasks that allows to evaluate both learning speed and final performance after learning, as well as how well the accumulating and the replacing traces perform in each task. These graphs are also intended to provide a uniform benchmark framework for future experiments using the same tasks.

4.1 Mountain-Car

One of the first presentations of this task seems to be in (Moore, 1991), where a model of the environment was learned. The task description used here is identical to the one in (Singh and Sutton, 1996). The task

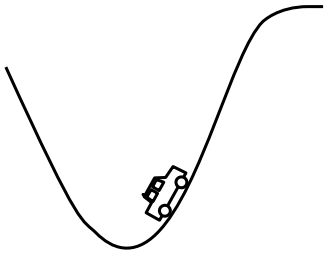


Figure 1: Mountain-Car task.

consists in accelerating an under-powered car up a hill, which is only possible if the car first gains enough inertia by backing away from the hill. The task has a continuous-valued state vector $\vec{s}_t = (x_t, v_t)$, i.e. the position x_t and the velocity v_t. At the beginning of each trial these are initialized randomly, uniformly from the range $x \in [-1.2, 0.5]$ and $v \in [-0.07, 0.07]$. The altitude is $\sin(3x)$. 8×8 RBF neurons were used with regularly spaced centroids. The agent chooses from three actions $\{+1, 0, -1\}$ corresponding to forward thrust, no thrust and reverse. The physics of the task are:

$$\begin{aligned} v_{t+1} &= \mathsf{bound}\left(v_t + 0.001a_t + g\cos(3x_t)\right) \\ x_{t+1} &= \max\left\{x_t + v_{t+1}, -1.2\right\} \end{aligned},$$

where $g = 0.0025$ is the force of gravity and the bound operation places the variable within its allowed range. If x_{t+1} is clipped by the max-operator, then v_{t+1} is reset to 0. The terminal state is any position with $x_{t+1} > 0.5$. The reward function is the *cost-to-go* function as in (Singh and Sutton, 1996), i.e. giving -1 reward for every step except when reaching the goal, where zero reward is given. The weights of the neural net were set to zero before every new run so the initial action-value estimates are zero. The discount rate γ was one.

Figure 2 shows the average number of steps per episode for the best α values (4.9, 4.3, 4.3, 4.7, 3.9, 3.3 and 2.5) as a function of λ, where the numbers are averaged for 30 agent runs with 20 episodes each as in (Singh and Sutton, 1996). For some reason, the results reported here indicate 50% less steps than in most other papers (except e.g. (Smart and Kaelbling, 2000) who have similar numbers as here), which should be taken into consideration when comparing results. Taking this difference into consideration, the CMAC results shown in Figure 2 are consistent with those in (Singh and Sutton, 1996). The results using gradient-descent Sarsa(λ) and NRBF are significantly better than those obtained with CMAC.

Figure 3 shows the average number of steps per episode for the best-performing agent with $\alpha = 3.9$ and $\lambda = 0.9$. Convergence to an average of 60 steps

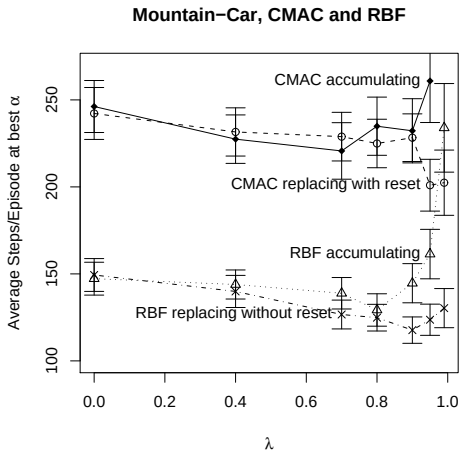


Figure 2: Average number of steps per episode for 30 agents running 20 episodes as in (Singh and Sutton, 1996).

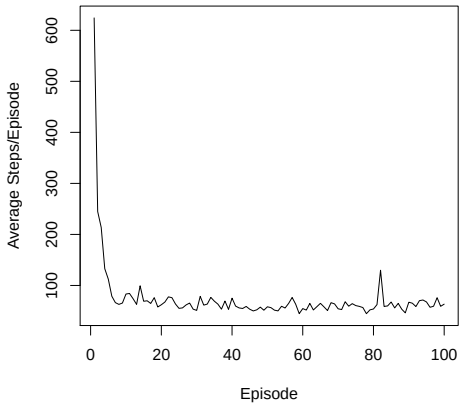


Figure 3: Average number of steps per episode for best-performing agent (replacing trace, $\alpha = 3.9$ and $\lambda = 0.9$).

per episode is achieved after about 40 episodes. This is better than the 'less than 100 episodes' indicated in (Abramson et al., 2003). The optimal number of steps indicated in (Smart and Kaelbling, 2000) is 56 steps so 60 steps per episode can be considered optimal due to the continuing exploration caused by the -1 reward on every step. Therefore the converged learning result is also as good as in (Abramson et al., 2003) and (Schaefer et al., 2007). (Abramson et al., 2003) used Learning Vector Quantization together with Sarsa instead of using NRBF so the complexity of the learning algorithm is similar to the one used in this paper. The reward function provided more guidance towards the goal than the one used here and a fixed starting point was used instead of a random one. The neural net used in (Schaefer et al., 2007) is more complex than the one used here while a reward of +1 was given at

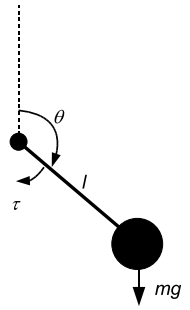


Figure 4: Pendulum swing-up with limited torque task.

the goal and zero otherwise. (Smart and Kaelbling, 2000) use an RBF-like neural network together with Sarsa but no eligibility traces and a reward that is inversely proportional to the velocity at goal (one if zero velocity) and zero otherwise. Results are indicated as a function of training steps rather than episodes but a rough estimate would indicate that convergence occurs after over 2000 episodes. The experimental settings and the ways in which results are reported in (Strens and Moore, 2002) do not make it possible to compare results in a coherent way. In (Mahadevan and Maggioni, 2007) convergence happens after about 100 episodes, which is much slower than in Figure 3 despite the use of much more mathematically and computationally complex methods. The results in (Främling, 2007a) are clearly superior to all others but they are obtained by having a parallel, hand-coded controller guiding the initial exploration, which could also be combined with gradient-descent Sarsa(λ) and NRBF. These differences make it difficult to compare results but the results shown in this paper are systematically better than comparable results reported in literature.

4.2 Pendulum Swing-up with Limited Torque

Swinging up a pendulum to the upright position and keeping it there despite an under-powered motor is a task that has been used e.g in (Doya, 2000), (Schaal, 1997) and (Santamaría et al., 1998). The task description used here is similar to the one in (Doya, 2000). There are two continuous state variables: the angle θ and the angular speed $\dot{\theta}$. The agent chooses from the two actions ± 5Nm corresponding to clockwise and anti-clockwise torque. The discount rate γ was set to one. The system dynamics are defined as follows:

$$ml^2\ddot{\theta} = -\mu\dot{\theta} + mgl\sin\theta + \tau, \theta \in [-\pi, \pi]$$
$$m = l = 1, g = 9.81, \mu = 0.01, \tau_{max} = 5Nm$$

At the beginning of each episode θ is initialized to a random value from $[-\pi, \pi]$ and $\dot{\theta} = 0$. Every episode lasted for 60 seconds with a time step $\Delta_t = 0.02$ using Euler's method. The performance measure used was the time that $|\theta| \leq \frac{pi}{2}$. The reward was $r_1 = \cos\theta - 0.7$, except when $|\theta| \leq \frac{pi}{5}$ where it was $r_2 = r_1 - |\dot{\theta}|$. Such a reward function encourages exploration when the pendulum cannot be taken directly to the upright position by the available torque. It also provides some guidance to that zero is the ideal speed in the upright position. This shaping reward remains less explicit than the one used e.g. in (Schaal, 1997) and (Santamaría et al., 1998). This learning task is more difficult than in (Schaal, 1997) and (Doya, 2000) where a model of the system dynamics either has been used directly or learned by a model. 10×10 NRBF neurons were used with regularly spaced centroids. An affine transformation mapped the actual state values from the intervals $[-\pi, \pi]$ and $[-1.5, 1.5]$ (however, angular speeds outside this interval are also allowed) into the interval [0,1] that was used as $\vec{s}$.

Figure 5 shows the time that the pendulum remains in the upright position for the best-performing agent (replacing trace, $\alpha = 1.5$ and $\lambda = 0.4$). An uptime over 40 seconds signifies that the pendulum is successfully balanced until the end, which occurs after about 10 episodes. In (Doya, 2000) a continuous actor-critic model was used that is computationally more complex than what is used in this paper. A model-free agent in (Doya, 2000) started performing successfully after about 50 episodes but never achieved a stable performance. The best results were achieved by an agent with a complete model of the system dynamics, which performed successfully after about 10 episodes. However, such model-based results should not be compared with model-free methods like the one used in this paper. In (Schaal, 1997) a model-based TD(λ) method was used that performed successfully after about 100 episodes, so despite the model-based approach the results are not as good as those in Figure 5. Gradient-descent Sarsa(λ) is used together with RBF networks in (Santamaría et al., 1998) also for the case of continuous actions. However, they used a reward function that guided learning much more than the one used here. Their instance-based agent that is computationally on the same level as the NRBF net used here did not manage to learn successfully, while the more complex case-based agent did so. The case-based agents managed to balance the pendulum after about 30 episodes but due to a cut-off in the top of the result-showing graphs it is difficult to assess the performance after that. In any case, the results shown in this paper are systematically better than those compared with here, while remaining simpler to use and computationally lighter.

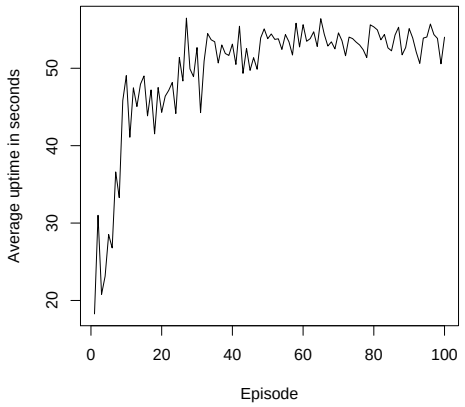


Figure 5: Average up-time for the best-performing agent in seconds as a function of episode for pendulum task.

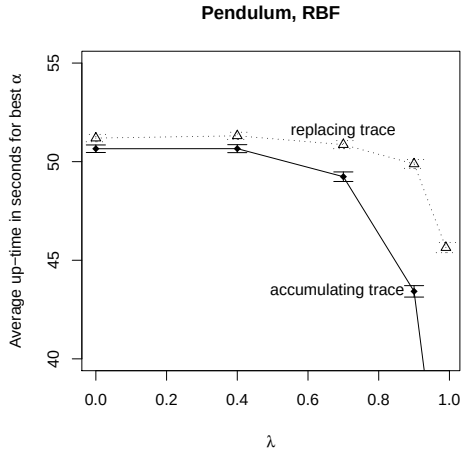


Figure 6: Average up-time in seconds for pendulum task with best α value as a function of λ. Error bars indicate one standard error.

As shown in Figure 6, the replacing trace consistently gives slightly better performance than the accumulating trace for this task. Learning results are also much more sensitive to the value of the learning rate for the accumulating trace than for the replacing trace.

4.3 Cart-Pole

One of the first uses of RL for the Cart-Pole task was reported in (Barto et al., 1983). It is unknown whether successful learning has been achieved with model-free Sarsa(λ) on this task. An actor-critic architecture was used in (Barto et al., 1983), (Kimura and Kobayashi, 1998) and (Schneegaß et al., 2007). Memory-based architectures were used in (Whitehead and Lin, 1995) while (Schaal, 1997) uses initial supervised learning for six episodes in order to provide an approximately correct initial action-value function. Some kind of reward shaping is also employed in order to simplify the learning task.

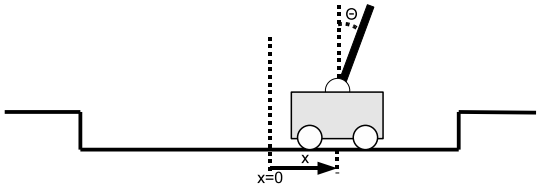


Figure 7: Cart-Pole task.

The task description used here is from (Barto et al., 1983). There are four continuous state variables: the angle θ and angular speed $\dot{\theta}$ of the pole and the position x and velocity $\dot{x}$ of the cart. The agent chooses from the two actions $\pm 10\mathrm{N}$ corresponding to right or left acceleration. The discount rate γ was set to one and ε-greedy exploration with $\varepsilon = 0.1$ was used. The system dynamics are:

$$\ddot{\theta} = \frac{g\sin\theta_t + \cos\theta_t \left[\frac{-F_t - ml\dot{\theta}_t^2 \sin\theta_t + \mu_c \mathrm{sgn}(\dot{x}_t)}{m_c+m}\right] - \frac{\mu_p \dot{\theta}_t}{ml}}{l\left[\frac{4}{3} - \frac{m\cos^2\theta_t}{m_c+m}\right]}$$

$$\ddot{x}_t = \frac{F_t + ml\left[\dot{\theta}_l^2 \sin\theta_t - \ddot{\theta}_t \cos\theta_t\right] - \mu_c \mathrm{sgn}(\dot{x}_t)}{m_c+m}$$

$$m_c = 1.0\mathrm{kg}, m = 0.1\mathrm{kg}, l = 0.5\mathrm{m}, g = 9.81 m/s^2,$$
$$\mu_c = 0.0005, \mu_p = 0.000002, F_t = \pm 10\mathrm{N}$$

Every episode started with $\theta = 0$, $\dot{\theta} = 0$, $x = 0$, $\dot{x} = 0$ and lasted for 240 seconds or until the pole tipped over ($\theta > 12^\circ$) with a time step $\Delta_t = 0.02$ using Euler's method. The reward was zero all the time except when the pole tipped over, where a -1 'reward' was given. The NRBF network used $6 \times 6 \times 6 \times 6$ RBF neurons with regularly spaced centroids. An affine transformation mapped the actual state values into the interval [0,1] as for the other tasks.

Table 1 indicates approximate numbers for how many episodes were required before most agents have succesfully learned to balance the pole. Comparison between the results presented in different sources is particularly difficult for this task due to the great variation between the worst and the best performing agents. Anyway, the results presented here are at least as good as in the literature listed here despite the use of conceptually and computationally significantly simpler methods in this paper. The only exception is (Mahadevan and Maggioni, 2007) that performs well in this task, most probably due to a considerably more efficient use of kernels in the parts of

Table 1: Approximate average episode when successful balancing occurs in different sources, indicated where available.

This paper	100
(Barto et al., 1983)	80
(Whitehead and Lin, 1995)	200
(Kimura and Kobayashi, 1998)	120
(Schneegaß et al., 2007)	100-150
(Lagoudakis and Parr, 2003)	100
(Mahadevan and Maggioni, 2007)	20

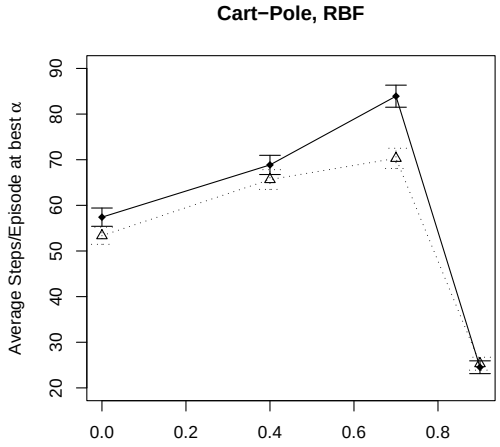


Figure 8: Average up-time in seconds for Cart-Pole task with best α value as a function of λ. Error bars indicate one standard error.

the state space that are actually visited, rather than the regularly spaced $6 \times 6 \times 6 \times 6$ RBF neurons used here, of which a majority is probably never used. However, the kernel allocation in (Mahadevan and Maggioni, 2007) requires initial random walks in order to obtain samples from the task that allow offline calculation of corresponding Eigen vectors and allocating 'Proto-Value Functions' from them.

A challenge in interpreting the results of the experiments performed for this paper is that the agent 'un-learns' after a certain number of successful episodes where no reward signal at all is received. As pointed out in (Barto et al., 1983), this lack of reward input slowly deteriorates the already learned action-value function in critic-only architectures such as Sarsa(λ). The results shown in Figure 8 include this phenomenon, i.e. the average values are calculated for 200 episodes during which the task is successfully learned but also possibly un-learned and re-learned again.

The results shown in Figure 8 differ from those of the previous tasks by the fact that the accumulating trace performs slightly better than the replacing trace and that the parameter sensitivity is quite similar for both trace types in this task. This difference compared to the two previous tasks is probably due to the delayed (and rare) reward that allows the eligibility trace to decline before receiving reward, thus avoiding weight divergence in equation 1. Further empirical results from tasks with delayed reward are still needed before declaring which eligibility trace type is better.

5 CONCLUSIONS

Gradient-descent Sarsa(λ) is not a new method for action-value learning with continuous-valued function approximation. Therefore it is surprising how little empirical knowledge exists that would allow practitioners to assess its usability compared with more complex RL methods. The only paper of those cited here that uses gradient-descent Sarsa(λ) is (Santamaría et al., 1998). The results obtained in this paper show that at least for the three well-known benchmark tests used, gradient-descent Sarsa(λ) together with NRBF function approximation tends to outperform all other methods while being the conceptually and computationally most simple-to-use method. The results presented here should provide a useful benchmark for future experiments because they seem to outperform most previously reported results, even those where a previously collected training set or a model of the system dynamics was used directly or learned.

These results are obtained due to the linking of different pieces together in an operational way. Important elements are for instance the choice of using NRBF and the normalisation of state values into the interval [0,1] that simplifies finding suitable learning rates and r^2 values. The replacing eligibility trace presented in (Främling, 2007b) is also more stable against learning parameter variations than the accumulating trace while giving slightly better performance. Future directions of this work could consists in performing experiments with more challenging tasks in order to see the limits of pure gradient-descent Sarsa(λ) learning. In tasks where gradient-descent Sarsa(λ) is not sufficient, combining it with pre-existing knowledge as in (Främling, 2007a) could be a solution for applying RL to new real-world control tasks.

REFERENCES

Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. In Schölkopf, B., Platt, J., and Hoff-

man, T., editors, *Advances in Neural Information Processing Systems 19*, pages 1–8, Cambridge, MA. MIT Press.

Abramson, M., Pachowicz, P., and Wechsler, H. (2003). Competitive reinforcement learning in continuous control tasks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN), Portland, OR*, volume 3, pages 1909–1914.

Albus, J. S. (1975). Data storage in the cerebellar model articulation controller (cmac). *Journal of Dynamic Systems, Measurement and Control*, September:228–233.

Barto, A., Sutton, R., and Anderson, C. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. on Systems, Man, and Cybernetics*, 13:835–846.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computation*, 12:219–245.

Främling, K. (2004). Scaled gradient descent learning rate - reinforcement learning with light-seeking robot. In *Proceedings of ICINCO'2004 conference, 25-28 August 2004, Setubal, Spain*, pages 3–11.

Främling, K. (2005). Adaptive robot learning in a non-stationary environment. In *Proceedings of the* 13^{th} *European Symposium on Artificial Neural Networks, April 27-29, Bruges, Belgium*, pages 381–386.

Främling, K. (2007a). Guiding exploration by pre-existing knowledge without modifying reward. *Neural Networks*, 20:736–747.

Främling, K. (2007b). Replacing eligibility trace for action-value learning with function approximation. In *Proceedings of the* 15^{th} *European Symposium on Artificial Neural Networks, April 25-27, Bruges, Belgium*, pages 313–318.

Kimura, H. and Kobayashi, S. (1998). An analysis of actor/critic algorithms using eligibility traces: Reinforcement learning with imperfect value functions. In *Proceedings of the* 15^{th} *Int. Conf. on Machine Learning*, pages 278–286.

Lagoudakis, M. G. and Parr, R. (2003). Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149.

Mahadevan, S. and Maggioni, M. (2007). Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *J. Mach. Learn. Res.*, 8:2169–2231.

Moore, A. (1991). Variable resolution dynamic programming. efficiently learning action maps in multivariate real-valued state-spaces. In *Machine Learning: Proceedings of the Eight International Workshop, San Mateo, CA.*, pages 333–337. Morgan-Kaufmann.

Santamaría, J., Sutton, R., and Ram, A. (1998). Experiments with reinforcement learning in problems with continuous state and action spaces. *Adaptive Behavior*, 6:163–217.

Schaal, S. (1997). Learning from demonstration. In *Advances in Neural Information Processing Systems (NIPS)*, volume 9, pages 1040–1046. MIT Press.

Schaefer, A. M., Udluft, S., and Zimmermann, H.-G. (2007). The recurrent control neural network. In *Proceedings of* 15^{th} *European Symposium on Artificial Neural Networks, Bruges, Belgium, 25-27 April 2007*, pages 319–324. D-Side.

Schneegaß, D., Udluft, S., and Martinetz, T. (2007). Neural rewards regression for near-optimal policy identification in markovian and partial observable environments. In *Proceedings of* 15^{th} *European Symposium on Artificial Neural Networks, Bruges, Belgium, 25-27 April 2007*, pages 301–306. D-Side.

Singh, S. and Sutton, R. (1996). Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22:123–158.

Smart, W. D. and Kaelbling, L. P. (2000). Practical reinforcement learning in continuous spaces. In *Proceedings of the Seventeenth* 17^{th} *International Conference on Machine Learning*, pages 903–910. Morgan Kaufmann.

Strens, M. J. and Moore, A. W. (2002). Policy search using paired comparisons. *Journal of Machine Learning Research*, 3:921–950.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning*. MIT Press, Cambridge, MA.

Tesauro, G. (1995). Temporal difference learning and td-gammon. *Communications of the ACM*, 38:58–68.

Whitehead, S. and Lin, L.-J. (1995). Reinforcement learning of non-markov decision processes. *Artificial Intelligence*, 73:271–306.

RFID BASED LOCATION IN CLOSED ROOMS

Implementation of a Location Algorithm using a Passive UHF-RFID System

Christoph Schönegger, Burkhard Stadlmann
University of Appled Sciences Upper Austria, Stelzhamerstr. 23, 4600 Wels, Austria
christoph.schoeneggert@fh-wels.at, burkhard.stadlmann@fh-wels.at

Michael E. Wernle
Meshed Systems GmbH, Alte Landstrasse 21, 85521 Ottobrunn, Germany
michael.e.wernle@meshedsystems.com

Keywords: Passive RFID, Location, RFID-standard-components, Positioning system, RSSI value.

Abstract: This paper presents a new concept for determining the location of an RFID-tag without any additional hardware. For this positioning system standard RFID components with passive RFID-tags within the UHF range are used. The measurement is based on a location algorithm which makes use of the RSSI value of the UHF reader. The RSSI value is the return signal strength indicator and, as it is shown in the paper in hand, this signal correlates to the distance between the RFID tag and the antenna of the reader. This positioning system is especially useful indoors, where other positioning systems may not work. For this reason it could prove very useful in various logistics applications. The maximum distance from antenna to the tag is approximately between 0.5 m and 3 m. To this end a special algorithm is used to obtain stable calculation results. A minimum of two antennas is needed to get a two-dimensional location.

1 INTRODUCTION

Identification using RFID (radio frequency identification device) is more or less standard in many industrial applications and in many logistics processes. There are a variety of applications where the combination of identification and location is very useful.

This paper presents a solution for the location of UHF-RFID tags within the range of a reader antenna. For this positioning system only standard RFID equipment is used. The concept, the algorithm, and known limitations of the system are presented.

2 THE BASIC IDEA

The basic concept of the positioning system is to measure the distance between an RFID antenna and the RFID tag using only standard RFID equipment. The measurement of the distance is done by interpreting the signal strength of the UHF signal. The proposed system is especially useful for indoor use.

2.1 Location Algorithms

There are a variety of different location methods proposed in literature and in practical use nowadays. Examples with RFID or WLAN can be found in (bekkali 2007, chon 2004, ekahau 2007, geroldt 2007, ibach 2005, lionel 2004, tomberge 2004, tsukiyama 2007). This chapter provides a comparison between these methods. The main focus is on determining location within buildings. This ability could be very useful for numerous applications in logistics.

2.1.1 Cell-of-Origin Concept

This type of positioning system makes use of an algorithm using a mobile tag and a fixed reader. Upon detecting a nearby reader, the tag determines its position to be nearly equal to that of the reader. For this the tag must be within the range of the corresponding reader.

This type of positioning system is frequently used with RFID systems in the 125 kHz and the 13.56 MHz range. Practical applications of this type can be found on AGV systems (autonomous guided vehicles) or mobile robots.

A commercially available example of this type of positioning system which uses WLAN or Bluetooth technology is the Ekahau Positioning Engine (ekahau 2007).

An "inverse cell-of-origin concept" is used if the RFID reader is on the mobile unit and different tags have a fixed position. If the reader can communicate with a specific tag, then the position of the mobile unit can be determined in relation to the position of the fixed tag. An example for this concept can be found in (tsukiyama 2007).

2.1.2 Triangulation Method

The location of a tag can be calculated by triangulation if the distance between the tag and several known reference stations can be determined. The measurement of the distance can be achieved by detecting the runtime of the radio signal or laser or by measuring the signal strength of the radio signal.

The best known positioning system of this type is the global positioning system (GPS). This method also finds use in RFID systems which use active tags and operate in the 2.45 GHz range.

2.2 RFID

An RFID system consists of two components which communicate via radio:

- a device called a "tag" or "transponder" which is capable of storing data
- a so called "reader" (providing read and write functionality) communicates with the tags using an appropriate antenna-system, controller and amplifier.

Nowadays tags are very cheap as they are produced in large numbers. Therefore they are widely used in many goods and logistics devices.

One reader can be equipped with several antennas. Within the range of one antenna multiple tags can be detected and communication is organised into a sequence of the different tags.

2.2.1 Types of RFID Systems

There are various types of RFID systems available. Different types operate at different frequencies, have different couplings and differ as well in the energy supply of the transponders.

The different frequencies are

- "low frequency" (119 … 148.5 kHz)
- "high frequency" (13.56 MHz)
- "ultra high frequency" (865 … 955 MHz)
- "microwave" (2.4 … 2.5 GHz)

The different coupling technologies are inductive coupling and modulated backscatter coupling.

Both the coupling method and the operating frequency influence the range of operation

- "close-coupling" – distance < 1 cm
- "remote-coupling" – distance < 1 m
- "long-range" – distance > 1 m

Tags can operate as passive tags, which get their power from the reader via the electromagnetic field of the antenna, or as active tags, which are powered by a remote battery or some other power supply.

2.2.2 Applications

As mentioned above logistics applications need a location algorithm to find a specific tag. Location, used in combination with RFID, has the advantage of combining identification and location using the same hardware.

One typical example is the localisation of parcels on a conveyor belt, persons walking through an RFID gate carrying a transponder or the localisation of a palette carried by a forklift. All these applications have the need for optimised performance, better process control and supervision. For proper use it is important to heed the basic limitations of this concept.

2.3 The RSSI-Value

The RSSI value (received signal strength indicator) is a commonly used value within radio communication systems. Modern RFID readers within the UHF frequency range have the ability to determine this value as a measure of the reflected UHF signal from the tag. That is where RSSI value gets its name "reflected signal strength indicator". one must consider that up to now this RSSI value has not been standardised and is therefore manufacturer dependent.

2.4 Used Technology

To achieve the goal of this location procedure, RFID technology is used which has a long distance range and is widely used in logistic systems. According to VDI 4472 the recommended frequency for logistics applications is 868 MHz (UHF range), which is indeed very commonly used. UHF systems have

long-range readability with passive tags and the reader normally has an output for the RSSI value.

Therefore a passive RFID system with a frequency of 868 MHz was chosen for this particular location system.

3 MEASURING DISTANCE USING THE RSSI VALUE

During the project a theoretical and practical analysis was undertaken to determine if the RSSI value is a well-working solution for measuring distance. Figure 1 shows the schematic design of the lab equipment for the practical investigations.

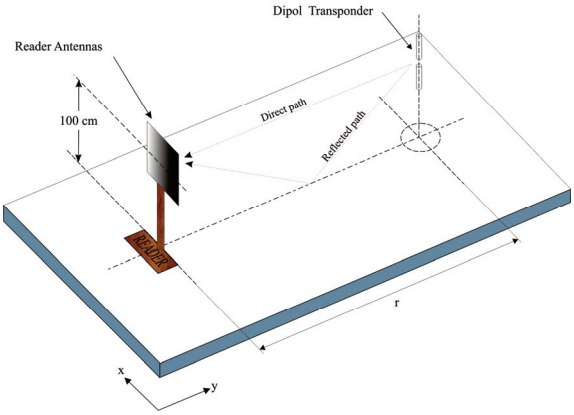


Figure 1: Schematic view of the lab equipment used for distance measuring.

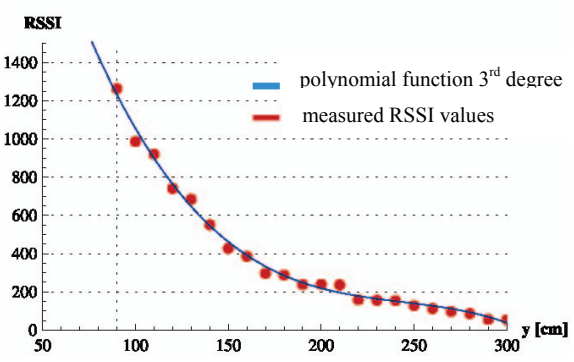


Figure 2: Measured RSSI-values varying only the y-coordinate of the tag-position.

For the practical measurements three different antennas and two different kinds of tags were used. Figure 2 and Figure 3 depict the mean value of the RSSI value of a 10dBi antenna. Figure 2 shows the variation of one coordinate and figure 3 the variation of two coordinates which leads to a three-dimensional radio-map. Each measured value is a mean value of 1000 datasets.

If the distance between the antenna of the reader and the tag is too low, the RSSI value may not be obtained due to saturation, which causes a heavy non-linear behaviour.

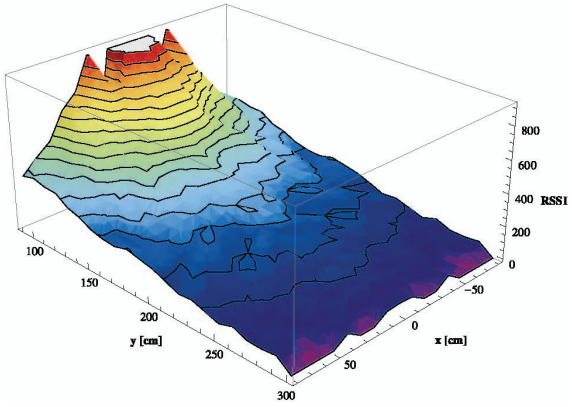


Figure 3: Three-dimensional radio map of the RSSI values, varying the x and the y coordinates of the tag position.

Three different types of antennas and two different types of tags were investigated (see Table 1 for the tags and Table 2 for the antennas of the reader). All antennas used have a circular right polarisation.

Table 1: Comparison of different types of tags.

	Philips	TI
Chip	Philips U-Code HSL	RI-UHF-00C02-04
Protocol	ISO 18000-6B	EPCUHF Gen 2
Antenna	Dipole λ/2	"dog bone"

Table 2: Comparison of different antennas of the reader.

	GP-ANTU	RH-ANTU	RH-ANTU
VSWR	< 1.3 : 1	< 1.5 : 1	< 1.5 : 1
Gain [dBi]	6	> 8.5	> 10
3 dB beamwidth horizontal	70°	63°	55°
Max. input power [W]	10	6	6

Due to the characteristics of the antenna on the tag, the best result was achieved using the combination of a λ/2-Dipole tag and the 10dBi antenna on the reader side.

Furthermore the influences of temperature were also investigated. This influence must be

compensated by adequate correction algorithms according to the measured temperature.

Additional influences are electromagnetic disturbances caused by fluorescent lamps and, of course, atmospheric humidity. As this system is proposed for indoor use humidity will not influence the system dramatically.

4 CALCULATION OF POSITION

In chapter 3 the strong correlation between the RSSI value and the distance between antenna and tag has been presented. To obtain the location of the tag, the results of more than one antenna have to be combined. During experiments in the project a two-dimensional location was tested. To determine this location at least two antennas are necessary which have contact to the same tag simultaneously. Knowing the RSSI value, it is possible to calculate a set of positions relative to the antenna where the tag might be based on the radio map.

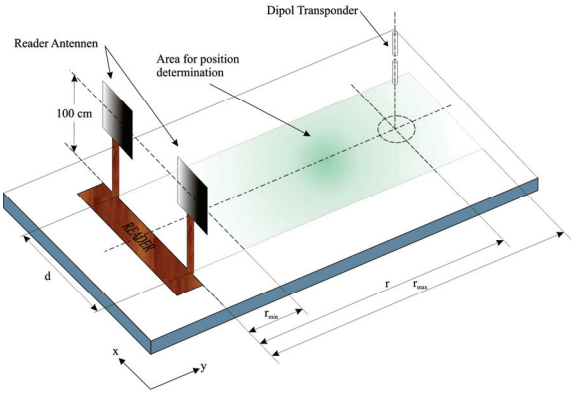


Figure 4: Schematic view of the enhanced lab equipment for location using two antennas.

Three different calculation algorithms have been investigated:

- Numeric iteration with finite differences
- Geometric intersection of polynomial approximations
- Weighted position determination

4.1 Numeric Iteration

This situation, where two antennas which communicate with the same tag is depicted in Figure 5. Antenna A has an RSSI value which can be located at the positions marked by red points (left curve in figure 5), antenna B has an RSSI value which can be located at the positions marked by blue points (right curve).

A scenic analysis (see also bahl 2000) yields the possible tag positions represented as a set of discrete points from each antenna in accordance with the measured RSSI value. The most probable tag position is the minimum distance between the possible locations of the two antennas within the overlapping area. If there is only a non-zero solution, the most probable position of the tag can be calculated by calculating the mean value. The Accuracy of this algorithm is, however, low.

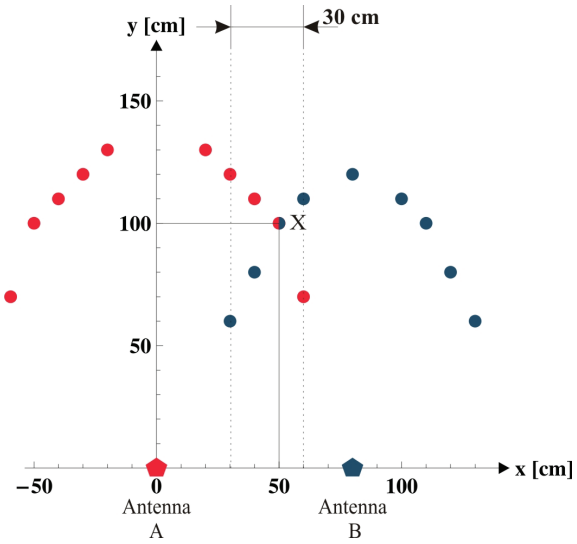


Figure 5: Scenario of two antennas for the algorithm "Numeric iteration".

4.2 Polynomial Approximation

This algorithm is based on a polynomial approximation of the line of constant RSSI value using a least square algorithm. For each antenna one polynomial exists for the measured RSSI value. Figure 6 depicts the appropriate situation.

The position of the tag is the intersection of the two polynomial functions. If this polynomial function is of second order the error of the calculated position is rather high. If the polynomial function is of 5^{th} order the error is very low but the calculation effort is very high.

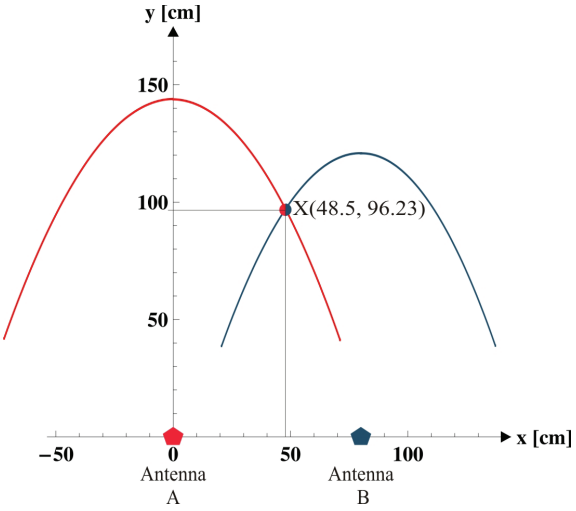


Figure 6: Scenario of two antennas for the algorithm "polynomial approximation".

4.3 Weighted Position Determination

Based on the same possible tag position as in chapter 4.1 the position of the tag is calculated with a "centre of gravity calculation". A similar algorithm is presented in (bulusu 2000).

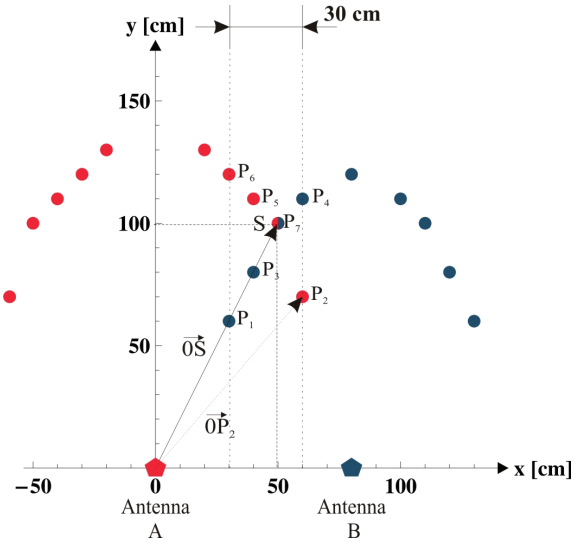


Figure 7: Scenario of two antennas for the algorithm "weighted position determination".

The position of the tag S is calculated by the following equation:

$$\vec{OS} = \frac{1}{\sum_i m_i}\left(\sum_i m_i \cdot \vec{OP_i}\right) \quad (1)$$

Hereby i are all points within the overlapping area and m_i are the appropriate weighting factors.

This algorithm requires fewer computing resources and achieves a higher level of accuracy. It combines easy computing and the possibility of adapting the radio map, allowing for the compensation of atmospheric humidity or other influences.

4.4 Recommendations

Due to the previously mentioned advantages and disadvantages the "weighted position determination" algorithm can be considered best.

This algorithm is very stable and returns a calculated position of the tag with respect to the available discrete relations between RSSI value and possible positions relative to the two antennas.

Accuracy has been checked too. In a wide part of the space it is quite good (less than 5%) but there are single points of rather high inaccuracy. Further investigations have to be carried out to determine their cause and improve this situation. Further details can be found in (schoenegger 2007).

5 CONCLUSIONS

The paper presents the results of having investigated an RFID based location system. Only standard RFID equipment with passive tags operating within the UHF range was used. Determination of location works within a range of approximately 3 meters and is based on the use of the reader's RSSI value. A tag can be located two dimensionally if it is situated within the range of at least two readers. This location algorithm might be used, for example, in combination with a fork-lift, whereby the forks are equipped with antennas. The algorithm is capable of providing a good notion where a specific tag is located relative to the fork.

In addition to logistics application this algorithm may be useful for positioning of mobile robots within production automation.

As the presented location system uses only standard hardware and is based on a simple calculation algorithm, it might be considered as "a new concept". No similar solution is known to the authors.

REFERENCES

Bahl, P, Padmanabhan, V.N., 2000. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proceedings of the 19th International Conference on Computer Communications (Infocom2000), Vol. 2, pp 775-784, Tel Aviv, Israel, March 2000*.

Bekkali, A., Sanson, H., Matsumoto, M., 2007. RFID Indoor Positioning Based on Probailistic RFID Maps and Kalman Filtering. In *3rd IEEE International conference on Wireless and Mobile Computing, Networking and Communications, October 2007, White Plains, USA, October 2007*.

Bulusu, N., 2000. GPS-less low cost outdoor localization for very small devices. In *IEEE Personal Communications Magazine, 7:28-34, October 2000*.

Chon, H. D. et.al., 2004. Using RFID for Accurate Positioning. In *The 2004 International Symposium on GNSS/GPS, Sydney, Australia, December 2004*.

Ekahau, 2007. Ekahau Positioning Engine 4.0 User Guide, http://www.ekahau.com (August 8th, 2007).

Finkenzeller, K., 2006. *RFID-Handbuch; Grundlagen und Anwendungen induktiver Funkanlagen, Transponder und Kontaktloser Chipkarten*, Hanser Fachbuchverlag, 4th edition.

Geroldt, C., Uckelmann, D., 2007. Tracking and Tracing in Production Scenarios with Passive RFID Transponders. In *3rd European Workshop on RFID Systems and Technologies, Duisburg, Germany, VDE-Verlag, June 2007*.

Ibach, P., Stantchev, V., Lederer F., Weiss A., 2005. WLAN-Based Asset Tracking for Warehouse management. In *IADIS International Conference e-Commerce, Porto, Portugal, December 2005*.

Lionel, M.Ni, Yunhao L., Yiu C.L., Abhishek P., 2004. LANDMARC: Indoor Location Sensing Using Active RFID. In *Wireless Networks, Volume 10, Number 6, Springer Netherlands, November 2004*.

Schoenegger, C., 2007. Untersuchung der Eignung eines passiven UHF RFID Systems für die Positionsbestimmung in geschlossenen Räumen. In *Diploma Thesis, University of Applied Sciences Upper Austria, School of Engineering, Wels, September 2007*.

Tomberge, P., 2004. Navigation mittels RFID – Betrachtung der Navigationsmöglichkeiten durch RFID-Eintrittskarten bei der WM 2006. In *Diploma Thesis, University of Münster, Germany, December 2004*.

Tsukiyama, T., Suzuki, A., 2007. Navigation System for indoor Mobile Robots based on RFID Tags, *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics – ICINCO, Angers, France, May 2007*.

OPTIMAL CONTROL WITH ADAPTIVE INTERNAL DYNAMICS MODELS

Djordje Mitrovic, Stefan Klanke and Sethu Vijayakumar
Institute of Perception, Action and Behavior, School of Informatics, University of Edinburgh, Edinburgh, U.K.
d.mitrovic@ed.ac.uk, s.klanke@ed.ac.uk, sethu.vijayakumar@ed.ac.uk

Keywords: Learning dynamics, optimal control, adaptive control, robot simulation.

Abstract: Optimal feedback control has been proposed as an attractive movement generation strategy in goal reaching tasks for anthropomorphic manipulator systems. The optimal feedback control law for systems with non-linear dynamics and non-quadratic costs can be found by iterative methods, such as the iterative Linear Quadratic Gaussian (iLQG) algorithm. So far this framework relied on an analytic form of the system dynamics, which may often be unknown, difficult to estimate for more realistic control systems or may be subject to frequent systematic changes. In this paper, we present a novel combination of learning a forward dynamics model within the iLQG framework. Utilising such adaptive internal models can compensate for complex dynamic perturbations of the controlled system in an online fashion. The specific adaptive framework introduced lends itself to a computationally more efficient implementation of the iLQG optimisation without sacrificing control accuracy – allowing the method to scale to large DoF systems.

1 INTRODUCTION

We address the problem related to control of movement in large degree of freedom (DoF) anthropomorphic manipulators, with specific emphasis on (target) reaching tasks. This is challenging mainly due to the large redundancies that such systems exhibit. For example, a controller has to make a choice between many different possible trajectories (kinematics) and a multitude of applicable motor commands (dynamics) for achieving a particular task. How do we resolve this redundancy?

Optimal control theory (Stengel, 1994) answers this question by postulating that a particular choice is made because it is the optimal solution to the task. Most optimal motor control models so far have focused on open loop optimisation in which the sequence of motor commands or the trajectory is directly optimised with respect to some cost function, for example, minimum jerk (Flash and Hogan, 1985), minimum torque change (Uno et al., 1989), minimum end point variance (Harris and Wolpert, 1998), etc. Trajectory *planning* and *execution* steps are separated and errors during execution are compensated for by using a feedback component (e.g., PID controller). However, these corrections are not taken into account in the optimisation process.

A suggested alternative to open loop models are closed loop optimisation models, namely optimal feedback controllers (OFC) (Todorov, 2004). In contrast to open loop optimisation that just produces a desired optimal trajectory, in OFC, the gains of a feedback controller are optimised to produce an optimal mapping from state to control signals (control law). A key property of OFC is that errors are only corrected by the controller if they adversely affect the task performance, otherwise they are neglected (minimum intervention principle (Todorov and Jordan, 2003)). This is an important property especially in systems that suffer from control dependent noise, since task-irrelevant correction could destabilise the system beside expending additional control effort. Another interesting feature of OFC is that desired trajectories do not need to be planned explicitly but they simply fall out from the feedback control laws. Empirically, OFC also accounts for many motion patterns that have been observed in natural, redundant systems and human experiments (Shadmehr and Wise, 2005)

including the confounding trial-to-trial variability in individual degrees of freedom that, remarkably, manages to not compromise task optimality (Li, 2006; Scott, 2004). Therefore, this paradigm is potentially a very attractive control strategy for artificial anthropomorphic systems (i.e., large DoF, redundant actuation, flexible lightweight construction, variable stiffness).

Finding an optimal control policy for nonlinear systems is a big challenge because they do not follow the well explained Linear-Quadratic-Gaussian formalism (Stengel, 1994). Global solutions could be found in theory by applying dynamic programming methods (Bertsekas, 1995) that are based on the Hamilton-Jacobi-Bellman equations. However, in their basic form these methods rely on a discretisation of the state and action space, an approach that is not viable for large DoF systems. Some research has been carried out on random sampling in a continuous state and action space (Thrun, 2000), and it has been suggested that sampling can avoid the curse of dimensionality if the underlying problem is simple enough (Atkeson, 2007), as is the case if the dynamics and cost functions are very smooth.

As an alternative, one may use iterative approaches that are based on local approximations of the cost and dynamics functions, such as differential dynamic programming (Jacobson and Mayne, 1970), iterative linear-quadratic regulator designs (Li and Todorov, 2004), or the recent iterative Linear Quadratic Gaussian (iLQG) framework (Todorov and Li, 2005; Li and Todorov, 2007), which will form the basis of our work.

A major shortcoming of iLQG is its dependence on an analytic form of the system dynamics, which often may be unknown or subject to change. We overcome this limitation by learning an adaptive internal model of the system dynamics using an online, supervised learning method. We consequently use the learned models to derive an iLQG formulation that is computationally less expensive (especially for large DoF systems), reacts optimally to transient perturbations as well as adapts to systematic changes in plant dynamics.

The idea of learning the system dynamics in combination with iterative optimisations of trajectory or policy has been explored previously in the literature, e.g., for learning to swing up a pendulum (Atkeson and Schaal, 1997) using some prior knowledge about the form of the dynamics. Similarly, Abeel et al. (Abbeel et al., 2006) proposed a hybrid reinforcement learning algorithm, where a policy and an internal model get subsequently updated from "real life" trials. In contrast to their method, however, we (or rather iLQG) employ second-order optimisation methods, and we iteratively refine the control policy solely from the internal model. To our knowledge, learning dynamics in conjunction with control optimisation has not been studied in the light of adaptability to changing plant dynamics.

The remainder of this paper is organised as follows: In the next section, we recall some basic concepts of optimal control theory and we briefly describe the iLQG framework. In Section 3, we introduce our extension to that framework, and we explain how we include a learned internal model of the dynamics. We demonstrate the benefits of our method experimentally in Section 4, and we conclude the paper with a discussion of our work and future research directions in Section 5.

2 LOCALLY-OPTIMAL FEEDBACK CONTROL

Let $\mathbf{x}(t)$ denote the state of a plant and $\mathbf{u}(t)$ the applied control signal at time t. In this paper, the state consists of the joint angles $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ of a robot, and the control signals $\mathbf{u}$ are torques. If the system would be deterministic, we could express its dynamics as $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$, whereas in the presence of noise we write the dynamics as a stochastic differential equation

$$d\mathbf{x} = \mathbf{f}(\mathbf{x},\mathbf{u})dt + \mathbf{F}(\mathbf{x},\mathbf{u})d\boldsymbol{\omega}. \quad (1)$$

Here, $d\boldsymbol{\omega}$ is assumed to be Brownian motion noise, which is transformed by a possibly state- and control-dependent matrix $\mathbf{F}(\mathbf{x},\mathbf{u})$. We state our problem as follows: Given an initial state $\mathbf{x}_0$ at time $t = 0$, we seek a control sequence $\mathbf{u}(t)$ such that the system's state is $\mathbf{x}^*$ at time $t = T$. Stochastic optimal control theory approaches the problem by first specifying a cost function which is composed of (i) some evaluation $h(\mathbf{x}(T))$ of the final state, usually penalising deviations from the desired state $\mathbf{x}^*$, and (ii) the accumulated cost $c(t,\mathbf{x},\mathbf{u})$ of sending a control signal $\mathbf{u}$ at time t in state $\mathbf{x}$, typically penalising large motor commands. Introducing a policy $\boldsymbol{\pi}(t,\mathbf{x})$ for selecting $\mathbf{u}(t)$, we can write the expected cost of following that policy from time t as (Todorov and Li, 2005)

$$v^{\boldsymbol{\pi}}(t,\mathbf{x}(t)) = \left\langle h(\mathbf{x}(T)) + \int_t^T c(s,\mathbf{x}(s),\boldsymbol{\pi}(s,\mathbf{x}(s)))ds \right\rangle \quad (2)$$

One then aims to find the policy $\boldsymbol{\pi}$ that minimises the total expected cost $v^{\boldsymbol{\pi}}(0,\mathbf{x}_0)$. Thus, in contrast to classical control, calculation of the trajectory (planning) and the control signal (execution) is not separated anymore, and for example, redundancy can actually be exploited in order to decrease the cost. If the dynamics $\mathbf{f}$ is linear in $\mathbf{x}$ and $\mathbf{u}$, the cost is

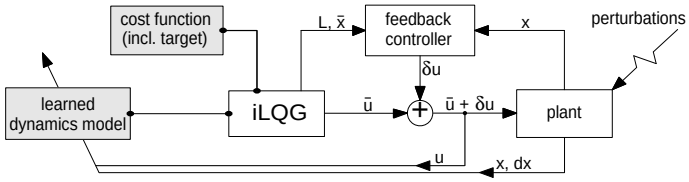


Figure 1: Illustration of our iLQG–LD learning and control scheme.

quadratic, and the noise is Gaussian, the resulting so-called LQG problem is convex and can be solved analytically (Stengel, 1994).

In the more realistic case of non-linear dynamics and non-quadratic cost, one can make use of time-varying linear approximations and apply a similar formalism to iteratively improve a policy, until at least a local minimum of the cost function is found. The resulting iLQG algorithm has only recently been introduced (Todorov and Li, 2005), so we give a brief summary in the following.

One starts with an initial time-discretised control sequence $\bar{\mathbf{u}}_k \equiv \bar{\mathbf{u}}(k\Delta t)$ and applies the deterministic forward dynamics to retrieve an initial trajectory $\bar{\mathbf{x}}_k$, where

$$\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + \Delta t\, \mathbf{f}(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k). \quad (3)$$

Linearising the discretised dynamics (1) around $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{u}}_k$ and subtracting (3), one gets a dynamics equation for the deviations $\delta\mathbf{x}_k = \mathbf{x}_k - \bar{\mathbf{x}}_k$ and $\delta\mathbf{u}_k = \mathbf{u}_k - \bar{\mathbf{u}}_k$:

$$\begin{aligned}\delta\mathbf{x}_{k+1} &= \left(\mathbf{I} + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\bar{\mathbf{x}}_k}\right)\delta\mathbf{x}_k + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{\bar{\mathbf{u}}_k}\delta\mathbf{u}_k \\ &\quad + \sqrt{\Delta t}\left(\mathbf{F}(\mathbf{u}_k) + \frac{\partial \mathbf{F}}{\partial \mathbf{u}}\Big|_{\bar{\mathbf{u}}_k}\delta\mathbf{u}_k\right)\boldsymbol{\xi}_k. \quad (4)\end{aligned}$$

Here, $\boldsymbol{\xi}_k$ represents zero mean Gaussian noise. Similarly, one can derive an approximate cost function which is quadratic in $\delta\mathbf{u}$ and $\delta\mathbf{x}$. Thus, in the vicinity of the current trajectory $\bar{\mathbf{x}}$, the two approximations form a "local" LQG problem, which can be solved analytically and yields an affine control law $\delta\mathbf{u}_k = \mathbf{l}_k + \mathbf{L}_k\delta\mathbf{x}_k$ (for details please see (Todorov and Li, 2005)). This control law is fed into the linearised dynamics (eq. 4 without the noise term) and the resulting $\delta\mathbf{x}$ are used to update the trajectory $\bar{\mathbf{x}}$. In the same way, the control sequence $\bar{\mathbf{u}}$ is updated from $\delta\mathbf{u}$. This process is repeated until the total cost cannot be reduced anymore. The resultant control sequence $\bar{\mathbf{u}}$ can than be applied to the system, whereas the matrices $\mathbf{L}_k$ from the final iteration may serve as feedback gains.

In our current implementation[1], we do not utilise an explicit noise model $\mathbf{F}$ for the sake of clarity of results; in any case, a matching feedback control law is only marginally superior to one that is optimised for a deterministic system (Todorov and Li, 2005).

[1]We used an adapted version of the iLQG implementation at: www.cogsci.ucsd.edu/∼todorov/software.htm

3 iLQG WITH LEARNED DYNAMICS (iLQG–LD)

In order to eliminate the need for an analytic dynamics model and to make iLQG adaptive, we wish to learn an approximation $\tilde{\mathbf{f}}$ of the real plant forward dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. Assuming our model $\tilde{\mathbf{f}}$ has been coarsely pre-trained, for example by motor babbling, we can refine that model in an online fashion as shown in Fig. 1. For optimising and carrying out a movement, we have to define a cost function (where also the desired final state is encoded), the start state, and the number of discrete time steps. Given an initial torque sequence $\bar{\mathbf{u}}^0$, the iLQG iterations can be carried out as described in the previous section, but utilising the learned model $\tilde{\mathbf{f}}$. This yields a locally optimal control sequence $\bar{\mathbf{u}}_k$, a corresponding desired state sequence $\bar{\mathbf{x}}_k$, and feedback correction gain matrices $\mathbf{L}_k$. Denoting the plant's true state by $\mathbf{x}$, at each time step k, the feedback controller calculates the required correction to the control signal as $\delta\mathbf{u}_k = \mathbf{L}_k(\mathbf{x}_k - \bar{\mathbf{x}}_k)$. We then use the final control signal $\mathbf{u}_k = \bar{\mathbf{u}}_k + \delta\mathbf{u}_k$, the plant's state $\mathbf{x}_k$ and its change $d\mathbf{x}_k$ to update our internal forward model $\tilde{\mathbf{f}}$. As we show in Section 4, we can thus account for (systematic) perturbations and also bootstrap a dynamics model from scratch.

The domain of real-time robot control demands certain properties of a learning algorithm, namely fast learning rates, high prediction speeds at run-time, and robustness towards negative interference if the model is trained incrementally. Locally Weighted Projection Regression (LWPR) has been shown to exhibit these properties, and to be very efficient for incremental learning of non-linear models in high dimensions (Vijayakumar et al., 2005). In LWPR, the regression function is constructed by blending local linear models, each of which is endowed with a locality kernel that defines the area of its validity (also termed its receptive field). During training, the parameters of the local models (locality and fit) are updated using incremental Partial Least Squares, and models can be pruned or added on an as-need basis, for example,

when training data is generated in previously unexplored regions.

Usually the receptive fields of LWPR are modelled by Gaussian kernels, so their activation or response to a query vector $\mathbf{z}$ (combined inputs $\mathbf{x}$ and $\mathbf{u}$ of the forward dynamics $\tilde{\mathbf{f}}$) is given by

$$w_k(\mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{c}_k)^T\mathbf{D}_k(\mathbf{z}-\mathbf{c}_k)\right), \quad (5)$$

where $\mathbf{c}_k$ is the centre of the k^{th} linear model and $\mathbf{D}_k$ is its distance metric. Treating each output dimension separately for notational convenience, the regression function can be written as

$$\tilde{f}(\mathbf{z}) = \frac{1}{W}\sum_{k=1}^{K} w_k(\mathbf{z})\psi_k(\mathbf{z}), \quad W = \sum_{k=1}^{K} w_k(\mathbf{z}), \quad (6)$$

$$\psi_k(\mathbf{z}) = b_k^0 + \mathbf{b}_k^T(\mathbf{z}-\mathbf{c}_k), \quad (7)$$

where b_k^0 and $\mathbf{b}_k$ denote the offset and slope of the k-th model, respectively.

LWPR learning has the desirable property that it can be carried out online, and moreover, the learned model can be adapted to changes in the dynamics in real-time. A forgetting factor λ (Vijayakumar et al., 2005), which balances the trade-off between preserving what has been learned and quickly adapting to the non-stationarity, can be tuned to the expected rate of external changes. As we will see later, the factor λ can be used to model biologically realistic adaptive behaviour to external force-fields.

So far, we have shown how the problem of unknown or changing system dynamics can be solved within iLQG–LD. Another important issue to address is the computational complexity. The iLQG framework has been shown to be the most effective locally optimal control method in terms of convergence speed and accuracy (Li, 2006). Nevertheless the computational cost of iLQG remains daunting even for simple movement systems, preventing their application to real-time, large DoF systems. A major component of the computational cost is due to the linearisation of the system dynamics, which involves repetitive calculation of the system dynamics' derivatives $\partial\mathbf{f}/\partial\mathbf{x}$ and $\partial\mathbf{f}/\partial\mathbf{u}$. When the analytical form of these derivatives is not available, they must be approximated using finite differences. The computational cost of such an approximation scales linearly with the sum of the dimensionalities of $\mathbf{x} = (\mathbf{q};\dot{\mathbf{q}})$ and $\mathbf{u} = \boldsymbol{\tau}$ (i.e., $3N$ for an N DoF robot). In simulations, our analysis show that for the 2 DoF manipulator, 60% of the total iLQG computations can be attributed to finite differences calculations. For a 6 DoF arm, this rises to 80%.
Within our iLQG–LD scheme, we can avoid finite difference calculations and rather use the analytic derivatives of the learned model, as has also been proposed in (Atkeson et al., 1997). Differentiating the LWPR predictions (6) with respect to $\mathbf{z} = (\mathbf{x};\mathbf{u})$ yields terms

$$\begin{aligned}\frac{\partial\tilde{f}(\mathbf{z})}{\partial\mathbf{z}} &= \frac{1}{W}\sum_k\left(\frac{\partial w_k}{\partial\mathbf{z}}\psi_k(\mathbf{z}) + w_k\frac{\partial\psi_k}{\partial\mathbf{z}}\right) \\ &\quad -\frac{1}{W^2}\sum_k w_k(\mathbf{z})\psi_k(\mathbf{z})\sum_l\frac{\partial w_l}{\partial\mathbf{z}} \quad (8)\\ &= \frac{1}{W}\sum_k(-\psi_k w_k\mathbf{D}_k(\mathbf{z}-\mathbf{c}_k) + w_k\mathbf{b}_k) \\ &\quad +\frac{\tilde{f}(\mathbf{z})}{W}\sum_k w_k\mathbf{D}_k(\mathbf{z}-\mathbf{c}_k) \quad (9)\end{aligned}$$

for the different rows of the Jacobian matrix

$$\begin{pmatrix}\partial\tilde{\mathbf{f}}/\partial\mathbf{x}\\ \partial\tilde{\mathbf{f}}/\partial\mathbf{u}\end{pmatrix} = \tfrac{\partial}{\partial\mathbf{z}}(\tilde{f}_1,\tilde{f}_2,\ldots\tilde{f}_N)^T.$$

Table 1 illustrates the computational gain (mean CPU time per iLQG iteration) across 4 test manipulators – highlighting added benefits for more complex systems. On a notebook running at 1.6 GHz, the average CPU times for a *complete* iLQG trajectory using the analytic method are 0.8 sec (2 DoF), 1.9 sec (6 DoF), and 9.8 sec (12 DoF), respectively. Note that LWPR is a highly parallelisable algorithm: Since the local models learn independently of each other, the respective computations could be distributed across multiple processors, which would yield a further significant performance gain.

Table 1: CPU time for one iLQG–LD iteration (sec).

manipulator:	**2 DoF**	**6 DoF**	**12 DoF**
finite differences	0.438	4.511	29.726
analytic Jacobian	0.193	0.469	1.569
improvement factor	2.269	9.618	18.946

4 EXPERIMENTS

We studied iLQG–LD on two different joint torque controlled manipulators. The first (Fig. 2, left) is a horizontally planar 2 DoF manipulator similar to the one used in (Todorov and Li, 2005). This low DoF system is ideal for performing extensive (quantitative) comparison studies and to test the manipulator under controlled perturbations and force fields during planar motion. The second experimental setup is a 6 DoF manipulator, the physical parameters (i.e., inertia tensors, mass, gear ratios etc.) of which are a faithful model of the actual *DLR Light-Weight Robot (LWR)* from the German Aerospace Centre[2] (Fig. 2, right).

[2]http://www.dlr.de

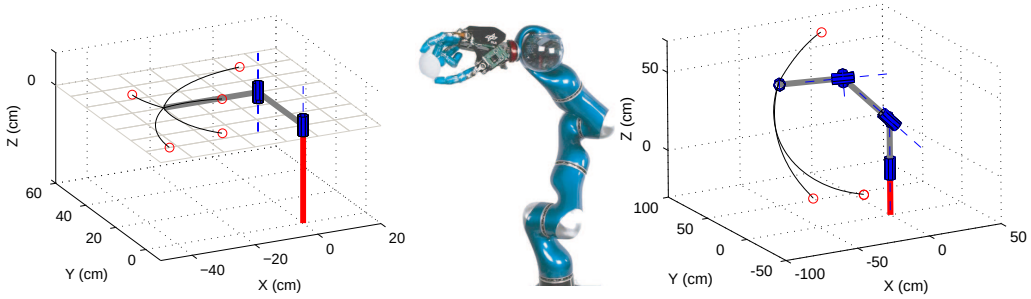


Figure 2: Two different manipulator models with selected targets (circles) and iLQG generated trajectories as benchmark data. All models are simulated using the Matlab Robotics Toolbox. Left: 2 DoF planar manipulator model; Middle: picture of real LWR; Right: 6 DoF LWR model (without hand).

This setup is used to evaluate iLQG–LD on a realistic, redundant anthropomorphic system.

Our simulation model computes the non-linear plant dynamics using standard equations of motion. For an N-DoF manipulator the joint torques $\boldsymbol{\tau}$ are given by

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{b}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}), \qquad (10)$$

where $\mathbf{q}$ and $\dot{\mathbf{q}}$ are the joint angles and joint velocities respectively; $\mathbf{M}(\mathbf{q})$ is the N-dimensional symmetric joint space inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ accounts for Coriolis and centripetal effects, $\mathbf{b}(\dot{\mathbf{q}})$ describes the viscous and Coulomb friction in the joints, and $\mathbf{g}(\mathbf{q})$ defines the gravity loading depending on the joint angles $\mathbf{q}$ of the manipulator. It is important to note that while the above analytical dynamics perfectly match the system dynamics in simulation, they are at best, an extremely crude approximation to the dynamics of the real hardware arm.

We study movements for a fixed motion duration of one second, which we discretise into $K = 100$ steps (Δt = 0.01s). The manipulator starts at an initial position $\mathbf{q}_0$ and reaches towards a target $\mathbf{q}_{tar}$. During movement we wish to minimise the energy consumption of the system. We therefore use the cost function[3]

$$v = w_p \,|\, \mathbf{q}_K - \mathbf{q}_{tar} \,|^2 + w_v \,|\, \dot{\mathbf{q}}_K \,|^2 + w_e \sum_{k=0}^{K} |\, \mathbf{u}_k \,|^2 \Delta t, \qquad (11)$$

where the factors for the target position accuracy (w_p), the final target velocity accuracy (w_v), and for the energy term (w_e) weight the importance of each component.

We compare the control results of iLQG–LD and iLQG with respect to the number of iterations, the end point accuracy and the generated costs. We first present results for the 2 DoF planar arm in order to test whether our theoretical assumptions hold and iLQG–LD works in practice (Sections 4.1 and 4.2). In a final experiment, we present qualitative and quantitative results to show that iLQG–LD scales up to the redundant 6 DoF anthropomorphic system (Section 4.3).

[3]We specify the target in joint space only for the 2-DoF arm.

4.1 Stationary Dynamics

First, we compared the characteristics of iLQG–LD and iLQG (both operated in open loop mode) in the case of stationary dynamics without any noise in the 2 DoF plant. Fig. 3 shows three trajectories generated by learned models of different predictive quality (reflected by the different test nMSE). As one would expect, the quality of the model plays an important role for the final cost, the number of iLQG–LD iterations, and the final target distances (cf. the table within Fig. 3). For the final learned model, we observe a striking resemblance with the analytic iLQG performance. Real world systems usually suffer from control dependent noise, so in order to be practicable, iLQG–LD has to be able to cope with this. Next, we carried out a reaching task to 5 reference targets covering a wide operating area of the planar arm. To simulate control dependent noise, we contaminated commands $\mathbf{u}$ just before feeding them into the plant, using Gaussian noise with 50% of the variance of the signal $\mathbf{u}$. We then generated motor commands to move the system towards the targets, both with and without the feedback controller. As expected, closed loop control (utilising gain matrices $\mathbf{L}_k$) is superior to open loop operation regarding reaching accuracy. Fig. 4 depicts the performance of iLQG–LD and iLQG under both control schemes. Averaged over all trials, both methods show similar endpoint variances and behaviour which is statistically indistinguishable.

	iLQG–LD	(L)	(M)	(H)	iLQG
Train. points		111	146	276	–
Prediction error (nMSE)		0.80	0.50	0.001	–
iterations		19	17	5	4
Cost		2777.36	1810.20	191.91	192.07
Eucl. target distance (cm)		19.50	7.20	0.40	0.01

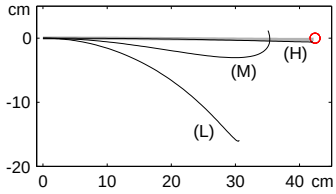


Figure 3: Behaviour of iLQG–LD for learned models of different quality. Right: Trajectories in task space produced by iLQG–LD (black lines) and iLQG (grey line).

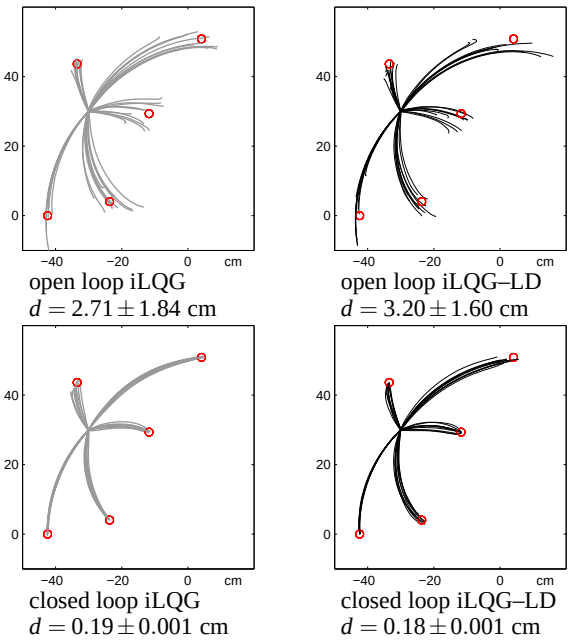


Figure 4: Illustration of the target reaching performances for the planar 2 DoF in the presence of strong control dependent noise, where d represents the average Euclidean distance to the five reference targets.

4.2 Non-stationary Dynamics

A major advantage of iLQG–LD is that it does not rely on an accurate analytic dynamics model; consequently, it can adapt 'on-the-fly' to external perturbations and to changes in the plant dynamics that may result from altered morphology or wear and tear.

We carried out adaptive reaching experiments in our simulation similar to the human manipulandum experiments in (Shadmehr and Mussa-Ivaldi, 1994). First, we generated a constant unidirectional force field (FF) acting perpendicular to the reaching movement (see Fig. 5). Using the iLQG–LD models from the previous experiments, the manipulator gets strongly deflected when reaching for the target because the learned dynamics model cannot account for the "spurious" forces. However, using the resultant deflected trajectory (100 data points) as training data, updating the dynamics model online brings the manipulator nearer to the target with each new trial. We repeated this procedure until the iLQG–LD performance converged successfully. At that point, the internal model successfully accounts for the change in dynamics caused by the FF. Then, removing the FF results in the manipulator overshooting to the other side, compensating for a non-existing FF. Just as before, we re-adapted the dynamics online over repeated trials.

Fig. 5 summarises the results of the sequential adaptation process just described. The closed loop control scheme clearly converges faster than the open loop scheme, which is mainly due to the OFC's desirable property of always correcting the system towards the target. Therefore, it produces relevant dynamics training data in a way that could be termed "active learning". Furthermore, we can accelerate the adaptation process significantly by tuning the forgetting factor λ, allowing the learner to weight the importance of new data more strongly (Vijayakumar et al., 2005). A value of $\lambda = 0.95$ produces significantly faster adaptation results than the default of $\lambda = 0.999$. As a follow-up experiment, we made the force field dependent on the velocity $\mathbf{v}$ of the end-effector, i.e. we applied a force

$$\mathbf{F} = \mathbf{B}\mathbf{v}, \quad \text{with} \quad \mathbf{B} = \begin{pmatrix} 0 & 50 \\ -50 & 0 \end{pmatrix} Nm^{-1}s \quad (12)$$

to the end-effector. The results are illustrated in Fig. 6: For the more complex FF, more iterations are needed in order to adapt the model, but otherwise iLQG–LD shows a similar behaviour as for the constant FF. Interestingly, the overshooting behaviour depicted in Fig. 5 and 6 has been observed in human adaptation experiments (Shadmehr and Mussa-Ivaldi, 1994). We believe this to be an interesting insight for future investigation of iLQG–LD and its role in modeling sensorimotor adaptation data in the (now extensive) human reach experimental paradigm (Shadmehr and Wise, 2005).

4.3 iLQG–LD for 6 DoF

In the 6 DoF LWR, we studied reaching targets specified in *task space* coordinates $\mathbf{r} \in \mathcal{R}^3$ in order to high-

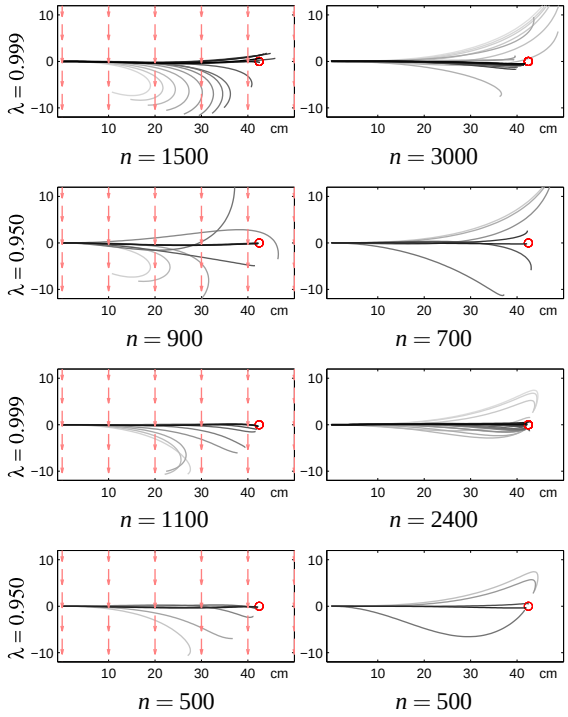


Figure 5: Illustration of adaptation experiments for open loop (rows 1,2) and closed loop (rows 3,4) iLQG–LD. Arrows depict the presence of a (constant) force field; n represents the number of training points required to successfully update the internal LWPR dynamics model. Darker lines indicate better trained models, corresponding to later trials in the adaption process.

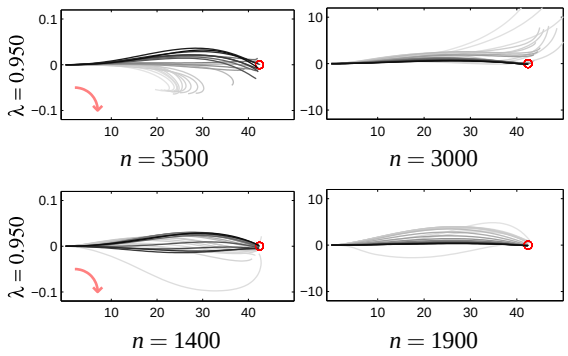


Figure 6: Adaptation to a velocity-dependent force field (as indicated by the bent arrow) and re-adaptation after the force field is switched off (right column). Top: open loop. Bottom: closed loop.

light the redundancy resolution capability and trial-to-trial variability in large DoF systems. Therefore, we replaced the term $|\mathbf{q}_K - \mathbf{q}_{tar}|^2$ by $|\mathbf{r}(\mathbf{q}_K) - \mathbf{r}_{tar}|^2$ in (11), where $\mathbf{r}(\mathbf{q})$ denotes the forward kinematics.

Similar to the 2 DoF, we bootstrapped a forward dynamics model through 'motor babbling'. Next, we used iLQG–LD (closed loop, with noise) to train our dynamics model online until it converged to stable reaching behaviour. Fig. 7 (left) depicts reaching trials, 20 for each reference target, using iLQG–LD with the final learned model. Table 2 quantifies the performance. The targets are reached reliably and no statistically significant differences can be spotted between iLQG–LD and iLQG. An investigation of the trials in *joint angle* space also shows similarities. Fig. 7 (right) depicts the 6 joint angle trajectories for the 20 reaching trials towards target (c). The joint angle variances are much higher compare d to the variances of the task space trajectories, meaning that task irrelevant errors are not corrected unless they adversely affect the task (minimum intervention principle of OFC). Moreover, the joint angle variances (trial-to-trial variability) between the iLQG–LD and iLQG trials are in a similar range, indicating an equivalent corrective behaviour – the shift of the absolute variances can be explained by the slight mismatch in between the learned and analytical dynamics. We can conclude from our results that iLQG–LD scales up very well to 6 DoF, not suffering from any losses in terms of accuracy, cost or convergence behaviour. Furthermore, its computational cost is significantly lower than the one of iLQG.

Table 2: Comparison of iLQG–LD and iLQG for controlling a 6 DOF arm to reach for three targets.

iLQG–LD	**Iter.**	**Run. cost**	**d (cm)**
(a)	17	**18.32** $\pm$ 0.55	**1.92** $\pm$ 1.03
(b)	30	**18.65** $\pm$ 1.61	**0.53** $\pm$ 0.20
(c)	51	**12.18** $\pm$ 0.03	**2.00** $\pm$ 1.02
iLQG	**Iter.**	**Run. cost**	**d (cm)**
(a)	19	**18.50** $\pm$ 0.13	**2.63** $\pm$ 1.63
(b)	26	**18.77** $\pm$ 0.25	**1.32** $\pm$ 0.69
(c)	48	**12.92** $\pm$ 0.04	**1.75** $\pm$ 1.30

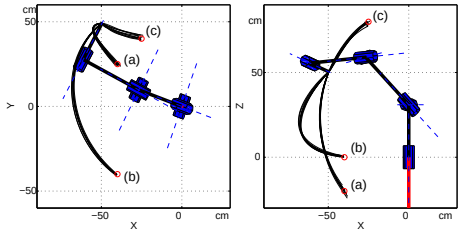


Figure 7: Illustration of the 6-DoF manipulator and the trajectories for reaching towards the targets (a,b,c). Left: top-view, right: side-view.

5 CONCLUSIONS

In this work we introduced iLQG–LD, a method that realises adaptive optimal feedback control by incorporating a learned dynamics model into the iLQG

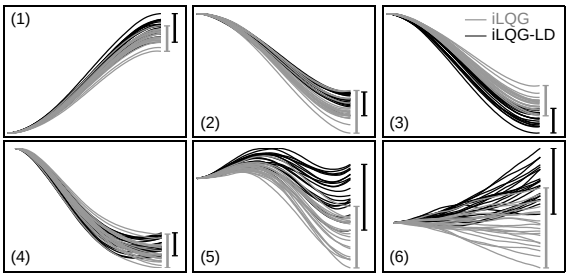


Figure 8: Illustration of the trial-to-trial variability of the 6-DoF arm when reaching towards target (c). The plots depict the joint angles (1–6) over time. Grey lines indicate iLQG, black lines stem from iLQG–LD.

framework. Most importantly, we carried over the favourable properties of iLQG to more realistic control problems where the analytic dynamics model is often unknown, difficult to estimate accurately or subject to changes.

Utilising the derivatives (8) of the learned dynamics model $\hat{\mathbf{f}}$ avoids expensive finite difference calculations during the dynamics linearisation step of iLQG. This significantly reduces the computational complexity, allowing the framework to scale to larger DoF systems. We empirically showed that iLQG–LD performs reliably in the presence of noise and that it is adaptive with respect to systematic changes in the dynamics; hence, the framework has the potential to provide a unifying tool for modelling (and informing) non-linear sensorimotor adaptation experiments even under complex dynamic perturbations. As with iLQG control, redundancies are implicitly resolved by the OFC framework through a cost function, eliminating the need for a separate trajectory planner and inverse kinematics/dynamics computation.

Our future work will concentrate on implementing the iLQG–LD framework on the anthropomorphic LWR hardware – this will not only explore an alternative control paradigm, but will also provide the only viable and principled control strategy for the biomorphic *variable stiffness* based highly redundant actuation system that we are currently developing. Indeed, exploiting this framework for understanding OFC and its link to biological motor control is another very important strand.

REFERENCES

Abbeel, P., Quigley, M., and Ng, A. Y. (2006). Using inaccurate models in reinforcement learning. In *Proc. Int. Conf. on Machine Learning*, pages 1–8.

Atkeson, C. G. (2007). Randomly sampling actions in dynamic programming. In *Proc. Int. Symp. on Approximate Dynamic Programming and Reinforcement Learning*, pages 185–192.

Atkeson, C. G., Moore, A., and Schaal, S. (1997). Locally weighted learning for control. *AI Review*, 11:75–113.

Atkeson, C. G. and Schaal, S. (1997). Learning tasks from a single demonstration. In *Proc. Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1706–1712, Albuquerque, New Mexico.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*. Athena Scientific, Belmont, Mass.

Flash, T. and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5:1688–1703.

Harris, C. M. and Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature*, 394:780–784.

Jacobson, D. H. and Mayne, D. Q. (1970). *Differential Dynamic Programming*. Elsevier, New York.

Li, W. (2006). *Optimal Control for Biological Movement Systems*. PhD dissertation, University of California, San Diego.

Li, W. and Todorov, E. (2004). Iterative linear-quadratic regulator design for nonlinear biological movement systems. In *Proc. 1st Int. Conf. Informatics in Control, Automation and Robotics*.

Li, W. and Todorov, E. (2007). Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. *International Journal of Control*, 80(9):14391453.

Scott, S. H. (2004). Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*, 5:532–546.

Shadmehr, R. and Mussa-Ivaldi, F. A. (1994). Adaptive representation of dynamics during learning of a motor task. *The Journal of Neurosciene*, 14(5):3208–3224.

Shadmehr, R. and Wise, S. P. (2005). *The Computational Neurobiology of Reaching and Ponting*. MIT Press.

Stengel, R. F. (1994). *Optimal control and estimation*. Dover Publications, New York.

Thrun, S. (2000). Monte carlo POMDPs. In Solla, S. A., Leen, T. K., and Müller, K. R., editors, *Advances in Neural Information Processing Systems 12*, pages 1064–1070. MIT Press.

Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9):907–915.

Todorov, E. and Jordan, M. (2003). A minimal intervention principle for coordinated movement. In *Advances in Neural Information Processing Systems*, volume 15, pages 27–34. MIT Press.

Todorov, E. and Li, W. (2005). A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proc. of the American Control Conference*.

Uno, Y., Kawato, M., and Suzuki, R. (1989). Formation and control of optimal trajectories in human multijoint arm movements: minimum torque-change model. *Biological Cybernetics*, 61:89–101.

Vijayakumar, S., D'Souza, A., and Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, 17:2602–2634.

FAIR AND EFFICIENT RESOURCE ALLOCATION
Bicriteria Models for Equitable Optimization

Włodzimierz Ogryczak
Institute of Control & Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, Warsaw, Poland
ogryczak@ia.pw.edu.pl

Keywords: Optimization, multiple criteria, efficiency, fairness, equity, resource allocation.

Abstract: Resource allocation problems are concerned with the allocation of limited resources among competing activities so as to achieve the best performances. In systems which serve many usersthere is a need to respect some fairness rules while looking for the overall efficiency. The so-called Max-Min Fairness is widely used to meet these goals. However, allocating the resource to optimize the worst performance may cause a dramatic worsening of the overall system efficiency. Therefore, several other fair allocation schemes are searched and analyzed. In this paper we focus on mean-equity approaches which quantify the problem in a lucid form of two criteria: the mean outcome representing the overall efficiency and a scalar measure of inequality of outcomes to represent the equity (fairness) aspects. The mean-equity model is appealing to decision makers and allows a simple trade-off analysis. On the other hand, for typical dispersion indices used as inequality measures, the mean-equity approach may lead to inferior conclusions with respect to the outcomes maximization (system efficiency). Some inequality measures, however, can be combined with the mean itself into optimization criteria that remain in harmony with both inequality minimization and maximization of outcomes. In this paper we introduce general conditions for inequality measures sufficient to provide such an equitable consistency. We verify the conditions for the basic inequality measures thus showing how they can be used not leading to inferior distributions of system outcomes.

1 INTRODUCTION

Resource allocation problems are concerned with the allocation of limited resources among competing activities (Ibaraki and Katoh, 1988). In this paper, we focus on approaches that, while allocating resources to maximize the system efficiency, they also attempt to provide a fair treatment of all the competing activities (Luss, 1999). The problems of efficient and fair resource allocation arise in various systems which serve many users, like in telecommunication systems among others. In networking a central issue is how to allocate bandwidth to flows efficiently and fairly (Bonald and Massoulie, 2001; Denda et al., 2000; Kleinberg et al., 2001; Pióro and Medhi, 2004). In location analysis of public services, the decisions often concern the placement of a service center or another facility in a position so that the users are treated fairly in an equitable way, relative to certain criteria (Ogryczak, 2000). Recently, several research publications relating the fairness and equity concepts to the multiple criteria optimization methodology have appeared (Kostreva et al., 2004; Luss, 1999).

The generic resource allocation problem may be stated as follows. Each activity is measured by an individual performance function that depends on the corresponding resource level assigned to that activity. A larger function value is considered better, like the performance measured in terms of quality level, capacity, service amount available, etc. Models with an (aggregated) objective function that maximizes the mean (or simply the sum) of individual performances are widely used to formulate resource allocation problems, thus defining the so-called mean solution concept. This solution concept is primarily concerned with the overall system efficiency. As based on averaging, it often provides solution where some smaller services are discriminated in terms of allocated resources. An alternative approach depends on the so-called Max-Min solution concept, where the worst performance is maximized. The Max-Min approach is consistent with Rawlsian (Rawls, 1971) theory of justice, especially when additionally regularized with the lexicographic order. The latter is called the Max-

Min Fairness (MMF) and commonly used in networking (Pióro and Medhi, 2004; Ogryczak et al., 2005). Allocating the resources to optimize the worst performances may cause, however, a large worsening of the overall (mean) performances. Therefore, there is a need to seek a compromise between the two extreme approaches discussed above.

Fairness is, essentially, an abstract socio-political concept that implies impartiality, justice and equity (Rawls and Kelly, 2001; Young, 1994), Nevertheless, fairness was frequently quantified with the so-called inequality measures to be minimized (Atkinson, 1970; Rothschild and Stiglitz, 1973). Unfortunately, direct minimization of typical inequality measures contradicts the maximization of individual outcomes and it may lead to inferior decisions. In order to ensure fairness in a system, all system entities have to be equally well provided with the system's services. This leads to concepts of fairness expressed by the equitable efficiency (Kostreva and Ogryczak, 1999; Luss, 1999). The concept of equitably efficient solution is a specific refinement of the Pareto-optimality taking into account the inequality minimization according to the Pigou-Dalton approach. In this paper the use of scalar inequality measures in bi-criteria models to search for fair and efficient allocations is analyzed. There is shown that properties of convexity and positive homogeneity together with some boundedness condition are sufficient for a typical inequality measure to guarantee that it can be used consistently with the equitable optimization rules.

2 EQUITY AND FAIRNESS

The generic resource allocation problem may be stated as follows. There is a system dealing with a set I of m services. There is given a measure of services realization within a system. In applications we consider, the measure usually expresses the service quality. In general, outcomes can be measured (modeled) as service time, service costs, service delays as well as in a more subjective way. There is also given a set Q of allocation patterns (allocation decisions). For each service $i \in I$ a function $f_i(\mathbf{x})$ of the allocation pattern $\mathbf{x} \in Q$ has been defined. This function, called the individual objective function, measures the outcome (effect) $y_i = f_i(\mathbf{x})$ of allocation $\mathbf{x}$ pattern for service i. In typical formulations a larger value of the outcome means a better effect (higher service quality or client satisfaction). Otherwise, the outcomes can be replaced with their complements to some large number. Therefore, without loss of generality, we can assume that each individual outcome y_i is to be maximized which allows us to view the generic resource allocation problem as a vector maximization model:

$$\max \{\mathbf{f}(\mathbf{x}) \,:\, \mathbf{x} \in Q\} \tag{1}$$

where $\mathbf{f}(\mathbf{x})$ is a vector-function that maps the decision space $X = R^n$ into the criterion space $Y = R^m$, and $Q \subset X$ denotes the feasible set.

Model (1) only specifies that we are interested in maximization of all objective functions f_i for $i \in I = \{1,2,\ldots,m\}$. In order to make it operational, one needs to assume some solution concept specifying what it means to maximize multiple objective functions. The solution concepts may be defined by properties of the corresponding preference model. The preference model is completely characterized by the relation of weak preference, denoted hereafter with $\succeq$. Namely, the corresponding relations of strict preference $\succ$ and indifference $\cong$ are defined by the following formulas:

$$\begin{aligned} \mathbf{y}' \succ \mathbf{y}'' \quad &\Leftrightarrow \quad (\mathbf{y}' \succeq \mathbf{y}'' \quad \text{and} \quad \mathbf{y}'' \not\succeq \mathbf{y}'), \\ \mathbf{y}' \cong \mathbf{y}'' \quad &\Leftrightarrow \quad (\mathbf{y}' \succeq \mathbf{y}'' \quad \text{and} \quad \mathbf{y}'' \succeq \mathbf{y}'). \end{aligned}$$

The standard preference model related to the Pareto-optimal (efficient) solution concept assumes that the preference relation $\succeq$ is *reflexive*:

$$\mathbf{y} \succeq \mathbf{y}, \tag{2}$$

transitive:

$$(\mathbf{y}' \succeq \mathbf{y}'' \quad \text{and} \quad \mathbf{y}'' \succeq \mathbf{y}''') \;\Rightarrow\; \mathbf{y}' \succeq \mathbf{y}''', \tag{3}$$

and *strictly monotonic*:

$$\mathbf{y} + \varepsilon \mathbf{e}_i \succ \mathbf{y} \quad \text{for } \varepsilon > 0;\ i = 1,\ldots,m, \tag{4}$$

where $\mathbf{e}_i$ denotes the i–th unit vector in the criterion space. The last assumption expresses that for each individual objective function more is better (maximization). The preference relations satisfying axioms (2)–(4) are called hereafter *rational preference relations*. The rational preference relations allow us to formalize the Pareto-optimality (efficiency) concept with the following definitions. We say that outcome vector $\mathbf{y}'$ rationally dominates $\mathbf{y}''$ ($\mathbf{y}' \succ_r \mathbf{y}''$), iff $\mathbf{y}' \succ \mathbf{y}''$ for all rational preference relations $\succeq$. We say that feasible solution $\mathbf{x} \in Q$ is a *Pareto-optimal (efficient)* solution of the multiple criteria problem (1), iff $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is rationally nondominated.

Simple solution concepts for multiple criteria problems are defined by aggregation (or utility) functions $g : Y \to R$ to be maximized. Thus the multiple criteria problem (1) is replaced with the maximization problem

$$\max \{g(\mathbf{f}(\mathbf{x})) \,:\, \mathbf{x} \in Q\} \tag{5}$$

In order to guarantee the consistency of the aggregated problem (5) with the maximization of all individual objective functions in the original multiple criteria problem (or Pareto-optimality of the solution),

the aggregation function must be strictly increasing with respect to every coordinate.

The simplest aggregation functions commonly used for the multiple criteria problem (1) are defined as the mean (average) outcome

$$\mu(\mathbf{y}) = \frac{1}{m}\sum_{i=1}^{m} y_i \tag{6}$$

or the worst outcome

$$M(\mathbf{y}) = \min_{i=1,\ldots,m} y_i. \tag{7}$$

The mean (6) is a strictly increasing function while the minimum (7) is only nondecreasing. Therefore, the aggregation (5) using the sum of outcomes always generates a Pareto-optimal solution while the maximization of the worst outcome may need some additional refinement. The mean outcome maximization is primarily concerned with the overall system efficiency. As based on averaging, it often provides a solution where some services are discriminated in terms of performances. On the other hand, the worst outcome maximization, ie, the so-called Max-Min solution concept is regarded as maintaining equity. Indeed, in the case of a simplified resource allocation problem with the knapsack constraints, the Max-Min solution meets the perfect equity requirement. In the general case, with possibly more complex feasible set structure, this property is not fulfilled. Nevertheless, if the perfectly equilibrated outcome vector $\bar{y}_1 = \bar{y}_2 = \ldots = \bar{y}_m$ is nondominated, then it is the unique optimal solution of the corresponding Max-Min optimization problem. In other words, the perfectly equilibrated outcome vector is a unique optimal solution of the Max-Min problem if one cannot find any (possibly not equilibrated) vector with improved at least one individual outcome without worsening any others. Unfortunately, it is not a common case and, in general, the optimal set to the Max-Min aggregation may contain numerous alternative solutions including dominated ones. The Max-Min solution may be then regularized according to the Rawlsian principle of justice (Rawls, 1971) which leads us to the lexicographic Max-Min concepts or the so-called Max-Min Fairness (Marchi and Oviedo, 1992; Ogryczak and Śliwiński, 2006).

In order to ensure fairness in a system, all system entities have to be equally well provided with the system's services. This leads to concepts of fairness expressed by the equitable rational preferences (Kostreva and Ogryczak, 1999). First of all, the fairness requires impartiality of evaluation, thus focusing on the distribution of outcome values while ignoring their ordering. That means, in the multiple criteria problem (1) we are interested in a set of outcome values without taking into account which outcome is taking a specific value. Hence, we assume that the preference model is impartial (anonymous, symmetric). In terms of the preference relation it may be written as the following axiom

$$(y_{\pi(1)},\ldots,y_{\pi(m)}) \cong (y_1,\ldots,y_m) \quad \forall \pi \in \Pi(I) \tag{8}$$

where $\Pi(I)$ denotes the set of all permutations of I. This means that any permuted outcome vector is indifferent in terms of the preference relation. Further, fairness requires equitability of outcomes which causes that the preference model should satisfy the (Pigou–Dalton) principle of transfers. The principle of transfers states that a transfer of any small amount from an outcome to any other relatively worse–off outcome results in a more preferred outcome vector. As a property of the preference relation, the principle of transfers takes the form of the following axiom

$$\mathbf{y} - \varepsilon\mathbf{e}_i + \varepsilon\mathbf{e}_j \succ \mathbf{y} \quad \text{for } 0 < \varepsilon < y_i - y_j \tag{9}$$

The rational preference relations satisfying additionally axioms (8) and (9) are called hereafter *fair (equitable) rational preference relations*. We say that outcome vector $\mathbf{y}'$ *fairly (equitably) dominates* $\mathbf{y}''$ ($\mathbf{y}' \succ_e \mathbf{y}''$), iff $\mathbf{y}' \succ \mathbf{y}''$ for all fair rational preference relations $\succeq$. In other words, $\mathbf{y}'$ fairly dominates $\mathbf{y}''$, if there exists a finite sequence of vectors $\mathbf{y}^j$ ($j = 1,2,\ldots,s$) such that $\mathbf{y}^1 = \mathbf{y}''$, $\mathbf{y}^s = \mathbf{y}'$ and $\mathbf{y}^j$ is constructed from $\mathbf{y}^{j-1}$ by application of either permutation of coordinates, equitable transfer, or increase of a coordinate. An allocation pattern $\mathbf{x} \in Q$ is called *fairly (equitably) efficient* or simply *fair* if $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is fairly nondominated. Note that each fairly efficient solution is also Pareto-optimal, but not vice verse.

In order to guarantee fairness of the solution concept (5), additional requirements on the class of aggregation (utility) functions must be introduced. In particular, the aggregation function must be additionally symmetric (impartial), i.e. for any permutation π of I,

$$g(y_{\pi(1)},y_{\pi(2)},\ldots,y_{\pi(m)}) = g(y_1,y_2,\ldots,y_m) \tag{10}$$

as well as be equitable (to satisfy the principle of transfers)

$$g(y_1,\ldots,y_i-\varepsilon,\ldots,y_j+\varepsilon,\ldots,y_m) > g(y_1,\ldots,y_m) \tag{11}$$

for any $0 < \varepsilon < y_{i'} - y_{i''}$. In the case of a strictly increasing function satisfying both the requirements (10) and (11), we call the corresponding problem (5) a *fair (equitable) aggregation* of problem (1). Every optimal solution to the fair aggregation (5) of a multiple criteria problem (1) defines some fair (equitable) solution.

Note that both the simplest aggregation functions, the sum (6) and the minimum (7), are symmetric although they do not satisfy the equitability requirement (11). To guarantee the fairness of solutions, some enforcement of concave properties is required. For any strictly concave, increasing utility function $u : R \to R$, the function $g(\mathbf{y}) = \sum_{i=1}^{m} u(y_i)$ is a strictly monotonic and equitable thus defining a family of the fair aggregations. Various concave utility functions u can be used to define such fair solution concepts. In the case of the outcomes restricted to positive values, one may use logarithmic function thus resulting in the *Proportional Fairness* (PF) solution concept (Kelly et al., 1997). Actually, it corresponds to the so-called Nash criterion which maximizes the product of additional utilities compared to the status quo. For a common case of upper bounded outcomes $y_i \le y^*$ one may maximize power functions $-\sum_{i=1}^{m} (y^* - y_i)^p$ for $1 < p < \infty$ which corresponds to the minimization of the corresponding p-norm distances from the common upper bound y^* (Kostreva et al., 2004).

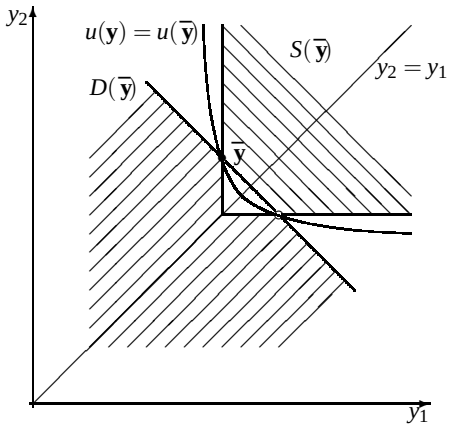


Figure 1: Structure of the fair dominance: $D(\overline{\mathbf{y}})$ – the set fairly dominated by $\overline{\mathbf{y}}$, $S(\overline{\mathbf{y}})$ – the set of outcomes fairly dominating $\overline{\mathbf{y}}$.

Fig. 1 presents the structure of fair dominance for two-dimensional outcome vectors. For any outcome vector $\overline{\mathbf{y}}$, the fair dominance relation distinguishes set $D(\overline{\mathbf{y}})$ of dominated outcomes (obviously worse for all fair rational preferences) and set $S(\overline{\mathbf{y}})$ of dominating outcomes (obviously better for all fair rational preferences). However, some outcome vectors are left (in white areas) and they can be differently classified by various specific fair rational preferences. The MMF fairness assigns the entire interior of the inner white triangle to the set of preferred outcomes while classifying the interior of the external open triangles as worse outcomes. Isolines of various utility functions split the white areas in different ways. One may notice that the set $D(\overline{\mathbf{y}})$ of directions leading to outcome vectors being dominated by a given $\overline{\mathbf{y}}$ is, in general, not a cone and it is not convex. Although, when we consider the set $S(\overline{\mathbf{y}})$ of directions leading to outcome vectors dominating given $\overline{\mathbf{y}}$ we get a convex set.

3 INEQUALITY MEASURES AND FAIR CONSISTENCY

Inequality measures were primarily studied in economics while recently they become very popular tools in Operations Research. Typical inequality measures are some deviation type dispersion characteristics. They are *translation invariant* in the sense that $\rho(\mathbf{y} + \nu\mathbf{e}) = \rho(\mathbf{y})$ for any outcome vector $\mathbf{y}$ and real number ν (where $\mathbf{e}$ vector of units $(1,\ldots,1)$), thus being not affected by any shift of the outcome scale. Moreover, the inequality measures are also *inequality relevant* which means that they are equal to 0 in the case of perfectly equal outcomes while taking positive values for unequal ones.

The simplest inequality measures are based on the absolute measurement of the spread of outcomes, like the *mean absolute difference*

$$\Gamma(\mathbf{y}) = \frac{1}{2m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} |y_i - y_j| \tag{12}$$

or the *maximum absolute difference*

$$d(\mathbf{y}) = \max_{i,j=1,\ldots,m} |y_i - y_j| \tag{13}$$

In most application frameworks better intuitive appeal may have inequality measures related to deviations from the mean outcome like the mean absolute deviation

$$\delta(\mathbf{y}) = \frac{1}{m} \sum_{i=1}^{m} |y_i - \mu(\mathbf{y})| \tag{14}$$

or the *maximum absolute deviation*

$$R(\mathbf{y}) = \max_{i \in I} |y_i - \mu(\mathbf{y})| \tag{15}$$

Note that the *standard deviation* σ (or the *variance* σ^2) represents both the deviations and the spread measurement as

$$\sigma^2(\mathbf{y}) = \frac{\sum_{i \in I} (y_i - \mu(\mathbf{y}))^2}{m} = \frac{\sum_{i \in I} \sum_{j \in I} (y_i - y_j)^2}{2m^2} \tag{16}$$

Deviational measures may be focused on the downside semideviations as related to worsening of outcome while ignoring upper semideviations related to improvement of outcome. One may define the *maximum (downside) semideviation*

$$\Delta(\mathbf{y}) = \max_{i \in I} (\mu(\mathbf{y}) - y_i) \tag{17}$$

and the *mean (downside) semideviation*

$$\bar{\delta}(\mathbf{y}) = \frac{1}{m}\sum_{i\in I}(\mu(\mathbf{y}) - y_i)_+ \tag{18}$$

where $(.)_+$ denotes the nonnegative part of a number. Similarly, the *standard (downside) semideviation* is given as

$$\bar{\sigma}(\mathbf{y}) = \sqrt{\frac{1}{m}\sum_{i\in I}(\mu(\mathbf{y}) - y_i)_+^2} \tag{19}$$

In economics one usually considers relative inequality measures normalized by mean outcome. Among many inequality measures perhaps the most commonly accepted by economists is the Gini coefficient, which is the relative mean difference. One can easily notice that direct minimization of typical inequality measures (especially the relative ones) may contradict the optimization of individual outcomes resulting in equal but very low outcomes. As some resolution one may consider a bicriteria mean-equity model:

$$\max\ \{(\mu(\mathbf{f}(\mathbf{x})), -\rho(\mathbf{f}(\mathbf{x})))\ :\ \mathbf{x}\in Q\} \tag{20}$$

which takes into account both the efficiency with optimization of the mean outcome $\mu(\mathbf{y})$ and the equity with minimization of an inequality measure $\rho(\mathbf{y})$. For typical inequality measures bicriteria model (20) is computationally very attractive since both the criteria are concave and LP implementable for many measures. Unfortunately, for any dispersion type inequality measures the bicriteria mean-equity model is not consistent with the outcomes maximization, and therefore is not consistent with the fair dominance. When considering a simple discrete problem with two allocation patterns P1 and P2 generating outcome vectors $\mathbf{y}' = (0,0)$ and $\mathbf{y}'' = (2,8)$, respectively, for any dispersion type inequality measure one gets $\rho(\mathbf{y}'') > 0 = \rho(\mathbf{y}')$ while $\mu(\mathbf{y}'') = 5 > 0 = \mu(\mathbf{y}')$. Hence, $\mathbf{y}''$ is not bicriteria dominated by $\mathbf{y}'$ and vice versa. Thus for any dispersion type inequality measure ρ, allocation P1 with obviously worse outcome vector than that for allocation P2 is a Pareto-optimal solution in the corresponding bicriteria mean-equity model (20).

Note that the lack of consistency of the mean-equity model (20) with the outcomes maximization applies also to the case of the maximum semideviation $\Delta(\mathbf{y})$ (17) used as an inequality measure whereas subtracting this measure from the mean $\mu(\mathbf{y}) - \Delta(\mathbf{y}) = M(\mathbf{y})$ results in the worst outcome and thereby the first criterion of the MMF model. In other words, although a direct use of the maximum semideviation in the mean-equity model may contradict the outcome maximization, the measure can be used complementary to the mean leading us to the worst outcome criterion which does not contradict the outcome maximization. This construction can be generalized for various (dispersion type) inequality measures. Moreover, we allow the measures to be scaled with any positive factor $\alpha > 0$, in order to avoid creation of new inequality measures as one could consider $\rho_\alpha(X) = \alpha\rho(X)$ as a different inequality measure. For any inequality measure ρ we introduce the corresponding underachievement function defined as the difference of the mean outcome and the (scaled) inequality measure itself, i.e.

$$M_{\alpha\rho}(\mathbf{y}) = \mu(\mathbf{y}) - \alpha\rho(\mathbf{y}). \tag{21}$$

This allows us to replace the original mean-equity bicriteria optimization (20) with the following bicriteria problem:

$$\max\{(\mu(\mathbf{f}(\mathbf{x})), \mu(\mathbf{f}(\mathbf{x})) - \alpha\rho(\mathbf{f}(\mathbf{x})))\ :\ \mathbf{x}\in Q\} \tag{22}$$

where the second objective represents the corresponding underachievement measure $M_{\alpha\rho}(\mathbf{y})$ (21). Note that for any inequality measure $\rho(\mathbf{y}) \geq 0$ one gets $M_{\alpha\rho}(\mathbf{y}) \leq \mu(\mathbf{y})$ thus really expressing underachievements (comparing to mean) from the perspective of outcomes being maximized.

We will say that an inequality measure ρ is *fairly* α*-consistent* if

$$\mathbf{y}' \succeq_e \mathbf{y}'' \ \Rightarrow\ \mu(\mathbf{y}') - \alpha\rho(\mathbf{y}') \geq \mu(\mathbf{y}'') - \alpha\rho(\mathbf{y}'') \tag{23}$$

The relation of fair α-consistency will be called *strong* if, in addition to (23), the following holds

$$\mathbf{y}' \succ_e \mathbf{y}'' \ \Rightarrow\ \mu(\mathbf{y}') - \alpha\rho(\mathbf{y}') > \mu(\mathbf{y}'') - \alpha\rho(\mathbf{y}''). \tag{24}$$

Theorem 1. *If the inequality measure $\rho(\mathbf{y})$ is fairly α-consistent (23), then except for outcomes with identical values of $\mu(\mathbf{y})$ and $\rho(\mathbf{y})$, every efficient solution of the bicriteria problem (22) is a fairly efficient allocation pattern. In the case of strong consistency (24), every allocation pattern $\mathbf{x}\in Q$ efficient to (22) is, unconditionally, fairly efficient.*

Proof. Let $\mathbf{x}^0 \in Q$ be an efficient solution of (22). Suppose that $\mathbf{x}^0$ is not fairly efficient. This means, there exists $\mathbf{x}\in Q$ such that $\mathbf{y} = \mathbf{f}(\mathbf{x}) \succ_e \mathbf{y}^0 = \mathbf{f}(\mathbf{x}^0)$. Then, it follows $\mu(\mathbf{y}) \geq \mu(\mathbf{y}^0)$, and simultaneously $\mu(\mathbf{y}) - \alpha\rho(\mathbf{y}) \geq \mu(\mathbf{y}^0) - \alpha\rho(\mathbf{y}^0)$, by virtue of the fair α-consistency (23). Since $\mathbf{x}^0$ is efficient to (22) no inequality can be strict, which implies $\mu(\mathbf{y}) = \mu(\mathbf{y}^0)$ and and $\rho(\mathbf{y}) = \rho(\mathbf{y}^0)$.

In the case of the strong fair α-consistency (24), the supposition $\mathbf{y} = \mathbf{f}(\mathbf{x}) \succ_e \mathbf{y}^0 = \mathbf{f}(\mathbf{x}^0)$ implies $\mu(\mathbf{y}) \geq \mu(\mathbf{y}^0)$ and $\mu(\mathbf{y}) - \alpha\rho(\mathbf{y}) > \mu(\mathbf{y}^0) - \alpha\rho(\mathbf{y}^0)$ which contradicts the efficiency of $\mathbf{x}^0$ with respect to (22). Hence, the allocation pattern $\mathbf{x}^0$ is fairly efficient. □

4 FAIR CONSISTENCY CONDITIONS

Typical dispersion type inequality measures are convex, i.e. $\rho(\lambda\mathbf{y}' + (1-\lambda)\mathbf{y}'') \leq \lambda\rho(\mathbf{y}') + (1-\lambda)\rho(\mathbf{y}'')$ for any $\mathbf{y}', \mathbf{y}''$ and $0 \leq \lambda \leq 1$. Certainly, the underachievement function $M_{\alpha\rho}(\mathbf{y})$ must be also monotonic for the fair consistency which enforces more restrictions on the inequality measures. We will show further that convexity together with positive homogeneity and some boundedness of an inequality measure is sufficient to guarantee monotonicity of the corresponding underachievement measure and thereby to guarantee the fair α-consistency of inequality measure itself.

We say that (dispersion type) inequality measure $\rho(\mathbf{y}) \geq 0$ is Δ-*bounded* if it is upper bounded by the maximum downside deviation, i.e.,

$$\rho(\mathbf{y}) \leq \Delta(\mathbf{y}) \quad \forall \mathbf{y}. \tag{25}$$

Moreover, we say that $\rho(\mathbf{y}) \geq 0$ is strictly Δ-bounded if inequality (25) is a strict bound, except from the case of perfectly equal outcomes, i.e., $\rho(\mathbf{y}) < \Delta(\mathbf{y})$ for any $\mathbf{y}$ such that $\Delta(\mathbf{y}) > 0$.

Theorem 2. *Let $\rho(\mathbf{y}) \geq 0$ be a convex, positively homogeneous and translation invariant (dispersion type) inequality measure. If $\alpha\rho(\mathbf{y})$ is Δ-bounded, then $\rho(\mathbf{y})$ is fairly α-consistent in the sense of (23).*

Proof. The relation of fair dominance $\mathbf{y}' \succeq_e \mathbf{y}''$ denotes that there exists a finite sequence of vectors $\mathbf{y}^0 = \mathbf{y}'', \mathbf{y}^1, \ldots, \mathbf{y}^t$ such that $\mathbf{y}^k = \mathbf{y}^{k-1} - \varepsilon_k \mathbf{e}_{i'} + \varepsilon_k \mathbf{e}_{i''}$, $0 \leq \varepsilon_k \leq y^{k-1}_{i'} - y^{k-1}_{i''}$ for $k = 1, 2, \ldots, t$ and there exists a permutation π such that $y'_{\pi(i)} \geq y^t_i$ for all $i \in I$. Note that the underachievement function $M_{\alpha\rho}(\mathbf{y})$, similar as $\rho(\mathbf{y})$ depends only on the distribution of outcomes. Further, if $\mathbf{y}' \geq \mathbf{y}''$, then $\mathbf{y}' = \mathbf{y}'' + (\mathbf{y}' - \mathbf{y}'')$ and $\mathbf{y}' - \mathbf{y}'' \geq 0$. Hence, due to concavity and positive homogeneity, $M_{\alpha\rho}(\mathbf{y}') \geq M_{\alpha\rho}(\mathbf{y}'') + M_{\alpha\rho}(\mathbf{y}' - \mathbf{y}'')$. Moreover, due to the bound (25), $M_{\alpha\rho}(\mathbf{y}' - \mathbf{y}'') \geq \mu(\mathbf{y}' - \mathbf{y}'') - \Delta(\mathbf{y}' - \mathbf{y}'') \geq \mu(\mathbf{y}' - \mathbf{y}'') - \mu(\mathbf{y}' - \mathbf{y}'') = 0$. Thus, $M_{\alpha\rho}(\mathbf{y})$ satisfies also the requirement of monotonicity. Hence, $M_{\alpha\rho}(\mathbf{y}') \geq M_{\alpha\rho}(\mathbf{y}^t)$. Further, let us notice that $\mathbf{y}^k = \lambda\bar{\mathbf{y}}^{k-1} + (1-\lambda)\mathbf{y}^{k-1}$ where $\bar{\mathbf{y}}^{k-1} = \mathbf{y}^{k-1} - (y_{i'} - y_{i''})\mathbf{e}_{i'} + (y_{i'} - y_{i''})\mathbf{e}_{i''}$ and $\lambda = \varepsilon/(y_{i'} - y_{i''})$. Vector $\bar{\mathbf{y}}^{k-1}$ has the same distribution of coefficients as $\mathbf{y}^{k-1}$ (actually it represents results of swapping $y_{i'}$ and $y_{i''}$). Hence, due to concavity of $M_{\alpha\rho}(\mathbf{y})$, one gets $M_{\alpha\rho}(\mathbf{y}^k) \geq \lambda M_{\alpha\rho}(\bar{\mathbf{y}}^{k-1}) + (1-\lambda)M_{\alpha\rho}(\mathbf{y}^{k-1}) = M_{\alpha\rho}(\mathbf{y}^{k-1})$. Thus, $M_{\alpha\rho}(\mathbf{y}') \geq M_{\alpha\rho}(\mathbf{y}'')$ which justifies the fair α-consistency of $\rho(\mathbf{y})$. □

For strong fair α-consistency some strict monotonicity and concavity properties of the underachievement function are needed. Obviously, there does not exist any inequality measure which is positively homogeneous and simultaneously strictly convex. However, one may notice from the proof of Theorem 2 that only convexity properties on equally distributed outcome vectors are important for monotonous underachievement functions.

We say that inequality measure $\rho(\mathbf{y}) \geq 0$ is *strictly convex on equally distributed outcome vectors*, if

$$\rho(\lambda\mathbf{y}' + (1-\lambda)\mathbf{y}'') < \lambda\rho(\mathbf{y}') + (1-\lambda)\rho(\mathbf{y}'')$$

for $0 < \lambda < 1$ and any two vectors $\mathbf{y}' \neq \mathbf{y}''$ representing the same outcomes distribution as some $\mathbf{y}$, i.e., $\mathbf{y}' = (y_{\pi'(1)}, \ldots, y_{\pi'(m)})$ π' and $\mathbf{y}'' = (y_{\pi''(1)}, \ldots, y_{\pi''(m)})$ for some permutations π' and π'', respectively.

Theorem 3. *Let $\rho(\mathbf{y}) \geq 0$ be a convex, positively homogeneous and translation invariant (dispersion type) inequality measure. If $\rho(\mathbf{y})$ is also strictly convex on equally distributed outcomes and $\alpha\rho(\mathbf{y})$ is strictly Δ-bounded, then the measure $\rho(\mathbf{y})$ is fairly strongly α-consistent in the sense of (24).*

Proof. The relation of weak fair dominance $\mathbf{y}' \succeq_e \mathbf{y}''$ denotes that there exists a finite sequence of vectors $\mathbf{y}^0 = \mathbf{y}'', \mathbf{y}^1, \ldots, \mathbf{y}^t$ such that $\mathbf{y}^k = \mathbf{y}^{k-1} - \varepsilon_k \mathbf{e}_{i'} + \varepsilon_k \mathbf{e}_{i''}$, $0 \leq \varepsilon_k \leq y^{k-1}_{i'} - y^{k-1}_{i''}$ for $k = 1, 2, \ldots, t$ and there exists a permutation π such that $y'_{\pi(i)} \geq y^t_i$ for all $i \in I$. The strict fair dominance $\mathbf{y}' \succ_e \mathbf{y}''$ means that $y'_{\pi(i)} > y^t_i$ for some $i \in I$ or at least one ε_k is strictly positive. Note that the underachievement function $M_{\alpha\rho}(\mathbf{y})$ is strictly monotonous and strictly convex on equally distributed outcome vectors. Hence, $M_{\alpha\rho}(\mathbf{y}') > M_{\alpha\rho}(\mathbf{y}'')$ which justifies the fair strong α-consistency of the measure $\rho(\mathbf{y})$. □

The specific case of fair 1-consistency is also called *the mean-complementary fair consistency*. Note that the fair $\bar{\alpha}$-consistency of measure $\rho(\mathbf{y})$ actually guarantees the mean-complementary fair consistency of measure $\alpha\rho(\mathbf{y})$ for all $0 < \alpha \leq \bar{\alpha}$, and the same remain valid for the strong consistency properties. It follows from a possible expression of $\mu(\mathbf{y}) - \alpha\rho(\mathbf{y})$ as the convex combination of $\mu(\mathbf{y}) - \bar{\alpha}\rho(\mathbf{y})$ and $\mu(\mathbf{y})$. Hence, for any $\mathbf{y}' \succeq_e \mathbf{y}''$, due to $\mu(\mathbf{y}') \geq \mu(\mathbf{y}'')$ one gets $\mu(\mathbf{y}') - \alpha\rho(\mathbf{y}') \geq \mu(\mathbf{y}'') - \alpha\rho(\mathbf{y}'')$ in the case of the fair $\bar{\alpha}$-consistency of measure $\rho(\mathbf{y})$ (or respective strict inequality in the case of strong consistency). Therefore, while analyzing specific inequality measures we seek the largest values α guaranteeing the corresponding fair consistency.

As mentioned, typical inequality measures are convex and many of them are positively homogeneous. Moreover, the measures such as the mean absolute (downside) semideviation $\bar{\delta}(\mathbf{y})$ (18), the standard downside semideviation $\bar{\sigma}(\mathbf{y})$ (19), and the mean absolute difference $\Gamma(\mathbf{y})$ (12) are Δ-bounded. Indeed,

one may easily notice that $\mu(\mathbf{y}) - y_i \leq \Delta(\mathbf{y})$ and therefore $\bar{\delta}(\mathbf{y}) \leq \frac{1}{m}\sum_{i\in I}\Delta(\mathbf{y}) = \Delta(\mathbf{y})$, $\bar{\sigma}(\mathbf{y}) \leq \sqrt{\Delta(\mathbf{y})^2} = \Delta(\mathbf{y})$ and $\Gamma(\mathbf{y}) = \frac{1}{m^2}\sum_{i\in I}\sum_{j\in I}(\max\{y_i, y_j\} - \mu(\mathbf{y})) \leq \Delta(\mathbf{y})$. Actually, all these inequality measures are strictly Δ-bounded since for any unequal outcome vector at least one outcome must be below the mean thus leading to strict inequalities in the above bounds. Obviously, Δ-bounded (but not strictly) is also the maximum absolute downside deviation $\Delta(\mathbf{y})$ itself. This allows us to justify the maximum downside deviation $\Delta(\mathbf{y})$ (17), the mean absolute (downside) semideviation $\bar{\delta}(\mathbf{y})$ (18), the standard downside semideviation $\bar{\sigma}(\mathbf{y})$ (19) and the mean absolute difference $\Gamma(\mathbf{y})$ (12) as fairly 1-consistent (mean-complementary fairly consistent) in the sense of (23).

We emphasize that, despite the standard semideviation is a fairly 1-consistent inequality measure, the consistency is not valid for variance, semivariance and even for the standard deviation. These measures, in general, do not satisfy the all assumptions of Theorem 2. Certainly, we have enumerated only the simplest inequality measures studied in the resource allocation context which satisfy the assumptions of Theorem 2 and thereby they are fairly 1-consistent. Theorem 2 allows one to show this property for many other measures. In particular, one may easily find out that any convex combination of fairly α-consistent inequality measures remains also fairly α-consistent. On the other hand, among typical inequality measures the mean absolute difference seems to be the only one meeting the stronger assumptions of Theorem 3 and thereby maintaining the strong consistency.

As mentioned, the mean absolute deviation is twice the mean absolute downside semideviation which means that $\alpha\bar{\delta}(\mathbf{y})$ is Δ-bounded for any $0 < \alpha \leq 0.5$. The symmetry of mean absolute semideviations $\bar{\delta}(\mathbf{y}) = \sum_{i\in I}(y_i - \mu(\mathbf{y}))_+ = \sum_{i\in I}(\mu(\mathbf{y}) - y_i)_+$ can be also used to derive some Δ-boundedness relations for other inequality measures. In particular, one may find out that for m-dimensional outcome vectors of unweighted problem, any downside semideviation from the mean cannot be larger than $m-1$ upper semideviations. Hence, the maximum absolute deviation satisfies the inequality $\frac{1}{m-1}R(\mathbf{y}) \leq \Delta(\mathbf{y})$, while the maximum absolute difference fulfills $\frac{1}{m}d(\mathbf{y}) \leq \Delta(\mathbf{y})$. Similarly, for the standard deviation one gets $\frac{1}{\sqrt{m-1}}\bar{\delta}(\mathbf{y}) \leq \Delta(\mathbf{y})$. Actually, $\alpha\bar{\sigma}(\mathbf{y})$ is strictly Δ-bounded for any $0 < \alpha \leq 1/\sqrt{m-1}$ since for any unequal outcome vector at least one outcome must be below the mean thus leading to strict inequalities in the above bounds. These allow us to justify the mean absolute semideviation with $0 < \alpha \leq 0.5$, the maximum absolute deviation with $0 < \alpha \leq \frac{1}{m-1}$, the maximum absolute difference with $0 < \alpha \leq \frac{1}{m}$ and the standard deviation with $0 < \alpha \leq \frac{1}{\sqrt{m-1}}$ as fairly α-consistent within the specified intervals of α. Moreover, the α-consistency of the standard deviation is strong.

Table 1: Fair consistency results.

Measure			α–consistency	
Mean abs. semidev.	$\bar{\delta}(\mathbf{y})$	(18)	1	
Mean abs. dev.	$\bar{\delta}(\mathbf{y})$	(14)	0.5	
Max. semidev.	$\Delta(\mathbf{y})$	(17)	1	
Max. abs. dev.	$R(\mathbf{y})$	(15)	$\frac{1}{m-1}$	
Mean abs. diff.	$\Gamma(\mathbf{y})$	(12)	1	strong
Max. abs. diff.	$d(\mathbf{y})$	(13)	$\frac{1}{m}$	
Standard semidev.	$\bar{\sigma}(\mathbf{y})$	(19)	1	
Standard dev.	$\sigma(\mathbf{y})$	(16)	$\frac{1}{\sqrt{m-1}}$	strong

The fair consistency results for basic dispersion type inequality measures considered in resource allocation problems are summarized in Table 1 where α values are given and the strong consistency is indicated. Table 1 points out how the inequality measures can be used in resource allocation models to guarantee their harmony both with outcome maximization (Pareto-optimality) and with inequalities minimization (Pigou-Dalton equity theory). Exactly, for each inequality measure applied with the corresponding value α from Table 1 (or smaller positive value), every efficient solution of the bicriteria problem (22), ie. $\max\{(\mu(\mathbf{f}(\mathbf{x})), \mu(\mathbf{f}(\mathbf{x})) - \alpha\rho(\mathbf{f}(\mathbf{x}))) : \mathbf{x} \in Q\}$, is a fairly efficient allocation pattern, except for outcomes with identical values of $\mu(\mathbf{y})$ and $\rho(\mathbf{y})$. In the case of strong consistency (as for mean absolute difference or standard deviation), every solution $\mathbf{x} \in Q$ efficient to (22) is, unconditionally, fairly efficient.

The consistency results summarized in Table 1 are sufficient conditions. This means that whenever the α limit is observed the corresponding consistency relation is valid for any problem. It may happen that for a specific problem instance and a specific inequality measure the fair consistency is valid for larger values of α. Nevertheless, we have provided strict bounds in the sense that for a larger value of α there exists a resource allocation problem on which the fair consistency is not valid, and the bicriteria problem (22) may generate dominated solution.

5 CONCLUSIONS

The problems of efficient and fair resource allocation arise in various systems which serve many users. Fairness is, essentially, an abstract socio-political concept that implies impartiality, justice and equity. Neverthe-

less, in operations research it was quantified with various solution concepts (Denda et al., 2000). The equitable optimization with the preference structure that complies with both the efficiency (Pareto-optimality) and with the Pigou-Dalton principle of transfers may be used to formalize the fair solution concepts. Multiple criteria models equivalent to equitable optimization allows to generate a variety of fair and efficient resource allocation patterns (Kostreva et al., 2004; Ogryczak et al., 2008).

In this paper we have analyzed how scalar inequality measures can be used to guarantee the fair consistency. It turns out that several inequality measures can be combined with the mean itself into the optimization criteria generalizing the concept of the worst outcome and generating fairly consistent underachievement measures. We have shown that properties of convexity and positive homogeneity together with being bounded by the maximum downside semideviation are sufficient for a typical inequality measure to guarantee the corresponding fair consistency. It allows us to identify various inequality measures which can be effectively used to incorporate fairness factors into various resource allocation problems while preserving the consistency with outcomes maximization. Among others, the standard semideviation and the mean semideviation turn out to be such a consistent inequality measure while the mean absolute difference is strongly consistent.

Our analysis is related to the properties of solutions to resource allocation models. It has been shown how inequality measures can be included into the models avoiding contradiction to the maximization of outcomes. We do not analyze algorithmic issues for the specific resource allocation problems. Generally, the requirement of convexity necessary for the consistency, guarantees that the corresponding optimization criteria belong to the class of convex optimization, not complicating the original resource allocation model with any additional discrete structure. Many of the inequality measures, we analyzed, can be implemented with auxiliary linear programming constraints. Nevertheless, further research on efficient computational algorithms for solving the specific models is necessary.

REFERENCES

Atkinson, A. (1970). On the measurement of inequality. *J. Economic Theory*, 2:244–263.

Bonald, T. and Massoulie, L. (2001). Impact of fairness on internet performance. In *Proceedings of ACM Sigmetrics*, pages 82–91.

Denda, R., Banchs, A., and Effelsberg, W. (2000). The fairness challenge in computer networks. *Lect. Notes Comp. Sci.*, 1922:208–220.

Ibaraki, T. and Katoh, N. (1988). *Resource Allocation Problems, Algorithmic Approaches*. MIT Press, Cambridge.

Kelly, F., Mauloo, A., and Tan, D. (1997). Rate control for communication networks: shadow prices, proportional fairness and stability. *J. Opnl. Res. Soc.*, 49:206–217.

Kleinberg, J., Rabani, Y., and Tardos, E. (2001). Fairness in routing and load balancing. *J. Comput. Syst. Sci.*, 63:2–21.

Kostreva, M. M. and Ogryczak, W. (1999). Linear optimization with multiple equitable criteria. *RAIRO Operations Research*, 33:275–297.

Kostreva, M. M., Ogryczak, W., and Wierzbicki, A. (2004). Equitable aggregations and multiple criteria analysis. *Eur. J. Opnl. Res.*, 158:362–377.

Luss, H. (1999). On equitable resource allocation problems: A lexicographic minimax approach. *Operations Research*, 47:361–378.

Marchi, E. and Oviedo, J. A. (1992). Lexicographic optimality in the multiple objective linear programming: the nucleolar solution. *Eur. J. Opnl. Res.*, 57:355–359.

Ogryczak, W. (2000). Inequality measures and equitable approaches to location problems. *Eur. J. Opnl. Res.*, 122:374–391.

Ogryczak, W., Pióro, M., and Tomaszewski, A. (2005). Telecommunications network design and max-min optimization problem. *J. Telecom. Info. Tech.*, 3:1–14.

Ogryczak, W. and Śliwiński, T. (2006). On direct methods for lexicographic min-max optimization. *Lect. Notes Comp. Sci.*, 3982:774–783.

Ogryczak, W., Wierzbicki, A., and Milewski, M. (2008). A multi-criteria approach to fair and efficient bandwidth allocation. *OMEGA*, 36:451–463.

Pióro, M. and Medhi, D. (2004). *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann, San Francisco.

Rawls, J. (1971). *The Theory of Justice*. Harvard Univ. Press, Cambridge.

Rawls, J. and Kelly, E. (2001). *Justice as Fairness: A Restatement*. Harvard Univ. Press, Cambridge.

Rothschild, M. and Stiglitz, J. E. (1973). Some further results in the measurement of inequality. *J. Econ. Theory*, 6:188–204.

Young, H. P. (1994). *Equity in Theory and Practice*. Princeton Univ. Press, Princeton.

RECONFIGURATION OF EMBEDDED SYSTEMS*

Mohamed Khalgui, Martin Hirsch, Dirk Missal and Hans-Michael Hanisch
Martin Luther University, Halle, Germany
http://at.informatik.uni-halle.de

Keywords: Industrial Embedded Systems, IEC61499 Function Blocks, Reconfiguration, Model-checking.

Abstract: This paper deals with automatic reconfiguration of embedded systems following the Component-based Standard IEC61499. First of all, we propose a new reconfiguration semantic allowing the improvement of the system performance even if there is no hardware fault. In addition, we characterize all possible reconfiguration forms in order to cover all possible execution scenarios and to apply an automatic reconfiguration, we define thereafter an agent-based architecture that we model with nested state machines to control the design complexity.

1 INTRODUCTION

Nowadays, the new generation of industrial control systems is addressing new criteria as flexibility and agility. To reduce their cost, these systems have to be changed and adapted to their environment without any disturbance. Several academic and industrial research works have been made to develop reconfigurable systems. We distinguish in these works two reconfiguration policies: the static and dynamic reconfigurations. The static reconfiguration is applied off-line to apply changes before the system cold start (Angelov et al., 2005), whereas the dynamic reconfiguration is applied dynamically at run-time. In the last policy, two cases exist: the manual reconfiguration applied by the user (Rooker et al., 2007) and the automatic reconfiguration applied by an intelligent agent localized in the system (Al-Safi and Vyatkin, 2007).

In this paper, we are interested in the automatic reconfiguration of industrial control systems which have to satisfy according to user requirements functional and temporal properties but their time to market has to be shorter than ever. To satisfy all these requirements, we use the component-based methodology supporting the modularity as well as the reusability of already developed components. Today, several rich technologies have been proposed to develop component-based manufacturing systems and among all these technologies, the standard IEC61499 is proposed by the International Electrotechnical Commission (IEC) to design the application as well as the corresponding execution environment. A Function Block (FB) is a unit of software supporting functionalities of an application and it is composed of an interface and an implementation where the interface contains data/event inputs and outputs supporting interactions with the environment. Events are responsible for the activation of the block while data contain valued information. The implementation contains algorithms to execute when the corresponding events occur. The selection of an algorithm to execute is performed by a state machine called the Execution Control Chart (ECC). The ECC is also responsible for sending output events at the end of the algorithm execution. A control application is specified by a network of FBs such as each event input (resp. output) of a block is linked to an event output (resp. input) by a channel and it corresponds otherwise to a global input (resp. output). Data inputs and outputs follow the same rules. Today, several research works on this standard have been proposed (Khalgui and Thramboulidis, 2008), rich tools [2] and several industrial platforms[3] have been developed while following this standard.

In this paper, we aim to develop reconfigurable IEC61499 control systems. Several research works

*This work is supported by the Humboldt foundation in Germany under the reference TUN1127196STP.

[2] http://www.isagraf.com, http://www.holobloc.com
[3] www.itia.cnr.it

dealing with the reconfiguration of these systems have been proposed (Angelov et al., 2005; Rooker et al., 2007; Al-Safi and Vyatkin, 2007) but they are limited to particular cases (e.g. to resolve hardware faults or to add new functionalities like the update of an algorithm in a block) and they do not address all the reconfiguration reasons that can possibly occur in industry. We define a new semantic of the reconfiguration to improve the system performance even if there is no hardware fault and we characterize all possible reconfiguration forms that we can apply on a system. To apply an automatic reconfiguration at runtime, we propose thereafter an agent-based architecture to handle reconfiguration scenarios bringing the system into safe states. We model the agent according to well suited formal formalism: the Net Condition/Event Systems (denoted NCES) proposed by Rausch and Hanisch in (Rausch and Hanisch, 1995).

We present in the next section the NCES formalism and in Section3 the EnAS production system to be used as an example in the paper. We define in Section4 a new semantic of the reconfiguration before we detail in Section5 all reconfiguration forms. Finally before conclusions, we present in Section6 the Agent-based architecture to handle automatic reconfigurations.

2 NCES FORMALISM

The formalism of Net Condition/Event Systems (NCES) is an extension of the well known Petri net formalism. It was introduced by Rausch and Hanisch in (Rausch and Hanisch, 1995) and further developed through the last years according to which a NCES is a place-transition net formally represented by a tuple:

$$NCES = (P, T, F, CN, EN, C^{in}, E^{in}, C^{out}, E^{out},$$
$$B_c, B_e, C_s, D_t, m_0) \text{ where,}$$

(1) P (resp, T) is a non-empty finite set of places (resp, transitions), (2) F is the set of flow arcs, $F : (PXT) \bigcup (TXP)$, (3) CN (resp, EN) is the set of condition (resp, event) arcs, $CN \subseteq (PXT)$ (resp, $EN \subseteq (TXT)$), (4) C^{in} (resp, E^{in}) is the set of condition (resp, event) inputs, (5) C^{out} (resp, E^{out}) is the set of condition (resp, event) outputs, (6) B_c (resp, B_e) is the set of condition (resp, event) input arcs in a NCES module, (7) $B_c \subseteq (C^{in}XT)$ (resp, $B_e \subseteq (E^{in}XT)$), (8) Cs (resp, Dt) is the set of condition (resp, event) output arcs, (9) $Cs \subseteq (PXE^{out})$ (resp, $Dt \subseteq (TXE^{out})$), (10) $m_0 : P \rightarrow 0,1$ is the initial marking.

The semantics of NCES are defined by the firing rules of transitions. There are several conditions to be fulfilled to enable a transition to fire. First, as it is in ordinary Petri nets, an enabled transition has to have a token concession. That means that all pre-places have to be marked with at least one token. In addition to the flow arcs from places, a transition in NCES may have incoming condition arcs from places and event arcs from other transitions. A transition is enabled by condition signals if all source places of the condition signals are marked by at least one token. The other type of influence on the firing can be described by event signals which come to the transition from some other transitions. Transitions having no incoming event arcs are called *spontaneous*, otherwise *forced*. A forced transition is enabled if it has token concession and it is enabled by condition and event signals (Rausch and Hanisch, 1995).

We note finally that the model-checker SESA is a useful tool to verify functional and temporal properties on NCES (Rausch and Hanisch, 1995). We apply it in our work to verify reconfigurable control systems following the standard IEC 61499.

3 INDUSTRIAL CASE STUDY

In this paper, we are interested in the manufacturing system EnAS used as a demonstrator at the Martin Luther University of Halle (in Germany). This system is implemented while following the standard IEC61499 and it allows the transportation of pieces from the production into storing units. The pieces shall be placed inside tins to close with caps afterwards[4]. Two different production strategies can be applied : we place in each tin one or two pieces according to production rates of pieces, tins and caps. In this paper, we denote respectively by nb_{pieces}, $nb_{tins+caps}$ the production number of pieces, tins (as well as caps) per hour. In the following, we denote by $Threshold$ a variable (defined in specifications) allowing to choose the adequate production strategy.

The EnAS system is mainly composed of a belt, two Jack stations (J_1 and J_2) and two Gripper stations (G_1 and G_2) (Figure 1). The Jack stations place new produced pieces and to close tins with caps, whereas the Gripper stations remove charged tins from the belt into the storing units.

Initially, the belt moves a particular pallet containing a tin and a cap into the first Jack station J_1. According to the production parameters, we distinguish two cases,

[4]For detailed information on the EnAS system, we refer the reader to our group website: http://at.informatik.uni-halle.de

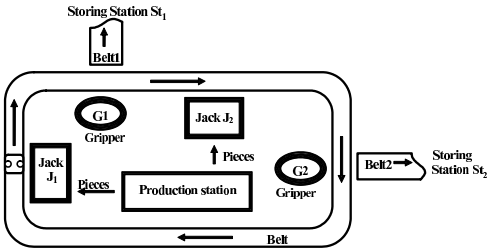


Figure 1: Distribution of the EnAS stations.

- **The First Production Policy ($nb_{pieces}/nb_{tins+caps} \leq Threshold$):** the Jack station J_1 places from the production station a new piece and closes the tin with the cap. In this case, the Gripper station G_1 removes the tin from the belt into the storing station St_1.
- **The Second Production Policy ($nb_{pieces}/nb_{tins+caps} > Threshold$):** the Jack station J_1 places just a piece in the tin which is moved thereafter into the second Jack station to place a second new piece. Once J_2 closes the tin with a cap, the belt moves the pallet into the Gripper station G_2 to remove the tin (with two pieces) into the second storing station St_2. If we follow this policy, the productivity as well as the factory receipt is improved.

4 NEW RECONFIGURATION SEMANTIC

We define in this paper the reconfiguration semantic as follows.

Definition. *A dynamic reconfiguration is any change according to well-defined conditions in the software as well as in the hardware components to lead the system into a better safe state at run-time.*

According to the standard IEC61499, we mean in this definition by a change in the software components any operation allowing to add, to remove or to update Function Blocks to improve the behavior of all the system. On the other hand, we mean by a change in the hardware components any operation allowing to add, remove or update devices used in the execution environment.

Running Example. *In the EnAS system, we apply the reconfiguration for two reasons: (i) to save the system when hardware problems occur at run-time. For example, when the Gripper G2 is broken, then we have only to follow the first production policy by placing only one piece in each tin. (ii) to improve the production gain when ($nb_{pieces}/nb_{tins+caps} > Threshold$). In this case, we have to apply the second policy to improve the factory receipt. Therefore, we have to apply changes in the system architecture and blocks to follow this policy. In this example, the reconfiguration is not only applied to resolve hardware problems as proposed in (Al-Safi and Vyatkin, 2007) but also to improve the system performance by increasing the production gain. This new semantic of the reconfiguration concept will be a future issue in the manufacturing industry.*

5 RECONFIGURATION CASES

We classify in this section all reconfiguration forms to possibly apply on a control system in order to cover all possible reasons described above. We distinguish the following forms:

- **First Form.** It deals with the change of the application architecture that we consider as a composition of Function Blocks. In this case, we have possibly to add, to remove or also to change the localization of Function Blocks (from one to another device). This reconfiguration form requires to load new (or to unload old, resp) blocks in (from, resp) the memory.

  Running Example. *We distinguish in EnAS two architectures: (i) We implement the system with the first architecture when we follow the first production policy. In this case, we load in the memory the Function Blocks $J1\_CTL$, $Belt\_CTL$ and $G1\_CTL$. (ii) We implement the system with the second architecture when we follow the second production policy. In this case, we load in the memory the Function Blocks $J1\_CTL$, $J2\_CTL$, $Belt\_CTL$ and $G2\_CTL$. If we follow the first production policy and $nb_{pieces}/nb_{tins+caps}$ becomes higher than $Threshold$, then we have to load the function block $G2\_CTL$ in the memory to follow the second production policy.*

- **Second Form.** it deals with the reconfiguration of the application without changing its architecture (e.g. without loading or unloading Function Blocks). In this case, we apply changes on the internal structure of blocks or on their composition as follows: (a) we change the ECC structure, (b) we add, update or remove data/events inputs/outputs, (c) we update algorithms, (d) we

change the configuration of connections between blocks.

Running Example. *In the EnAS system, if we follow the second policy and the Jack station J2 is broken, then we have to change the internal behavior (e.g. the ECC structure) of the block J1_CTL to close the tin with a cap once it places only one piece. The tin will be moved directly thereafter to the Gripper G2. In this example, we do not change the application architecture (e.g. loading or unloading blocks) but we just change the behavior of particular blocks.*

- **Third Form.** it simply deals with the reconfiguration of the application data (e.g. internal data of blocks or global data of the system). The reconfiguration in this case is easy to apply.

  Running Example. *In the EnAS system, if a hardware problem occurs at run-time, we propose to change the value of Threshold to a great number max_value. In this case we will not be interested in the performance improvement but in the rescue of the system to guarantee a minimal level of productivity.*

Finally, this classification covers all possible reconfiguration forms to dynamically bring a manufacturing system into a safe and better state while satisfying the user requirements and the environment changes.

6 AUTOMATIC RECONFIGURATION OF MANUFACTURING SYSTEMS

To apply an automatic reconfiguration, we define in this section an agent-based architecture of a control system where the Agent checks the environment evolution and takes also into account the user requirements to apply reconfiguration scenarios on the system. To control the design complexity, we specify this agent with nested NCES supporting the different reconfiguration forms presented above. We specify also the possible behaviors of the system with NCES in order to verify with SESA functional and temporal properties described in user requirements.

6.1 Architecture of the Reconfiguration Agent

According to the reconfiguration forms proposed above, we define the agent behavior with the following units belonging to three levels:

- **First Level:** (**Architecture Unit**) this unit checks the system behavior and changes its architecture (add/remove Function Blocks) if particular conditions are satisfied. We note that *Standardized Manager Function Blocks* are used in this unit to load or unload such blocks in the memory.

- **Second Level:** (**Control Unit**), for a particular *loaded* architecture, this unit checks the system behavior and : (i) reconfigures the blocks composition (e.g. changes the configuration of connections), (ii) adds/removes data/events inputs/outputs, (iii) reconfigures the internal behavior of blocks (e.g. modification of the ECC structure or the update of algorithms),

- **Third Level:**(**Data Unit**), this unit updates data if particular conditions are satisfied.

To control its complexity, we design the agent with nested state machines. In this case, the Architecture unit is specified by an Architecture State Machine (denoted by ASM) where each state corresponds to a particular architecture of the application. Therefore, each transition of the ASM corresponds to the load (resp, or unload) of Function Blocks in (resp, or from) the memory. We construct for each state S of the ASM a particular Control State Machine (denoted by CSM) in the Control unit. This state machine specifies all the reconfiguration forms to possibly apply when the system architecture corresponding to the state S is loaded (e.g. modification of the blocks composition or of their internal behavior). Each transition of any ASM has to be fired if particular conditions are satisfied. Finally, the Data unit is specified also by Data State Machines (denoted by DSMs) where each one corresponds to a state of a CSM or the ASM.

Running Example. *In the EnAS system, we design the agent with nested state machines as depicted in Figure 2. The first level is specified with the ASM where each state defines a particular architecture of the system (e.g. a particular FB composition to load in the memory). The state S_1 (resp, S_2) corresponds to the second (resp, first) policy where the stations J_1, J_2 and G_2 (resp, only J_1 and G_1) are loaded in the memory. We associate for each one of these states a CSM in the Control unit. Finally, the data unit is specified with a DSM defining the values that Threshold takes under well defined conditions. Note that if we follow the second production policy (state S_1) and the gripper G_2 is broken, then we have to change the policy and also the system architecture by loading the block*

*G1_CTL to remove pieces into Belt*1. *On the other hand, we associate in the second level for the state* S_1 *the CSM* CSM_1 *defining the different reconfiguration forms to apply when the first architecture is loaded in the memory. In particular, when the state* S_{11} *is active and the Jack station* J_1 *is broken, then we activate the state* S_{12} *in which the Jack station* J_2 *is alone running to place only one piece in the tin. In this case, the internal behavior of the block Belt_CTL has to be changed (e.g. the tin has to be transported directly to the station* J_2*). Finally, we specify in the data unit a DSM where we change the value of Threshold if the Gripper* G_1 *is broken (we suppose as an example that we are not interested in the system performance when the Gripper* G_1 *is broken).*

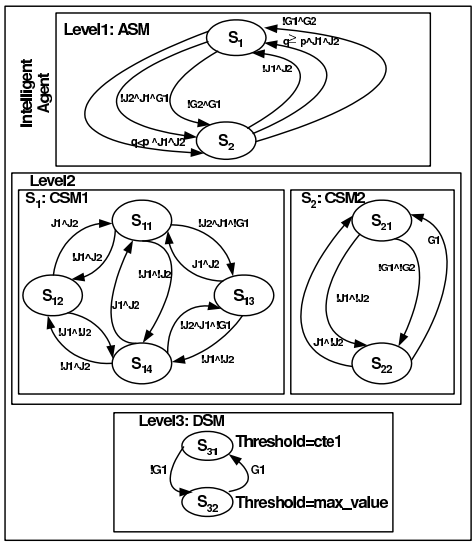


Figure 2: Behavior of the reconfiguration agent.

6.2 System Behaviors

The different reconfiguration scenarios applied by the agent define all the possible configurable behaviors to follow by the system blocks when conditions are satisfied. We specify in this paper these behaviors with a unique System State Machine (denoted by SSM) where each state corresponds to a particular behavior of a block when a corresponding input event occurs.

Running Example. *In the EnAS system, we specify in Figure 3 the different system behaviors that we can follow to resolve hardware problems or to improve the system performance. In this example, we distinguish 4 traces encoding 4 different behaviors. The trace* trace1 *implements the system behavior when the Jack station* J_1 *is broken. The trace* trace2 *implements the system behavior to apply the second production policy. The trace* trace3 *implements the system behavior when the Jack station* J_2 *is broken. Finally the last scenario implements the system behavior when the Gripper* G_2 *is broken or when we have to apply the first production policy. Note finally that each state corresponds to a particular behavior of a system block when the corresponding input event occurs.*

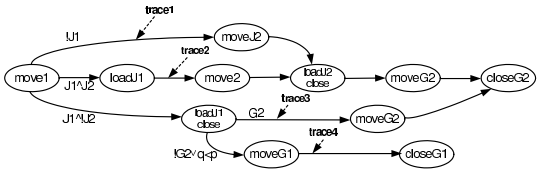


Figure 3: The system state machine: SSM.

6.3 Specification with Net Condition/event Systems

To specify the synchronisation between the agent and the system models, we apply the formalism NCES which provides useful facilities allowing such specification. We use in particular event/condition signals from the agent to fix the behavioral trace to follow in the SSM (e.g. a reconfiguration) and we use event signals to synchronize the agent state machines: ASM, CSM and DSM. Once the system is specified, we apply the *SESA* model-checker available at our laboratory to verify functional (like the attainability of states or the deadlock) and temporal properties.

Running Example. *We show in the Figure 4 the agent and system models according to the NCES formalism. When the Jack station* J_1 *is broken, the agent activates the place* P_{12} *and sends a condition signal to activate the trace* trace1 *in the system. Note that the architecture and control state machines are communicating by event signals to synchronize the agent behavior. Finally, the state "Well" represents a deadlock in the system when the Jack stations* J_1 *and* J_2 *are broken.*

7 CONCLUSIONS AND FUTURE WORKS

We propose in this paper a new reconfiguration semantic of embedded systems and a classification of all reconfiguration forms before we propose an agent-based architecture to take into account all reasons as

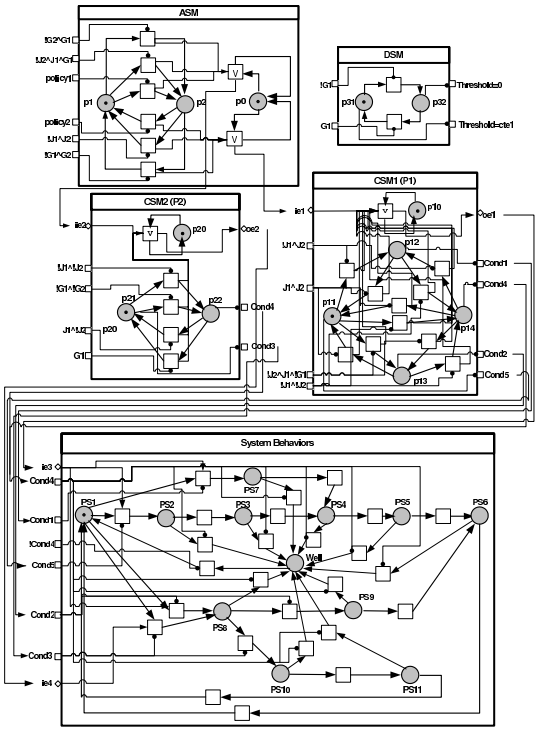


Figure 4: Design of the reconfigurable system with the NCES formalism.

well as reconfiguration forms. The agent applies automatic reconfigurations under fixed conditions, it is specified with nested NCES to reduce the design complexity and it is checked by the model checker SESA to verify functional and temporal properties. In our future works, we plan to propose an approach analyzing the schedulability of a system in the different reconfiguration scenarios in order to meet real-time constraints.

REFERENCES

Al-Safi, Y. and Vyatkin, V. (2007). An ontology-based reconfiguration agent for intelligent mechatronic systems. In *Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems*.

Angelov, C., Sierszecki, K., and Marian, N. (2005). Design models for reusable and reconfigurable state machines. In *L.T. Yang and All (Eds): EUC 2005, LNCS 3824, pp:152-163. International Federation for Information Processing.*

Khalgui, M. and Thramboulidis, K. (2008). An iec61499-based development approach with focus on the deployment of industrial control applications. In *Accepted in International Journal of Modelling, Identification and Control*.

Rausch, M. and Hanisch, H.-M. (1995). Net condition/event systems with multiple condition outputs. In *Symposium on Emerging Technologies and Factory Automation. Vol.1, pp.592-600.*

Rooker, M. N., Sunder, C., Strasser, T., Zoitl, A., Hummer, O., and Ebenhofer, G. (2007). Zero downtime reconfiguration of distributed automation systems : The εcedac approach. In *Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems*.

FEEDING A GENETIC ALGORITHM WITH AN ANT COLONY FOR CONSTRAINED OPTIMIZATION

An Application to the Unit Commitment Problem

Guillaume Sandou, Stéphane Font, Sihem Tebbani
Supelec Automatic Control Department, 3 rue Joliot Curie, 91192 Gif-sur-Yvette, France
Guillaume.Sandou@supelec.fr, Stephane.Font@supelec.fr, Sihem.Tebbani@supelec.fr

Arnaud Hiret, Christian Mondon
EDF Recherche et Développement, 6 quai Watier, 78401 Chatou, France
Arnaud.Hiret@supelec.fr, Christian.Mondon@supelec.fr

Keywords: Metaheuristics, unit commitment, ant colony, genetic algorithm, feasibility.

Abstract: In this paper, a new optimisation strategy for the solution of the classical Unit Commitment problem is proposed. This problem is known to be an often large scale, mixed integer programming problem. Due to high combinatorial complexity, the exact solution is often intractable. Thus, a metaheuristic based method has to be used to compute a very often suitable solution. The main idea of the approach is to use ant colony algorithm, to explicitly deal with the feasibility of the solution, and to feed a genetic algorithm whose goal is to intensively explore the search space. Finally, results show that the proposed method leads to the tractable computation of satisfying solutions for the Unit Commitment problem.

1 INTRODUCTION

The Unit Commitment problem is a classical optimization mixed integer problem, referring to the optimal scheduling computation of several production units while satisfying consumer's demand. Due to temporal coupling of constraints (time up and time down constraints), a long temporal horizon is required, implying a large number of binary variables. Numerous methods have already been applied to tackle the difficulties of the problem (Sen and Kothari, 1998). Roughly speaking, the following classification can be made: Exact methods (exhaustive enumeration, "Branch and Bound" (Chan and Wang, 1993), dynamic programming (Ouyang and Shahidehpour, 1991)); deterministic approximated methods (priority lists (Senjyu, *et al*., 2004)); Lagrangian relaxation (Zhai and Guan, 2002); Stochastic methods, also called metaheuristics (simulated annealing (Yin Wa Wong, 1998), tabu search (Rajan and Mohan, 2004), genetic algorithms (Swarup and Yamashiro, 2002)).

All these approaches have pros and cons: exact methods suffer from combinatorial complexity, deterministic approaches are very easy and tractable, but can be strongly suboptimal, Lagrangian relaxation allows taking into account constraints and can be used to medium scale cases, but, due to the non convexity of the objective function, no guarantee can be given on the actual optimality. In the case of metaheuristics methods, there is no guarantee on the actual optimality of the solution, but one can very often compute a very satisfying suboptimal with low computation times. This kind of methods is very interesting, especially for large scale cases. However, one of the problems of such stochastic methods is the management of the feasibility of solutions. As the algorithm "walks" randomly in the search space, there is no guarantee that the final solution is in the feasible set. This is particularly the case for the Unit Commitment problem, which is a strongly constrained problem.

In this paper a new optimization strategy is defined for the use of metaheuristics optimization methods, leading to a satisfying solution and the guarantee of its actual feasibility. The first step of the procedure is to use an ant colony algorithm as a "feasible solutions generator". As such an algorithm is constructive it is possible to explicitly manage the

constraints of the problem to create a set of feasible solutions. The second step is to define a criterion to optimize the problem with a genetic algorithm from the initial population created by ant colony. In section 2, the optimization strategy is depicted and then is adapted to the Unit Commitment problem in section 3. Numerical results are given in section 4. Forthcoming works are given in section 5 and concluding remarks are drawn in section 6.

2 OPTIMIZATION STRATEGY

2.1 Optimization Method Synopsis

The synopsis of the algorithm is depicted on figure 1. The idea is to firstly sample the feasible space with an ant colony algorithm. This algorithm is constructive. Thus, solutions are explicitly built as feasible ones (see section 2.2). However, ant colony may fail to find a very good solution: because of its positive feedback structure, it may be trapped in a local minimum. Thus, a genetic algorithm will be used to intensively explore the search space (see section 2.4) from the population computed by ants. For that purpose, a new criterion is defined, for which feasibility is guaranteed (see section 2.3).

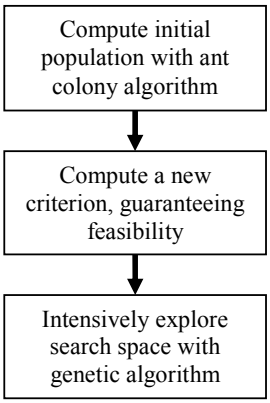


Figure 1: Optimization strategy synopsis.

The proposed methodology can be used to solve various and large scale optimization problems, the satisfaction of all constraints being guaranteed.

2.2 Ant Colony Optimization

Ant colony optimization was introduced by Marco Dorigo (Dorigo, *et al.*, 1996). It is based on the way ants are looking for food. The metaphor is used to solve graph exploration problems. Various criterions can be optimized. For instance, a cost may be associated to each node. The goal is to minimize the sum of costs while exploring the graph. This is the well known Travelling Salesman Problem, for which ant colony algorithm has been firstly used (Dorigo, et al., 1997). During iteration t of the algorithm, F ants walk on the graph of figure 2.

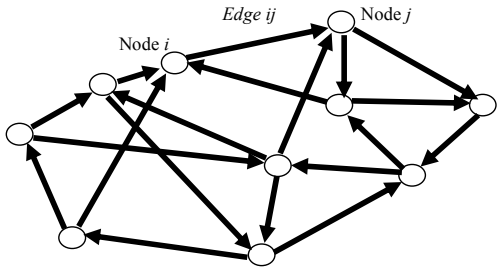


Figure 2: Graph exploration.

If ant f has reached node i, the probability that it chooses the next node j is:

$$p_i^{(f)}(j) = \frac{\eta_{ij}{}^{\alpha}\tau_{ij}(t)^{\beta}}{\sum_{m\in J_f(i)}\eta_{im}{}^{\alpha}\tau_{im}(t)^{\beta}}. \qquad (1)$$

- $\tau_{ij}(t)$ is the pheromone trail on edge ij during iteration t. Its value depends on the results of previous ants;
- η_{ij} is the attractiveness. It refers to the « local choice ». For the Traveling Salesman Problem, $\eta_{ij} = 1/d_{ij}$, where d_{ij} is the cost associated to the edge ij of the graph;
- α and β are weighting factors;
- $J_f(i)$ is the feasible set (for the travelling Salesman problem, this feasible set contains all nodes connected to node i, but those nodes which have been already explored by the ant are to be removed).

At the end of iteration t of the algorithm, F ants have computed F potential solutions, which are evaluated. The pheromone trail which laid on the graph of figure 2 is then updated:

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \Delta\tau_{ij}(t) \qquad (2)$$

$\Delta\tau_{ij}$ is the update coefficient whose value depends on the results of ants in iteration t. The more the ants which have walked on edge ij have obtained good results, the higher is $\Delta\tau_{ij}$. Several strategies can be

used for this update. ρ is the evaporation coefficient. This coefficient is an analogy with physical evaporation of pheromone in nature. Usually, there are about 20 to 30 ants, and $\alpha \approx 1$ and $\beta \approx 2$. It is often necessary to bound the pheromone trail on each edge ($\sigma \in [\sigma_{\min}, \sigma_{\max}]$), to avoid premature convergence; see (Stützle, *et al.*, 2000). Note that ant colony optimisation is a constructive algorithm. As results, it can explicitly take into account all the constraints of the problem, with the help of the feasible sets $J_f(i)$ which are built for each ant.

2.3 Defining a Feasibility Criterion

Consider the following optimization problem:

$$\min_x f(x) \quad s.t \begin{cases} h(x) = 0 \\ g(x) \le 0 \end{cases} \tag{3}$$

If a feasible solution with cost c^f is known, the following feasibility criterion can be defined:

$$\min_x f(x) + ((1+\varepsilon)\, c^f + h(x)).B(x) \tag{4}$$

where

- ε is a small positive real;
- $h(x)$ is a penalty function.
- $B(x)$ is a boolean function with value 1 for non feasible solutions and 0 for feasible ones.

Any unfeasible solution has a higher cost than the feasible known solution. Then the criterion can be optimized by any unconstrained optimization algorithm, and for instance by genetic algorithm. In the proposed strategy, c^f is the cost of the best solution found by the ant algorithm.

2.4 Genetic Algorithm

Genetic algorithm is a classical global optimization method. The general flow chart of this algorithm is given in figure 3. Classical cross-over and mutation operators are represented in figure 4 and 5 for a binary optimization problem. The aim of the crossover operator is to create 2 new potentially efficient individuals from 2 parents by mixing their variables.

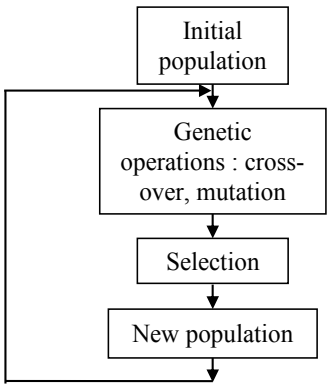


Figure 3: General flow chart of a genetic algorithm.

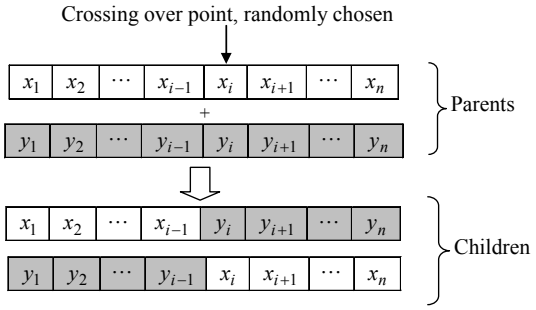


Figure 4: Crossing-over operator.

The mutation operator allows the introduction of new genes in the population by randomly changing one of the variables. The selection operator is made via the classical biased roulette selection.

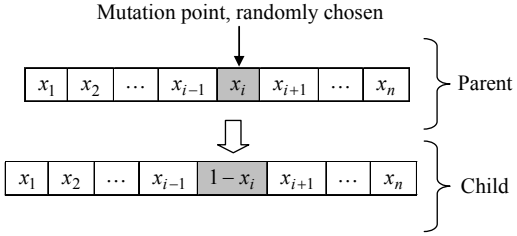


Figure 5: Mutation operator.

3 APPLICATION TO THE UNIT COMMITMENT

3.1 Unit Commitment Problem

The Unit Commitment refers to the minimization of global costs of *K* production units, over a time horizon *N*, satisfying a consumer's demand:

$$\min_{\{u_n^k, Q_n^k\}} \sum_{n=1}^{N} \left(\sum_{k=1}^{K} \begin{pmatrix} c_{prod}^k(Q_n^k, u_n^k) \\ + c_s^k(u_n^k, u_{n-1}^k) \end{pmatrix} \right), \quad (5)$$

where:

- Q_n^k is the produced power of unit k at time n;
- u_n^k is the on/off status of unit k at time n (binary variable);
- $c_{prod}^k(Q_n^k, u_n^k) = (\alpha_1^k Q_n^k + \alpha_0^k) u_n^k$ is the cost function of unit k (α_i^k are technical data).
- $c_s^k(u_n^k, u_{n-1}^k) = c_{on}^k u_n^k (1 - u_{n-1}^k) + c_{off}^k u_{n-1}^k (1 - u_n^k)$ is the start up and shut down cost of unit k.

Constraints of the problem are:

- Capacity constraints:

$$\underline{Q}^k u_n^k \le Q_n^k \le \overline{Q}^k u_n^k \quad (6)$$

- Satisfaction of consumer's demand Q_n^{dem}:

$$\sum_{k=1}^{K} Q_n^k \ge Q_n^{dem} \quad (7)$$

- Time-up T_{up}^k and time-down T_d^k constraints:

$$(u_{n-1}^k = 0, u_n^k = 1) \Rightarrow \left(u_{n+1}^k = 1, u_{n+2}^k = 1, \cdots, u_{n+T_{up}^k - 1}^k = 1 \right) \quad (8a)$$

$$(u_{n-1}^k = 1, u_n^k = 0) \Rightarrow \left(u_{n+1}^k = 0, u_{n+2}^k = 0, \cdots, u_{n+T_d^k - 1}^k = 0 \right) \quad (8b)$$

3.2 Graph Exploration Formulation for Ant Colony

The Unit Commitment problem can be depicted by the graph represented in figure 6. Nodes of the graph are all the possible states of production system: $\{u_n^1, \ldots, u_n^K\}$. The goal is to go from one of the possible states at time 1, to one of the possible states at time N, while satisfying all constraints and minimising global costs. Start up/shut down are associated to edges of the graph; production costs are associated to nodes.

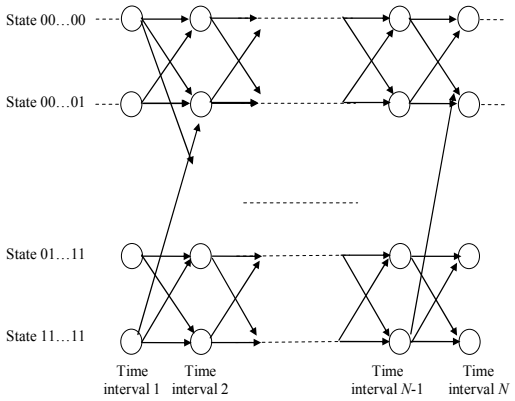


Figure 6: Graph exploration formulation of Unit Commitment.

3.3 Computation of Real Variables

Ant colony and genetic algorithm are mostly dedicated to integer programming. The problem can be reformulated in a purely integer programming problem. Consider that binary variables are given and refer to a feasible solution. Real variables are computed as the solution of the following optimization problem:

$$\begin{aligned} &\arg\min_{\{Q_n^k\}} \sum_{n=1}^{N} \left(\sum_{k=1}^{K} \begin{pmatrix} (c_{prod}^k(Q_n^k, u_n^k) \\ + c_s^k(u_n^k, u_{n-1}^k)) \end{pmatrix} \right) \\ &= \arg\min_{\{Q_n^k\}} \left(\sum_{n=1}^{N} \sum_{k=1}^{K} \alpha_1^k Q_n^k u_n^k \right) \end{aligned} \quad (9)$$

The optimal solution is to produce as much as possible with low-cost units, while satisfying capacity constraints:

$$\begin{aligned} Q_n^1 &= \min\left(\max\left(Q_n^{dem} - \sum_{i=2}^{K} \underline{Q}^i u_n^i, \underline{Q}^1 \right), \overline{Q}^1 \right) u_n^1 \\ &\vdots \\ Q_n^k &= \min\left(\max\begin{pmatrix} Q_n^{dem} - \sum_{i=1}^{k-1} Q_n^i \\ - \sum_{i=k+1}^{K} \underline{Q}^i u_n^i, \underline{Q}^k \end{pmatrix}, \overline{Q}^k \right) u_n^k \\ &\vdots \\ Q_n^K &= \min\left(\max\left(Q_n^{dem} - \sum_{i=1}^{K-1} Q_n^i, \underline{Q}^K \right), \overline{Q}^K \right) u_n^K \end{aligned} \quad (10)$$

3.4 Enhanced Genetic Algorithm

It has been observed that the classical genetic algorithm can be more efficient by using the a priori knowledge of the system (Sandou, *et al.*, 2007). A "selective mutation operator" is added to the classical genetic operators. Consider figure 7, with a particular unit scheduling. Very often, a random mutation leads to an infeasible solution, because of time-up and time-down constraints. To increase the probability of reaching a feasible point with a mutation, the muted gene has to be located at switching times of the planning.

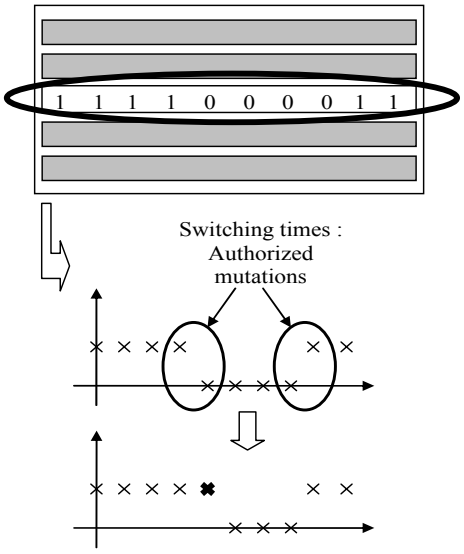


Figure 7: Selective mutation operator.

4 NUMERICAL RESULTS

The proposed strategy has been tested with Matlab 6.5 with a Pentium IV 2.5GHz. Optimization horizon is 24 hours with a sampling time of one hour. A comparison is made with pure genetic algorithm. A "four unit" case is considered. Characteristics are given in table 1. c_{on} is 2€ and c_{off} is 10€ for all units.

Table 1: Characteristics of the "4 unit case".

k	$Q_{\min}$ (MW)	$Q_{\max}$ (MW)	α_0 (€)	α_1	T_d (h)	T_{up} (h)
1	10	40	25	2.6	2	4
2	10	40	25	7.9	2	4
3	10	40	25	13.1	3	3
4	10	40	25	18.3	3	3

For this small case example, a "Branch and Bound" method has been developed so as to gat the global optimum and validate the method. Parameters of the the ant colony algorithm are: $\alpha = 1; \beta = 2,\ \rho = 0.2$, $\tau_{\min} = 1; \tau_{\max} = 5$. For the genetic algorithm, parameters are: Crossover probability 70%, classical mutation probability 5%, selective mutation: 10%. Results are given in table 2. As stochastic algorithms are considered, 100 tests are performed for each case, and statistical data about the results are given: mean cost (compared to the global optimum), worst case, rate of success (number of times that the global optimum is found) and computation times. The population are set to 50 individuals for genetic algorithm. When it is fed by ant colony, 5 iterations of 10 ants are performed to compute 50 initial solutions.

Table 2: Results for the "4 unit" case.

	Mean	**Worst**	**Success**	**Comp. times**
Ants + GA 100 iter	+2.5%	+9.4%	30%	15 s
GA 100 iter	+3.1%	+13.4%	20%	13 s
Ants + GA 200 iter	+0.4%	+3.5%	80%	25 s
GA 200 iter	+0.5%	+4.5%	77%	24 s

Computation times are very low for all cases. Results show that the use of ant colony as a "feasible solutions generator" leads to an increase in the quality of the solution. In particular, the worst case cost is better with the cooperative method. Thus, the cooperative method is very satisfying, especially for the "50 Ants – GA 200 iterations" case, as the mean result is just 0.4% higher than the optimal solution, and the worst case leads to a slight increase (less than 4%). Computation times are about 25 seconds.

For 200 generations, results seem to prove that the ant generation has no influence anymore, compared with the pure genetic algorithm. However, for successful tests, it is interesting to have a look on the iteration number for which the best solution has been found. For pure generic algorithm, the best solution is found after 126 generations (average number), whereas it is found after 96 generations for the ant colony/genetic algorithm. Convergence is achieved earlier with the cooperative method. Note that results of the pure genetic algorithm are still very satisfying, thanks to the selective mutation operator, as shown in (Sandou, et al., 2007).

5 DISCUSSION

5.1 Very Large Scale Cases

The interest of a feasible initial population has been shown in previous results. In this paper, this feasible population is computed by an ant algorithm. The ant colony algorithm can be seen as a stochastic dynamic programming algorithm. The size of the state space is 2^K. This is the main limiting point of the proposed method. Thus, for high values of K, the computation times of the ant colony algorithm grows very quickly. Thus, one of the main points is the application of ant colony to very large scale cases.

5.2 Global Optimization and Cooperation

Ant colony algorithm and genetic algorithm are two global optimization techniques and it may be astonishing to use them as a cooperative method. The goal of this hybridising was to couple the feasibility properties of ant colony algorithm and the intensive exploration of genetic algorithm. The cooperation is a sequential procedure, and a more alternated procedure could be profitable. For instance, results of genetic algorithms can be used to define the attractiveness parameters in a new iteration of ant colony algorithms. Furthermore, it will be interesting to couple the method with a local search.

6 CONCLUSIONS

In this paper, an optimization strategy has been defined and applied to solve the Unit Commitment. The main idea is to use an ant algorithm as a feasible solutions generator. These feasible solutions are brought together in an initial population for a genetic algorithm. To guarantee the feasibility of the final solution, a special criterion is computed from the results of the ant algorithm. To increase the efficiency of the classical genetic algorithm, a knowledge-based operator is defined (selective mutation). Finally, the proposed method leads to high quality solution, with guarantees of feasibility and with low computation times. The main limiting point appears to be the computation times of ant colony algorithm for very large scale cases. However, the use of feasible solutions in the initial population of a genetic algorithm is an interesting way to decrease the number of iterations required to find near optimal solutions. Forthcoming works deal with the use of such algorithm for predictive control of non linear hybrid systems.

REFERENCES

Chen C.-L, Wang S.-C., 1993. Branch and Bound scheduling for thermal generating units. In: *IEEE Transactions on Energy Conversion*, 8(2), 184-189.

Dorigo M., Maniezzo V., Colorni A., 1996. The Ant System: Optimization by a Colony of Cooperating Agents. In: *IEEE Transactions on Systems, Man and Cybernetics-Part B*, 26(1), 1-13.

Dorigo M., Gambardella, L. M., 1997. Ant Colony System : a Cooperative Learning Approach to the Traveling Salesman Problem. In: *IEEE Transactions on Evolutionary Computation*, 1, 53-66.

Ouyang Z., Shahidehpour S. M., 1991. An intelligent dynamic programming for unit commitment application. In: *IEEE Transactions on Power Systems*, 6(3), 1203-1209.

Rajan C. C. A, Mohan M. R., 2004. An evolutionary programming-based tabu search method for solving the unit commitment problem. In: *IEEE Transactions on Power Systems*, 19(1), 577-585.

Sandou, G., Font, S., Tebbani, S., Hiret, A., Mondon, C., 2007. Enhanced genetic algorithm with guarantee of feasibility for the Unit Commitment problem. In: *Proceeding of the 8th International Conference on Artificial Evolution*, Tours, France.

Sen S., Kothari D. P., 1998. Optimal Thermal Generating Unit Commitment: a Review. In: *Electrical Power & Energy Systems*, 20(7), 443-451.

Senjyu T., Shimabukuro, K., Uezato K. and Funabashi T., 2004. A fast technique for Unit Commitment problem by extended priority list. In: *IEEE Transactions on Power Systems*, 19(4), 2119-2120.

Stützle T., Hoos, H. H., 2000. MAX-MIN Ant System, In: *Future Generation Computer Systems*, 16, 889-914.

Swarup K., Yamashiro, S., 2002. Unit commitment solution methodology using genetic algorithm. In: *IEEE Transactions on Power Systems*, 17(1), 87-91.

Yin Wa Wong S., 1998. An Enhanced Simulated Annealing Approach to Unit Commitment. In: *Electrical Power & Energy Systems*, 20(5), 359-368.

Zhai Q; Guan X., 2002. Unit Commitment with identical units: successive subproblems solving method based on Lagrangian relaxation. In: *IEEE Transactions on Power Systems*, 17(4), 1250-1257.

SMART SEMANTIC MIDDLEWARE FOR THE INTERNET OF THINGS

Artem Katasonov, Olena Kaykova, Oleksiy Khriyenko, Sergiy Nikitin and Vagan Terziyan
Agora Center, University of Jyväskylä, P.O.Box 35, 40014, Finland
akataso@jyu.fi, helen@jyu.fi, olkhriye@jyu.fi, senikiti@jyu.fi, vagan@jyu.fi

Keywords: Internet of Things, Middleware, Interoperability, Heterogeneous Resources, Software Agents, Semantic Technologies, Ontologies.

Abstract: As ubiquitous systems become increasingly complex, traditional solutions to manage and control them reach their limits and pose a need for self-manageability. Also, heterogeneity of the ubiquitous components, standards, data formats, etc, creates significant obstacles for interoperability in such complex systems. The promising technologies to tackle these problems are the Semantic technologies, for interoperability, and the Agent technologies for management of complex systems. This paper describes our vision of a middleware for the Internet of Things, which will allow creation of self-managed complex systems, in particular industrial ones, consisting of distributed and heterogeneous components of different nature. We also present an analysis of issues to be resolved to realize such a middleware.

1 INTRODUCTION

Recent advances in networking, sensor and RFID technologies allow connecting various physical world objects to the IT infrastructure, which could, ultimately, enable realization of the Internet of Things and the Ubiquitous Computing visions. Also, this opens new horizons for industrial automation, i.e. automated monitoring, control, maintenance planning, etc, of industrial resources and processes. A much larger than in present number of resources (machines, infrastructure elements, materials, products) can get connected to the IT systems, thus be monitored and potentially controlled. Such development will also necessarily create demand for a much wider integration with various external resources, such as data storages, information services, and algorithms, which can be found in other units of the same organization, in other organizations, or on the Internet.

The interconnectivity of computing and physical systems could, however, become ”the nightmare of ubiquitous computing” (Kephart and Chess, 2003) in which human operators will be unable to manage the complexity of interactions in the system, neither even architects will be able to anticipate that complexity, and thus to design the system. The IBM vision of autonomic computing (Kephart and Chess, 2003) proclaims the need for computing systems capable of ”running themselves” with minimal human management which is mainly limited to definition of some higher-level policies rather than direct administration. The computing systems will therefore be self-managed, which, according to the IBM vision, includes self-configuration, self-optimization, self-protection, and self-healing. The IBM vision emphasizes that the run-time self-manageability of a complex system requires its components to be to a certain degree autonomous themselves. Following this, we envision that the *software agent technologies* will play an important part in building such complex systems. Agent-based approach to software engineering is considered to be facilitating the design of complex systems (Jennings, 2001). A significant attention is paid in the field of multi-agent systems to the task of building decentralized systems capable of supporting spontaneous configuration, tolerating partial failures, or arranging adaptive reorganization of the whole system (Mamei and Zambonelli, 2006).

A major problem is inherent heterogeneity in ubiquitous computing systems, with respect to the na-

ture of components, standards, data formats, protocols, etc, which creates significant obstacles for interoperability among the components. *Semantic technologies* are viewed today as a key technology to resolve the problems of interoperability and integration within heterogeneous world of ubiquitously interconnected objects and systems. Semantic technologies are claimed to be a qualitatively stronger approach to interoperability than contemporary standards-based approaches (Lassila, 2005). The Internet of Things should become in fact the *Semantic Web of Things* (Brock and Schuster, 2006). We subscribe to this view. Moreover, we apply semantic technologies not only to facilitate the discovery of heterogeneous components and data integration, but also for the behavioral control and coordination of those components (i.e. prescriptive specification of the expected behaviour, declarative semantic programming).

It seems to be generally recognized that achieving the interoperability by imposing some rigid standards and making everyone comply could not be a case in ubiquitous environments. Therefore, the interoperability requires existence of some middleware to act as the glue joining heterogeneous components together. A consistent set of middleware, offering application programming interfaces, communications and other services to applications, will simplify the creation of applications and help to move from static programming approaches towards a configurable and dynamic composition capability (Buckley, 2006).

There are a couple of EU FP6 research projects that have as one of their goals the development of some middleware for embedded systems. They are RUNES (Reconfigurable Ubiquitous Networked Embedded Systems, 2004-2007) and ongoing SOCRADES (Service-Oriented Cross-Layer Infrastructure for Distributed Smart Embedded Devices, 2006-2009). We believe, however, that the middleware needs of the Internet of Things domain go well beyond interconnectivity of embedded systems themselves. There is a more general need for middleware to enable something we refer to as Global Enterprise Resource Integration (GERI), where all different types of resources get seamlessly integrated: physical devices with embedded electronics, web services, software applications, humans along with their interfaces, and other. In the concept of GERI, we also stress the need for *true global interoperability, not just interconnectivity*. The components of ubiquitous computing systems should be able not only to communicate and exchange data, but also to flexibly coordinate with each other, discover and use each other, and jointly engage in different business processes.

Such more general middleware needs are emphasized in the Strategic Research Agenda (SRA) of the ARTEMIS European Technology Platform. ARTEMIS' SRA has "Seamless Connectivity and Middleware" as one of its three parts. Some of the relevant research priorities listed are the middleware as the key enabler for *declarative programming* paradigm, where the components and their interaction are defined and configured declaratively rather than programmatically (and we believe that the semantic technologies are a natural candidate here), efficiently bridging information between *global, enterprise, and embedded systems*, use of *ontologies* for cross-domain systems' organization and for interoperability in heterogeneous environments, dynamic reconfiguration capabilities, adaptive resource management, and appropriate security infrastructures.

In this paper, we describe our vision of such a middleware for the Internet of Things, which has also formed the basis for our research project UBIWARE. The project aims at a new generation middleware platform which will allow creation of self-managed complex systems, in particular industrial ones, consisting of distributed, heterogeneous, shared and reusable components of *different* nature, e.g. smart machines and devices, sensors, RFIDs, web-services, software applications, humans along with their interfaces, and others. Such middleware will enable various components to automatically discover each other and to configure a system with complex functionality based on the atomic functionalities of the components.

The rest of the paper is structured as follows. Section 2 describes our general vision. Section 3 presents an analysis of how the goals of UBIWARE can be achieved. The result is a set of sub-problems, which we address in corresponding work-packages of the project. Section 4 describes several industrial cases (application areas) that we consider in the project; those are proposed by the project's industrial partners. Finally, Section 5 concludes the paper.

2 THE GENERAL VISION

This section describes our general vision of the smart semantic middleware for the Internet of Things. We believe that tasks of automatic integration, orchestration and composition of complex systems on the Internet of Things will be impossible in a centralized manner due to the scalability and other issues. Therefore, the components of such systems should be to a certain degree autonomous and proactive. In other words, utilization of the agent technologies is needed to enable flexible communication and coordination of the components. Agent technologies also allow mobility of service components between various platforms, decentralized service discovery, utilization

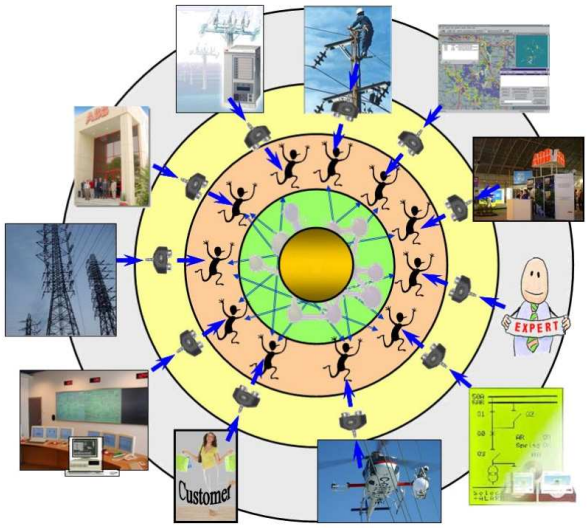


Figure 1: The conceptual architecture: resources, adapters, agents and ontology.

of FIPA communication protocols, and negotiation-based integration/composition of services. Interoperability among the components requires use of metadata and ontologies. As the amount of components can grow dramatically, without their ontological classification and semantic annotation processes, the automatic discovery will be impossible.

Figure 1 depicts our vision. In this vision, each resource to be connected has a representative – an autonomous software agent (a proactive ”player” within certain integration scenario). This agent is assumed to be able to monitor the state of the resource, make decisions on the behalf on the resource, and to discover, request and utilize external help if needed. The connection between the resource itself and its agent is organized through the semantic mediation of an *adapter* (or interface). Such an adapter may include (if necessary) sensors with digital output, data structuring (e.g. XML), semantic adapter components (converting to a semantic representation), and actuators. According to this vision, each relevant component of a ubiquitous system is to become a *”smart resource”* - proactive and self-managing. The part-of hierarchy of resources may result in corresponding hierarchy of the agents. This can also be recursive. For example, an adapter of a system component can become a smart resource itself, i.e. it can have its own responsible agent, semantically adapted sensors and actuators, history, commitments with other resources, and self-monitoring and self-maintenance activities. This could be done to guarantee high level of dynamism and flexibility of the adapter.

The semantic technologies have in our vision a two-fold value. First, they are the basis for the discovery of heterogeneous resources and data integration across multiple domains (a well-known advantage). Second, they are used for behavioural control and coordination of the agents representing those resources (a novel use). Therefore, semantic technologies are used both for *descriptive* specification of the services delivered by the resources and for *prescriptive* specification of the expected behaviour of the resources as well as the integrated system. Such two-fold application of semantics can, in ideal case, lead to a ”global understanding” between the resources. This means that a resource A can understand all of (1) the properties and the state of a resource B, (2) the potential and actual behaviors of B, and (3) the business processes in which A and B, and maybe other resources, are jointly involved.

The UBIWARE is to be, roughly speaking, a set of tools providing means for effective development of agents and adapters, and a run-time environment for the operation of those. An important central component of UBIWARE is the *agent core* (see Section 3.1). Each agent is assumed to be created based on this core and specialized through semantic declarative programs and a set of Reusable Atomic Behaviors (RAB). Combination of the run-time environment, the agent core and a set of standard semantic programs and RABs will make UBIWARE a platform providing both (semantic) communication services and collaboration-support (scenario-driven integration) services for heterogeneous resources.

In a sense, our intention is to apply the concepts of automatic discovery, composition, orchestration, integration, execution monitoring, communication, negotiation, context awareness, etc. (which were, so far, mostly related only to the Semantic Web-Services domain) to a more general ”Semantic Web

of Things" domain. Also we want to expand this list by adding automatic self-management including (self-*)organization, diagnostics, forecasting, configuration, tuning, and maintenance.

3 ANALYSIS

This section presents our analysis of how the goals of UBIWARE can be achieved. The result is a set of subproblems, which we address in corresponding workpackages of the project. UBIWARE project aims at a relatively complete and self-sufficient middleware platform. Therefore, it elaborates on our central ideas (Section 2), and also works towards solutions in supporting but mandatory-to-treat areas such as security, human interfaces and other.

3.1 Core Agent Platform

Although the flexibility of agent interactions has many advantages when it comes to engineering complex systems, the downside is that it leads to unpredictability in the run time system; as agents are autonomous, the patterns and the effects of their interactions are uncertain (Jennings, 2000). It is common in specific systems to circumvent these difficulties by using interaction protocols whose properties can be formally analyzed, by adopting rigid and preset organizational structures, and/or by limiting the nature and the scope of the agent interplay. However, these restrictions also limit the power of the agent-based approach; thus, in order to realize its full potential some longer term solutions are required (Jennings, 2000).

Realization of the UBIWARE vision requires a reliable core platform that would provide means for building systems that are *flexible* and consist of autonomous components, yet *predictable* in operation. Two important research directions, acknowledged in the literature, are: social level characterization of agent-based systems, and ontological approaches to coordination. The former direction presents the need for a better understanding of the impact of sociality and organizational context on an individuals behavior and of the symbiotic link between the behavior of the individual agents and that of the overall system (Jennings, 2000). In particular, it requires modeling behavior of an agent as being defined or restricted by the roles the agent plays in one or several organizations (Vazquez-Salceda et al., 2005). Role-based modelling also provides the advantage of separation of concerns and in so design of complex systems (Cabri et al., 2004). The latter direction presents the need to enable agents to communicate their intentions with respect to future activities and resource utilization and to reason about the actions, plans, and knowledge of each other, in real time (Tamma et al., 2005).

Our previous work resulted in a platform (Katasonov and Terziyan, 2007) (Figure 2) that has done some steps into both these directions. It can be seen as consisting of three layers: reusable atomic behaviors (RAB), behavior models corresponding to different roles the agent plays, and the behavior engine (or the agent core). A RAB is a Java component implementing a reasonably atomic function (sensing, acting or data processing). A behavior model is an RDF-based document specifying a certain organizational role. A behavior model consists of a set of beliefs representing the knowledge needed for playing the role and a set of behavior rules. Roughly speaking, a behavior rule specifies conditions of (and parameters for) execution of a RAB. The behavior engine is responsible for parsing RDF-based scripts, and it implements the run-time loop of an agent. In the platform, the agents access the behavior models from an external repository, which is assumed to be managed by the organization which "hires" the agents to enact those roles. As can be seen from the picture, the platform allows also on-demand access of RABs.

Such a 3-layer agent architecture with externalization of behavior models and on-demand access of atomic code components provides a basis for development of the UBIWARE core platform. At present, we are working on the following important research issues: (1) Development of the language for roles' scripts towards a full-scale Semantic Agent Programming Language. (2) Implementation of the separation between a role's capabilities (individual functionality), and the business processes in which this role can be involved (complex functionality). (3) Development of mechanisms for flexible treating the potential (and likely) conflicts among the roles played by one agent. (4) Development of mechanisms to enable agents to flexibly discover each other, based both on the roles played and on particular capabilities possessed. (5) Analysis of concrete benefits and mechanisms for accessing and using a role's script by agents who are not playing that role but wish to coordinate or interact with an agent that does. (6) Making an agent's roles to be higher-level commitments of the agent that restrict its behavior, still leaving freedom for learning and adaptation on lower-levels, instead of totally and rigidly prescribing the behavior.

3.2 Distributed Resource Histories

In the UBIWARE vision, every resource is represented by an agent. Among the responsibilities of the agent is monitoring the condition of the resource and

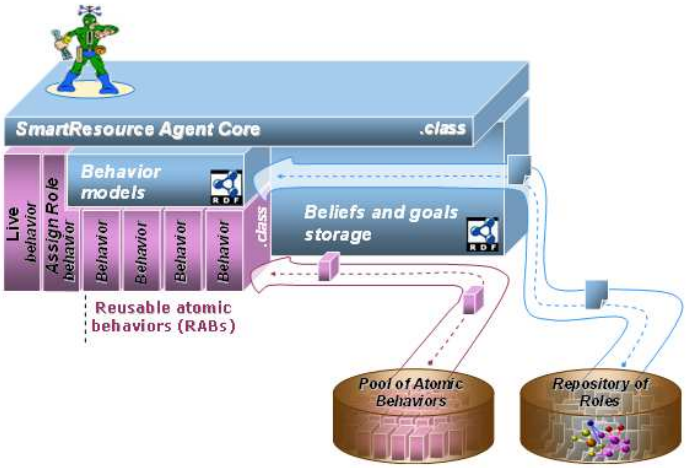


Figure 2: The current core platform.

the resource's interactions with other components of the system and humans. The beliefs storage of the agent will, therefore, naturally include the history of the resource, in a sense "blogged" by the agent. Obviously, the value of such a history is not limited to that particular resource. A resource may benefit from the information collected with respect to other resources of the same (or similar) type, e.g. in a situation which it faces for the first time while other may have faced that situation before. Also, mining the data collected and integrated from many resources may result in discovery of some knowledge important at the level of the whole ubiquitous computing system.

A straightforward approach would be maintaining a central repository integrating the histories of all the resources of the system. However, such an approach is often impractical in realistic dynamic applications because even just keeping one agent informed of all the events and actions in the system would swamp the available bandwidth, and also such agent would become a severe bottleneck and might render the remaining components unusable if it failed (Jennings, 1993). A scalable solution requires mechanisms for inter-agent information sharing and data mining on integrated information which would allow keeping the resource histories distributed without need to copy those histories to a central repository.

At present, we are working on the following important research issues: (1) Development of mechanisms for representing the history of a resource in a system making it reusable for other resources and at the system-level. (2) Development of mechanisms for effective and efficient sharing of information between the agents representing different resources. (3) Development of mechanisms for querying and integration of responses from distributed, autonomous, and, hence, inevitably semantically heterogeneous resource histories. (4) Enabling mining (utilizing intelligent data mining and machine learning techniques) of distributed histories.

3.3 Peer-to-Peer Discovery

Following the IEEE FIPA agent system model, the UBIWARE platform is to include a system agent called Directory Facilitator (DF). In UBIWARE, the Directory Facilitator maintains a mapping between agents and the roles they play. If the behavior model of an agent X (see Section 3.1) prescribes the need of interaction with another agent Y, the agent Y is always specified by its role, not the name or another unique identifier of a particular agent. Therefore, the agent X must contact the Directory Facilitator in order to discover the unique ID (needed for communication) of the agent or agents playing the role needed. Also, because every agent plays at least one role, DF naturally has a list of all the agents on the platform. This can be used, e.g., when an agent needs some information and wants to broadcast the request for that information to all the agents on the platform (see Section 3.2).

Existence of such a Directory Facilitator is an effective and efficient solution. However, DF obviously presents a severe bottleneck in the system, and can render the whole system unusable if it failed. To improve the survivability of UBIWARE-based systems, there has to be a complementary mechanism which can be utilized in an exception situation where DF became for some reason unavailable.

After the system is deployed and operational for some time, a significant part of the DF knowledge, piece by piece, ends up in local knowledge storages of different platform agents. The combination of the local storages presents thus a kind of distributed directory. Therefore, a Peer-to-Peer (P2P) mechanism im-

plemented on such a distributed directory can work as the needed mechanism complementary to the central DF. Of course, replicating DF is another and probably simpler option. It would not, however, provide the same level of survivability as P2P. Also, in some business scenarios, it is possible that some of the services would prefer not to advertise themselves through the central DF altogether, for security or other reasons. Therefore, P2P discovery of such services would not be an exception path but the only viable solution.

The objective here is the design of mechanisms which will extend the scale of semantic resource discovery in UBIWARE with P2P discovery. Such mechanisms have to enable an agent (1) to discover agents playing a certain organizational role, (2) to discover an agent (or agents) possessing certain needed information, and (3) to discover resources (through its agents) of certain type or possessing certain properties (e.g. a device in state X, a web-resource providing some information searched for, or a human with some specific skills). In all cases, existence of a central DF is not assumed. Rather, the request is sent to all/some of the agents on the contact list of the agent in question (ones with who it has a history of communication, or at least about whom it heard from others). Those agents can forward the request to all/some of the agents on their lists, and so on.

3.4 Configurability

UBIWARE aims to be a platform that can be applied in different application areas. Therefore, the elements of the platform have to be adjustable, could be tuned or configured allowing the platform to run different business scenarios in different environments. Such flexibility calls for existence of a special configuration layer of the platform. All building blocks of the platform, i.e. software agents, agent behaviors, resource adapters, etc, become subject to configuration. Also, a flexible system should have a long lifespan. Hence, the platform should allow extensions, component replacements, and component adjustments during the operation time. In addition, every agent in UBIWARE is self-aware and self-manageable entity. Therefore, it may evolve with time and modify some of its functional or non-functional properties. At the same time, we need a stable and predictable environment for running business scenarios. Thus, the adjustment made to a component must also be propagated to higher platform levels such as business processes and contracting. Also vice versa, a change in a business process may require some adjustments to be made to several participating components.

On the level of software design, we aim at defining patterns for development of configurable elements. On the level of inter-agent communication, we aim at establishing some protocols for negotiation in different configurability cases. For example, when an agent changes its behavior, a check must be performed for the consequences of this change for other components, and for who should be informed about this change. Also, there should be a mechanism for restricting the range of possible adjustments to certain functionality of certain components in order to maintain the consistency in critical business processes.

An important issue is configurability of resource adapters. There is a need for a methodology both for creating such adapters and for their run-time reconfiguration. It is reasonable to expect, e.g., that when a resource modifies the format of its output, the existing adapter can be adjusted to this new format without a need to build a new adapter from scratch.

Configuration of business processes poses a separate problem. A business process is an abstract entity which mainly consists of a flow of messages between agents. There is nothing like a business process execution engine. There could be a dedicated responsible agent, which would coordinate the involved agents, resolve conflicts and exceptions. However, similarly to the case of a centralized history storage (see Section 3.2), such a controller agent would become a severe bottleneck and might render the remaining components unusable if it failed. In a most scalable solution, a business process would be just a set of agents with some commitments, i.e. signed Service Level Agreements. In such autonomic medium, we will need a (re-)configuration mechanism for stable error-tolerant operation of components leading to successfully achieving the business goals.

3.5 Security in UBIWARE

The security is often seen as an add-on feature of a system. However, in many systems (and UBIWARE is one of them), the system remains nothing more but a research prototype, without a real potential of practical use, until an adequate security infrastructure is embedded into it. UBIWARE will advance existing technologies to a qualitatively new level and bring to life new complex ubiquitous environments, where traditional approaches to manage security fall short. Also, existing security measures for the technologies on which UBIWARE relies, e.g. multi-agent, are not in a mature stage and still require significant elaboration to mitigate associated risks.

Traditional security goals like confidentiality, availability, reliability, integrity, manageability, accountability, responsibility etc, together with conventional measures and mechanisms that support security, do not cover all the needs and threats of new

emerging computing environments. Ambient intelligence and ubiquity of information technologies have tightened the digital and physical worlds to the extent when security becomes the ultimate issue. The major implication of penetrating ICTs on security is that the risks and negative consequences of security threats become higher than ever. On the other hand, the security infrastructure itself has to become pervasive, interoperable and intelligent enough to naturally fit UBIWARE. The security cannot be added to the UBIWARE platform later but the design decisions regarding security have to be thoroughly correlated with the requirements, characteristics and design of the platform, due to mutual impact on resulting features of UBIWARE.

The main objective here is the design of an infrastructure for policy-based optimal collecting, composing, configuring and provisioning of security measures in multi-agent systems like UBIWARE (Naumenko et al., 2007). This work is to follow the general UBIWARE vision – configuring and adding new functionality on-the-fly by changing high level declarative descriptions. Regarding security, this means that UBIWARE will be able of smoothly including new, and reconfiguring existing, security policies (similar to behavior models) and mechanisms (similar to RABs) in response to the dynamically changing environment. The optimal state is always a tradeoff between security and other qualities like performance, functionality, usability, applicability and other.

3.6 Smart Interfaces

In UBIWARE, humans are important resources, which can play several distinct roles: (1) User – one getting some information or services from other resources. (2) Resource under care – one under online care of the integrated system (e.g. monitoring the health of an employee). (3) Service provider – one providing services to other resources (e.g. a maintenance expert). (4) UBIWARE administrator – one monitoring and configuring the integrated system.

Obviously, humans need graphical interfaces to interact with the rest of the system. The same person can play several roles, switch between them depending on the context, and, in result, require different interfaces at different times. In addition, a UBIWARE-based system presents a large integration environment with potentially huge amounts of heterogeneous data. Therefore, there is a need for tools facilitating information access and manipulation by humans.

From the UBIWARE point of view, a human interface is just a special case of a resource adapter. We believe, however, that it is unreasonable to embed all the data acquisition, filtering and visualization logic into such an adapter. Instead, external services and application should be effectively utilized. Therefore, the intelligence of a smart interface will be a result of collaboration of multiple agents: the human's agent, the agents representing resources of interest (those to be monitored or/and controlled), and the agents of various visualization services. This approach makes human interfaces different from other resource adapters and indicates a need for devoted research. There is a need to enable creation of such smart human interfaces through flexible collaboration of an Intelligent GUI Shell, various visualization modules, which we refer to as MetaProvider-services, and the resources of interest (Khriyenko, 2007).

A MetaProvider is responsible for acting as a portal so that various relevant resources can register on it, for data integration, for context-dependent filtering (inclusion of only relevant resources and relevant properties of those resources), and for creating visualizations of data. The GUI shell is, in turn, responsible for context-dependent selection of MetaProviders, communication with them, and cross-MetaProvider browsing and integration.

Based on such an approach, an infrastructure will be embedded into UBIWARE enabling effective realization of the following system functions: (1) Visualization of data provided by a service in response to a request. (2) Search, retrieving and visualization of data required by a human expert. (3) Providing access to contextual information, and visualization of it. (4) Visualization of resource registration, configuration, and security policy establishment processes. (5) Resource discovery via MetaProviders (because they act as thematic portals).

4 INDUSTRIAL CASES

This section describes several industrial cases (application areas) that we consider in our project. These cases are proposed by UBIWARE industrial partners and are being analyzed, designed and prototyped based on the UBIWARE platform. These cases provide examples of benefits that a smart semantic middleware can bring into various industries.

4.1 Power Network Maintenance

One of the project's partners is a vendor of hardware and software for power networks. With respect to UBIWARE, this partner's main area of interest is in extending, mainly through integration with external resources, the functionality of its existing software systems. These systems provide a graphical view over

the power network, data acquisition from the substations and remote control over the relays, switches, etc. They also include implementations of various algorithms: for fault localization, for calculation of optimal reconfiguration of the network and other.

The UBIWARE technology could allow connecting to the existing products some new system intelligence tools, for example, statistical and data mining tools. One case that is considered is about possibility of using data mining techniques for automated interpretation of the situation in the power network. The motivating issue is that a certain condition, e.g. a fault, in the power network causes a series of alert messages to be sent to the operation center. It remains a responsibility of a human operator to understand the reason underlying such a sequence of alert messages. Based on UBIWARE, a system could be created allowing that: (1) alert data is automatically collected, (2) the experts are able to annotate sequences of events with their reasons, (3) data mining algorithms are applied to discover some patterns, and (4) the operator is provided with an interface giving a more structured view of events (e.g. filtered based on the source of event) and including the system's interpretation (from the models built) of what is the meaning of the situation and its reason.

In collaboration with this partner, we analysed further the potential add-value that the company and their customers, i.e. electrical companies, could receive from introducing UBIWARE into their business. Below, we sketch several scenarios when UBIWARE can enable new features, or help otherwise.

One scenario is related to extending the user interfaces, so that various groups of users could get flexible access to data and functionality present in the existing products. Traditionally, those software products are only used inside the walls of operation centers by the network operators, through proprietary user interfaces. The data and functionality of those systems, however, has a value beyond that use. A UBIWARE-based solution will enable a more ubiquitous and flexible information access to data and algorithms and will extend the user base to, e.g., maintenance workers, management, etc.

Another scenario arises from the fact that the medium-voltage sub-networks of the integral network are usually owned, controlled and maintained then by some local companies. It is noticeable that the operation centers of different companies have no connection to each other, so information exchange among them is nearly impossible. In the case of a fault affecting two different sub-networks, such information exchange, though, may be very important, for all of fault localization, network reconfiguration, and network restoration. Introducing an inter-organizational system based on UBIWARE could solve this issue. The information flow will go through the agents representing the sub-networks on the UBIWARE platform. Utilization of semantic technologies will allow such interoperability even if the sub-networks use software systems from different vendors, and thus maybe different data formats and protocols.

One more scenario is related to a new business model. At present, all expertise of the company gets embedded into hardware or software systems and sold to the customers as it is. A new business model would be to start own Web-service providing implementation of certain algorithms, so the customers will utilize those algorithms online when needed. The company will be always able to update algorithms, add new, and so on. UBIWARE will ensure interoperability and coordination between such Web-service and customers' software systems, and also a relative ease of implementation of such a solution – because it will not require changes in existing software systems, only extension with the UBIWARE platform. Noticeable that, if semantically defined, such Web-service can potentially be utilized across the globe even by the customers who never purchased any of the company's hardware or software.

The next scenario is related to the possibility of integrating data, which is currently utilized in the power network management (network structure and configuration, feeder relay readings), with contextual information from the external sources. Such integration can be used in at least three tasks. The first is risk analysis. Information about weather conditions, ongoing forest works, or forest fires can be used for evaluating existing threats for the power network. This may be used to trigger an alert state for the maintenance team, or even to do a precautionary reconfiguration of the network to minimize possible damage. The second is facilitation of fault localization. The output of fault localization algorithms is not always certain. The information about threats for the power network that existed at the time when the fault occurred (which thus may have caused the fault) may greatly facilitate the localization. In some situations, contextual information alone may even be sufficient for localization. The third is operator interface enhancement. Contextual information may be used also just to extend the operators' view of the power network. For example, satellite imagery can be used for geographic view (instead of locally stored bitmaps as it is in the current systems); also, dynamically-changing information can be accessed and represented on the interface.

The last scenario is about the possibility of transferring the knowledge of human experts to automated systems, by means of various data mining tools. In

the power network management case, one scenario that seems to be highly appropriate for such knowledge transfer is the following. In present, it is always a decision of a human expert which of the existing fault localization algorithms will perform the best in the context of the current configuration of the power network and the nature of the fault. Such decisions made by an expert along with the input data could, be forwarded to a learning Web-service. After a sufficient learning sample, this Web-service could start to be used in some situations instead of the human expert, e.g. in situations when a faster decision is needed or when the expert is unavailable.

4.2 Maintenance of Paper Machines

Another of the project's partners is a supplier of machinery and automation systems for a set of industries including the paper industry. With respect to UBIWARE, this partner's main areas of interest are in information integration and analytical data processing. To support customers with additional information along the product (a paper machine) lifecycle, the company foresees the need for intelligent product history management. Such a history is supposed to integrate alert reports, experts' diagnoses, maintenance work performed, maintenance costs, and other types of information.

Every paper machine, installed at a customer's site, is equipped with a set of sensors and embedded intelligence for observing the state of the machine and alarming when an exceptional situation occurs. The alarm information is forwarded to the central hub for diagnostics and decision making.

First of all, UBIWARE can provide means for proactive integration of alarm data with different corporate information systems, such as the engineering database, the financial database. This will give a common basis for product history management and further intelligent data processing. UBIWARE can also enable integration with external web services to provide more powerful analytical processing of data. Also, the flexibility and ease of extension via incorporation of external functionality will enable evolution of the service infrastructure to meet the customer needs in a sustainable fashion. The UBIWARE platform will also enable a high level of security in the collaborative work of the company and its customers.

Among the central problems to be solved by UBIWARE, there are those related to the evolution of the service infrastructure. During the paper machine operation lifecycle, formats of messages or algorithms for issuing alarms may change. Furthermore, some changes to the supporting ERP or other automation systems may take place. In other words, data formats and data structures, which were bound to ontology, may require re-adaptation (i.e. adapter reconfiguration) to become consistent with the integrated storage and applications. From a life-time perspective, the complexity of business processes between the company and the customer may become another obstacle for updates and renovation of the service infrastructure. It is because such changes are hard to trace and, therefore, it might be difficult to analyze their impact on the overall system.

4.3 Telecom Operator's Service Desk

One more project's partner is a provider of telecommunication services, both wireless and fixed-line. With respect to UBIWARE, this partner's main area of interest is in possibility of constructing, mainly through integration of existing components, systems that would (partially) automate some traditionally manual processes. Such a manual process to be considered is a Service Desk. Among other things, the company's service desk is a point of contact for the customers seeking to report unavailability of service (mobile or fixed-line phone connection, IDSL Internet connection, and other) or another problem. If the problem is at the customer side, the service desk experts are supposed to provide instructions for troubleshooting it. The whole interaction is performed over the phone, i.e. the customer calls, explains the problem to a service desk worker, probably gets transferred to a relevant expert, tries what the expert recommends, reports the results, and so on. In result, the waiting times for customers to get through to the service desk are long, while the effectiveness of the troubleshooting process is quite low.

Equipping the service desk with an UBIWARE-based solution could enable the following new functionality: (1) Customers get possibility to report their problems through a web-service interface. (2) The system could collect, integrate and present to the expert all the available information on the customer (type of connection, address, etc), removing a need to spend time asking that information during the phone conversation. (3) Automated analysis of related problem reports could enable fast responses like: "it is a network problem, we have received other similar reports from your area, we are already working on this problem". In the case of a non-working Internet connection, some relevant data can be acquired, i.e. sensed, from the network side. Also, some modems can be configured remotely by uploading a configuration file from the network. In such a case, a fully automatic troubleshooting could even be implemented.

5 CONCLUSIONS

In this paper, we described our vision of a solution to meet the middleware needs of the domain of the Internet of Things. We aim at a new generation middleware platform which will allow creation of self-managed complex systems, in particular industrial ones, consisting of distributed, heterogeneous, shared and reusable components of different nature.

Self-management of systems is one of the central themes in the EU 7-th Framework ICT Programme (2007-2013). The Objective "Service and Software Architectures" of the Challenge 1 "Network and Service Infrastructures" includes the need for strategies and technologies enabling mastery of complexity, dependability and behavioral stability, and also the need for integrated solutions supporting the networked enterprise. Also, the Objective "The Network of the Future" of this Challenge includes the need for re-configurability, self-organization and self-management for optimized control, management and flexibility of the future network infrastructure. In addition, the whole Challenge 2 "Cognition, Interaction, Robotics" has as its motivation the need for creating "artificial systems that can achieve general goals in a largely unsupervised way, and persevere under adverse or uncertain conditions; adapt, within reasonable constraints, to changing service and performance requirements, without the need for external re-programming, re-configuring, or re-adjusting". It is noticeable that the systems (stand-alone or networked) monitoring and controlling material or informational processes is one of the three focus areas of this Challenge.

According to a more global view to the Internet of Things technology, UBIWARE will classify and register various ubiquitous devices and link them with web resources, services, software and humans as business processes' components. UBIWARE will also consider sensors, sensor networks, embedded systems, alarm detectors, actuators, communication infrastructure, etc. as "smart objects" and will provide similar care to them as to other resources.

The innovative nature of UBIWARE is demonstrated well by the very idea of the case in Section 4.3, the smart service desk. An operator's service desk is an important service element which, however, still remains largely non-automated and thus has a low effectiveness. Any automation of such a process would require integration and complex interoperability of highly heterogeneous components, including hardware (e.g. customer's equipment), software and database systems, and humans (the customer, the service desk operator, and experts). In result, the possibility of automation of the service desk was not really considered before. However, the introduction of the UBIWARE concept has led to realizing that some automation can be possible after all.

REFERENCES

Brock, D. and Schuster, E. (2006). On the semantic web of things. In *Semantic Days 2006, Stavanger, Norway, April 26-27*.

Buckley, J. (2006). From RFID to the Internet of Things: Pervasive networked systems. In *Report on the Conference organized by DG Information Society and Media, Networks and Communication Technologies Directorate, Brussels, March 6-7*.

Cabri, G., Ferrari, L., and Zambonelli, F. (2004). Role-based approaches for engineering interactions in large-scale multi-agent systems. In *Software Engineering for Multi-Agent Systems II, LNCS v.2940*, pages 19–25. Springer.

Jennings, N. (1993). Commitments and conventions: The foundation of coordination in multiagent systems. *Knowledge Engineering Review*, 8(3):223–250.

Jennings, N. (2000). On agent-based software engineering. *Artificial Intelligence*, 117(2):277–296.

Jennings, N. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41.

Katasonov, A. and Terziyan, V. (2007). SmartResource platform and Semantic Agent Programming Language (S-APL). In *Proc. 5th Conf. Multi-Agent Technologies, LNAI v.4687*, pages 25–36. Springer.

Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1):41–50.

Khriyenko, O. (2007). 4i (FOR EYE) technology: Intelligent interface for integrated information. In *Proc. 9th Conf. Enterprise Information Systems, Volume 5*, pages 278–281.

Lassila, O. (2005). Using the Semantic Web in mobile and ubiquitous computing. In *Proc. IFIP WG12.5 Conf. Industrial Applications of Semantic Web*, pages 19–25. Springer.

Mamei, M. and Zambonelli, F. (2006). *Field-Based Coordination for Pervasive Multiagent Systems*. Springer.

Naumenko, A., Katasonov, A., and Terziyan, V. (2007). A security framework for smart ubiquitous industrial resources. In *Proc. 3rd Conf. Interoperability of Enterprise Software and Applications*, pages 183–194. Springer.

Tamma, V., Aart, C., Moyaux, T., Paurobally, S., Lithgow-Smith, B., and Wooldridge, M. (2005). An ontological framework for dynamic coordination. In *Proc. 4th International Semantic Web Conference'05, LNCS vol. 3729*, pages 638–652. Springer.

Vazquez-Salceda, J., Dignum, V., and Dignum, F. (2005). Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 11(3):307–360.

SENSOR AND ACTUATOR FAULT ANALYSIS IN ACTIVE SUSPENSION IN VIEW OF FAULT-TOLERANT CONTROL

Claudio Urrea and Marcela Jamett
Departamento de Ingeniera Eléctrica, Av. Ecuador 3519, Est. Central, Santiago, Chile
e_urrea@usach.cl, m_jamett@usach.cl

Keywords: Active suspension, sensor and actuator faults, full-vehicle suspension model.

Abstract: This paper shows the first step of a fault tolerant control system (FTCS) to control active suspension on a full-car suspension model. In this paper, the elimination of the inevitable pitch and roll actions of a spring suspension between each axle and the body of a vehicle is studied. An actuator (linear motor) producing an electromagnetic force and a pneumatic force acting simultaneously on the same output element is used. This linear motor acts as a force generator that compensates instantly for the disturbing effects of the road surface. Simulation results to illustrate the system's performance in front of the occurrence of sensor and actuator faults are shown.

1 INTRODUCTION

Vehicle suspension systems have developed over the last 100 years to a very high level of sophistication (Buckner, Schuetze, and Beno, 2000; Fukao, Yamawaki, and Adachi, 2000). Most vehicle today use a passive suspension system employing some type of springs in combination with hydraulic or pneumatic shock absorbers, and linkages with tailored flexibility in various directions. These suspension system designs are mostly based on ride analysis.

Traditionally automotive suspension designs have been a compromise between the three conflicting criteria of road holding, load carrying and passenger comfort. In fact, despite the wide range of designs currently available by using passive components, we can only offer a compromise between these conflicting criteria by providing spring and damping coefficients with fixed rates[1].

On the other hand, active suspensions have been extensively studied in the last three decades (Giua, Seatzu, and Usai, 2000; Lefebvre, Chevrel,and Richard, 2001; Lakehal-Ayat, Diop, and Fenaux,2002). In an active suspension the interaction between vehicle body and wheel is regulated by an actuator of variable length capable of supplying the entire control force system's requirements.

Ride comfort in ground vehicles usually depends on a combination of vertical motion (heave) and angular motion (pitch and roll). Active suspension is characterized by a built-in actuator which can generate control forces to suppress the above mentioned roll and pitch motions.

Including the dynamics of the hydraulic system consisting of fluids, valves, pumps, etc., complicates the active suspension control problem even further since it introduces nonlinearities to the system. It has been noted that the hydraulic dynamics and fast servo-valve dynamics make controls design very difficult (Karlsson, Teely, and Hrovatz, 2001; Alleyne, and Hedrick, 1995). The actuator dynamics significantly change the vibrational characteristics of the vehicle system (Engelman, and Rizzon, 1993). Using a force control loop to compensate for the hydraulic dynamics can destabilize the system (Alleyne, Liu, and Wright, 1998). This full nonlinear control problem of active suspensions has been investigated using several approaches including optimal control based on a linearized model (Engelman, and Rizzon, 1993), adaptive nonlinear control (Alleyne, and Hedrick, 1995), and adaptive control using backstepping (Karlsson, Teely, and Hrovatz, 2001). These schemes use linear approximations for the hydraulic dynamics or they neglect the servo-valve model dynamics in which a current or voltage is what ultimately controls the opening of the valve to allow flow of hydraulic fluid to or from the suspension system. However, nowadays a novel family of highly dynamic electro-magnetic direct drives exit, *i.e.* servomotors

[1] Components for passive suspension can only store and dissipate energy in a pre-determined manner.

that looks like a hydraulic piston with acceleration rates of over 200 $[m/s^2]$ make cyclic movement at several Hertz possible.

Active suspension control systems reduce undesirable effects by isolating car body motion from vibrations at the wheels, but their component failures/faults are inevitable and unpredictable, and without careful and prompt treatment, they tend to develop into the severe total failure of the whole system. Continued operation of these systems has both economic and safety implications. Every mechanical system is vulnerable to faults that can lead to failure of the complete system, unless mitigating strategies are included at the design stage (Noura, Theilliol, and Sauter, 2000). Control element failures not only degrade the performance of control systems, but also may introduce instability and thus can cause serious operation and safety problems. Therefore, fault tolerance has been one of the major issues in process control.

In automated systems, the goal of the fault-tolerance is to continue operation in spite of failures, if this is possible. A general problem has been that fault conditions could not be treated as an integrated part of system design. Therefore, automated systems to provide uninterrupted service, even in the presence of failures are required (Zhang, Jiang, 2002).

Most of the past work uses the quarter-car model, which includes only two degree-of-freedom of the vehicle motion in the vertical direction (Lakehal-Ayat, Diop, and Fenaux,2002). In general, the heave, pitch and roll motions are coupled and an impulse at the front or rear wheels excites all three motions. This means that pitch-, heave- and roll-controllers cannot be independently designed. Therefore, we take a model based in (Ikenaga, Lewis, Campos, Davis, 2000) including the full vehicle suspension dynamics considering heave, pitch and roll motions.

The paper is organized as follows: in section 2 the system description is given. In section 3, the state-space model for the full-vehicle suspension model is presented. In section 4, the vehicle controller is developed. Section 5 presents fault analysis and some simulation results. Finally, in section 6, the conclusions and outline future work are discussed.

2 SYSTEM DESCRIPTION

2.1 Full-Vehicle Suspension Model (Seven DOF System)

In this work, a full-vehicle suspension mathematical model depicted in Fig.1 is considered.

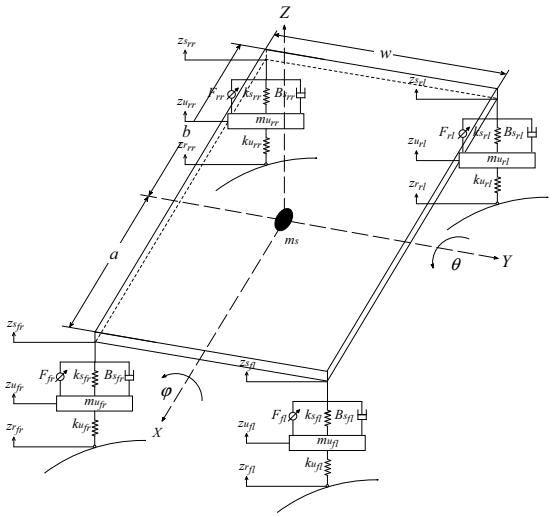


Figure 1: Full-vehicle suspension model.

where the followings parameters and variables are taken which respect to the static equilibrium position (Ikenaga, Lewis, Campos, Davis, 2000):

- ms is sprung mass $[kg]$,
- mu is unsprung mass $[kg]$,
- Ks_{fl} is front-left suspension spring stiffness $[N/m]$,
- Ks_{fr} is front-right suspension spring stiffness $[N/m]$,
- Ks_{frl} is rear-left suspension spring stiffness $[N/m]$,
- Ks_{rr} is rear-right suspension spring stiffness $[N/m]$,
- Bs_{fl} is front-left suspension damping $[N/m/s]$,
- Bs_{fr} is front-right suspension damping $[N/m/s]$,
- Bs_{frl} is rear-left suspension damping $[N/m/s]$,
- Bs_{rr} is rear-right suspension damping $[N/m/s]$,
- Ku_{fl} is tire-left spring stiffness $[N/m]$,
- Ku_{fr} is tire-right spring stiffness $[N/m]$,
- Ku_{frl} is tire-left spring stiffness $[N/m]$,
- Ku_{rr} is tire-right spring stiffness $[N/m]$,
- a is length between front of vehicle and center of gravity of sprung mass $[m]$,
- b is length between rear of vehicle and center of gravity of sprung mass $[m]$,
- w is width of sprung mass $[m]$,
- I_{xx} is roll axis moment of inertia $[kg \cdot m^2]$,
- I_{yy} is pitch axis moment of inertia $[kg \cdot m^2]$,

- F_{fl} is force at the front-left suspension $[N]$,
- F_{fr} is force at the front-right suspension $[N]$,
- F_{rl} is force at the rear-left suspension $[N]$,
- F_{rr} is force at the rear-right suspension $[N]$,
- Zr_{fl} is terrain disturbance heights at the front-left wheel $[m]$,
- Zr_{fr} is terrain disturbance heights at the front-right wheel $[m]$,
- Zr_{rl} is terrain disturbance heights at the rear-left wheel $[m]$,
- Zr_{rr} is terrain disturbance heights at the rear-right wheel $[m]$,
- g is the constant of graveness in the terrestrial surface, 9.80665 $[m/s^2]$.

In this model, the car body is represented as a *sprung mass*, and the wheels are represented as an *unsprung mass* connected to the ground via the tire spring. The tire is an *undamped spring* between the axle and the ground. The suspension consists of passive dampers in parallel with four actuators and four springs.

2.2 Actuators

The suspension actuators are taken to be a force actuator acting between the car body and the axle of the car. The chosen *ServoRam*^{TM2} actuator, depicted in Fig. 2, produces an electromagnetic force and a pneumatic force acting simultaneously on the same output element. This electromagnetic actuator has zero mechanical hysteresis, since the force is applied directly to the output element. It has zero electrical hysteresis because a microamp in one direction produces a positive force, and a microamp in the opposite direction produces a negative force, so that the force output is an exactly linear function of the current input. A linear transducer measures the position of the piston. The small control time constant allows the force to be changed at a rate of thousands of Newtons per millisecond, so the suspension system can instantly adapt to every road condition. These linear motors consists of just two parts: the fixed stator and the moveable slider. These two parts are not connected by slip rings or by cables. Since the linear stroke directly without the use of mechanical gears, belts or ball screws, there is no wear or mechanical play. In a practical point of view, the ram is placed between the wheel point (on the vehicle chassis) and the wheel stub axle, so as to carry all the vertical forces, see Fig. 3. The forces transferred from the wheel to the chassis may be precisely controlled by the electromagnetic forces. A force-measuring transducer may be used to control the current to the coil system, so as to maintain the total upward force at a constant value, irrespective of the wheel vertical motion. The desired value of this constant force may be determined in turn by the output from a wheel-point accelerometer, so as to hold the vehicle steady against pitch and roll motions, for example.

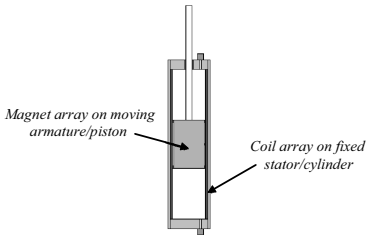


Figure 2: Simplified schematic of the *ServoRam*^{TM} actuator.

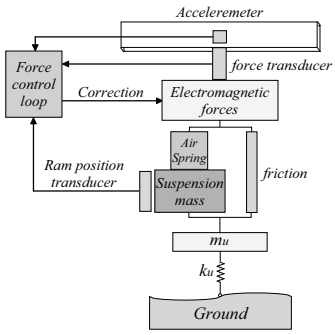


Figure 3: *ServoRam*^{TM} actuator system for active suspension.

3 STATE SPACE-MODEL

The governing equations of this system are presented considering the following state variables (Ikenaga, Lewis, Campos, Davis, 2000):

- $x_1 = z$ is the heave position (ride height of sprung mass),
- $x_2 = \dot{z}$ is the heave velocity (payload velocity of sprung mass),
- $x_3 = \theta$ is the pitch angle,
- $x_4 = \dot{\theta}$ is the pitch angular velocity,
- $x_5 = \phi$ is the roll angle,
- $x_6 = \dot{\phi}$ is the roll angular velocity,

[2]Name applied to the AMT® ram technology, and all AMT® electromagnetic rams are, in fact, *ServoRam*^{TM}. This technology has been developed over the last 10 years by AMT's Chief Scientist, Phillip Denne.

- $x_7 = Zu_{fl}$ is the front-left wheel unsprung mass height,
- $x_8 = \dot{Z}u_{fl}$ is the front-left wheel unsprung mass velocity,
- $x_9 = Zu_{fr}$ is the front-right wheel unsprung mass height,
- $x_{10} = \dot{Z}u_{fr}$ is the front-right wheel unsprung mass velocity,
- $x_{11} = Zu_{rl}$ is the rear-left wheel unsprung mass height,
- $x_{12} = \dot{Z}u_{rl}$ is the rear-left wheel unsprung mass velocity,
- $x_{13} = Zu_{rr}$ is the rear-right wheel unsprung mass height,
- $x_{14} = \dot{Z}u_{rr}$ is the rear-right wheel unsprung mass velocity,

Linear differential equations that describe the dynamics can be formulated as:

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= (F_{fl}+F_{fr}+F_{rl}+F_{rr}-(Ks_{fl}+Ks_{fr}+Ks_{rl} \\
&\quad +Ks_{rr})\cdot x_1-(Bs_{fl}+Bs_{fr}+Bs_{rl}+Bs_{rr})\cdot x_2\cdot \\
&\quad +(a\cdot(Ks_{fl}+Ks_{fr})-b\cdot(Ks_{rl}+Ks_{rr}))x_3+ \\
&\quad (a\cdot(Bs_{fl}+Bs_{fr})-b\cdot(Bs_{rl}+Bs_{rr}))\cdot x_4+ \\
&\quad Ks_{fl}\cdot x_7+Bs_{fl}\cdot x_8+Ks_{fr}\cdot x_9+Bs_{fr}\cdot x_{10}+ \\
&\quad Ks_{rl}\cdot x_{11}+Bs_{rl}\cdot x_{12}+Ks_{rr}\cdot x_{13}+Bs_{rr}\cdot x_{14}) \\
&\quad /ms-g \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= (-a\cdot(F_{fl}+F_{fr})+b\cdot(F_{rl}+F_{rr})+(a\cdot(Ks_{fl} \\
&\quad +Ks_{fr})-b\cdot(Ks_{rl}+Ks_{rr}))\cdot x_1+(a\cdot(Bs_{fl} \\
&\quad +Bs_{fr})-b\cdot(Bs_{rl}+Bs_{rr}))\cdot x_2-(a^2\cdot(Ks_{fl} \\
&\quad +Ks_{fr})+b^2\cdot(Ks_{rl}+Ks_{rr}))\cdot x_3-(a^2\cdot(\\
&\quad Bs_{fl}+Bs_{fr})+b^2\cdot Bs_{rl}+Bs_{rr}))\cdot(x_4-a\cdot \\
&\quad Ks_{fl}\cdot x_7-a\cdot Bs_{fl}\cdot x_8-a\cdot Ks_{fr}\cdot x_9-a\cdot \\
&\quad Bs_{fr}\cdot x_{10}+\ b\cdot Ks_{rl}\cdot x_{11}+b\cdot Bs_{rl}\cdot x_{12}+b\cdot \\
&\quad Ks_{rr}\cdot x_{13}+b\cdot Bs_{rr}\cdot x_{14})/I_{yy}
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= ((F_{fl}-F_{fr}+F_{rl}-F_{rr})-\frac{w}{2}\cdot((Ks_{fl}+Ks_{fr} \\
&\quad +Ks_{rl}+Ks_{rr})\cdot x_5+(Bs_{fl}+Bs_{fr}+Bs_{rl}+ \\
&\quad Bs_{rr}))\cdot x_6+Ks_{fl}\cdot x_7+Bs_{fl}\cdot x_8-Ks_{fr}\cdot\ x_9 \\
&\quad -Bs_{fr}\cdot x_{10}+Ks_{rl}\cdot x_{11}+Bs_{rl}\cdot x_{12}-Ks_{rr}\cdot \\
&\quad x_{13}+Bs_{rr}\cdot x_{14})/(2\cdot w/I_{xx}) \\
\dot{x}_7 &= x_8 \\
\dot{x}_9 &= x_{10} \\
\dot{x}_{10} &= (-F_{fr}+Ks_{fr}\cdot x_1+Bs_{fr}\cdot x_2-a\cdot Ks_{fr}\cdot x_3 \\
&\quad -a\cdot Bs_{fr}\cdot x_4-\frac{w}{2}\cdot Ks_{fr}\cdot x_5-\frac{w}{2}\cdot Bs_{fr}\cdot x_6 \\
&\quad -(Ku_{fr}+Ks_{fr})\cdot x_9-Bs_{fr}\cdot x_{10}+Ku_{fr}\cdot \\
&\quad Zr_{fr})/mu_{fr}-g \\
\dot{x}_{11} &= x_{12} \\
\dot{x}_{12} &= (-F_{rl}+Ks_{rl}\cdot x_1+Bs_{rl}\cdot x_2+b\cdot Ks_{rl}\cdot x_3+ \\
&\quad b\cdot Bs_{rl}\cdot x_4+\frac{w}{2}\cdot Ks_{rl}\cdot\ x_5+\frac{w}{2}\cdot Bs_{rl}\cdot x_6- \\
&\quad (Ku_{rl}+Ks_{rl})\cdot x_{11}-Bs_{rl}\cdot x_{12}+Ku_{rl}\cdot Zr_{rl}) \\
&\quad /mu_{rl}-g \\
\dot{x}_{13} &= x_{14} \\
\dot{x}_{14} &= (-F_{rr}+Ks_{rr}\cdot x_1+Bs_{rr}\cdot x_2+b\cdot Ks_{rr}\cdot x_3+ \\
&\quad b\cdot Bs_{rr}\cdot x_4-\frac{w}{2}\cdot Ks_{rr}\cdot x_5-\frac{w}{2}\cdot Bs_{rr}\cdot x_6- \\
&\quad (Ku_{rr}+Ks_{rr})\cdot x_{13}-Bs_{rr}\cdot x_{14}+Ku_{rr}\cdot\ Zr_{rr}) \\
&\quad /mu_{rr}-g
\end{aligned}
\tag{2}
$$

This system can be summarized by the following linear space-state representation:

$$
\begin{aligned}
\dot{x}(t) &= A\cdot x(t)+B\cdot u(t)+B_p\cdot u_p(t) \\
y(t) &= C\cdot x(t),
\end{aligned}
\tag{3}
$$

where:

- $x \in \Re^{14x1}$ is the system state vector,
- $u \in \Re^{4x1}$ is a vector composed of the control forces. $u = [F_{fl}\ F_{fr}\ F_{rl}\ F_{rr}]^{\mathrm{T}}$,
- $u_p \in \Re^{5x1}$ is a vector whose components are the disturbance inputs. $u_p = [g\ Zr_{fl}\ Zr_{fr}\ Zr_{rl}\ Zr_{rr}]^{\mathrm{T}}$,
- $y \in \Re^{3x1}$ is the system output vector. $y = [Z\ \theta\ \varphi]^{\mathrm{T}}$,
- A, B, B_p, and C are constant matrices of appropriate dimensions.

4 CONTROLLER MODEL

The controller design considers:

- Control loops that stabilize heave, pitch and roll responses,

- Input decoupling transformation that blends the inner and outer control loops allowing streamlined design yet gives performance better than over-simplified decoupled techniques.

By employing a state observer, the full state feedback is available for the entire vehicle, and pitch angle is inferred from a combination of rate sensors and accelerometers. The control law $\bar{u}$ is computed by a state feedback associated with the integral of the tracking error: $\bar{u}$ = - [K_1 K_2] · [x q]$^{\mathrm{T}}$ = - $K \cdot \bar{x}$; K_1 and K_2 are computed according to Eq.4. This controller design is illustrated in Fig.4.

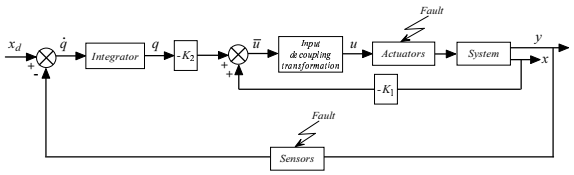


Figure 4: Controller block diagram.

The control procedure can be briefly summarized as follows: Let us consider the following optimization problem.

The performance index to be minimized is:

$$J = \frac{1}{2} \cdot \int_0^{\infty} [\bar{x}^T(t) \cdot Q \cdot \bar{x}(t) + \bar{u}^T(t) \cdot R \cdot \bar{u}(t)] dt, \qquad (4)$$

where:

- $\bar{x} \in \Re^{17x1}$ is the augmented system state vector. $\bar{x}$ = [x^{T} q^{T}]$^{\mathrm{T}}$,
- $\bar{u} \in \Re^{3x1}$ is a vector whose components are the control forces, provided by the actuators, for heave (F_z), pitch (F_θ) and roll (F_φ),
- $Q \in \Re^{17x17}$ is positive semi-definite matrix,
- $R \in \Re^{3x3}$ is tuning diagonal matrix.

The good performance of the suspension system is related to the minimization of the term $\bar{x}^T \cdot Q \cdot \bar{x}$ and an adequate choice of R, because the comfort depends of the term $\bar{u}^T \cdot R \cdot \bar{u}$. The optimal control strategy that minimizes the cost function was found to be $\bar{u}(t)$ = - $K \cdot \bar{x}(t)$, where the gain matrix Kcan be computed by solving an algebraic Riccati equation. So, a set of LQR optimal feedback gains corresponding to different weighting factors in the quadratic function J is chosen. The feedback control is designed to increase the relative damping of a particular mode of motion in the system by augmenting one or more of the coefficients of the equation of motion by actuating the control signals in response to motion feedback variables.

From Fig.1, the equivalent relationship between $F_z(t)$, $F_\theta(t)$ and $F_\varphi(t)$, and the forces generated by the actuators can be defined by:

$$\begin{bmatrix} F_{fl}(t) \\ F_{fr}(t) \\ F_{rl}(t) \\ F_{rr}(t) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \frac{b}{(a+b)} & \frac{-1}{(a+b)} & \frac{1}{w} \\ \frac{b}{(a+b)} & \frac{-1}{(a+b)} & \frac{-1}{w} \\ \frac{a}{(a+b)} & \frac{1}{(a+b)} & \frac{1}{w} \\ \frac{a}{(a+b)} & \frac{1}{(a+b)} & \frac{-1}{w} \end{bmatrix} \cdot \begin{bmatrix} F_z(t) \\ F_\theta(t) \\ F_\varphi(t) \end{bmatrix}, \qquad (5)$$

Therefore the nominal control is given by:

$$\begin{bmatrix} F_z(t) \\ F_\theta(t) \\ F_\varphi(t) \end{bmatrix} = - K \cdot \bar{x}(t), \qquad (6)$$

where $K \in \Re^{3x17}$.

5 FAULT ANALYSIS AND SIMULATION RESULTS

In the following simulations, the full-car suspension system was simulated for the input terrain disturbances $Zr(t)$ actuating between $2 \leq t \leq 8$ s:

$$Zr(t) = \begin{bmatrix} Zr_{fl}(t) \\ Zr_{fr}(t) \\ Zr_{rl}(t) \\ Zr_{rr}(t) \end{bmatrix} = \begin{bmatrix} 0.05 \cdot sin(w \cdot t) \\ 0.15 \cdot sin(w \cdot t) \\ 0.05 \cdot sin(w \cdot (t+\tau)) \\ 0.15 \cdot sin(w \cdot (t+\tau)) \end{bmatrix}, \qquad (7)$$

where:

- ω = 9 [rad/s], is the terrain disturbance frequency,
- τ = 0.1409 [s], is the time delay given by: $\tau = L/v = (a+b)/v$,
- L is the distance between the front and rear axles of the vehicle [m],
- v = 22 [m/s], is the speed at which the vehicle travels.

The given initial conditions are $\bar{x}$ = 0, and the required torque to be delivered by the actuators is determined as in fault-free cases. The objective here is to analyze the effects of a sensor- and actuator-faults on the active suspension.

5.1 Sensor Fault

At t = 5.5 s a sensor fault occurs in the second sensor measuring θ. This fault corresponds to a bias equal

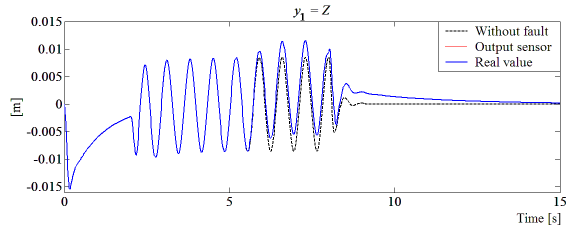


Figure 5: y_1: Heave position (ride height of sprung mass).

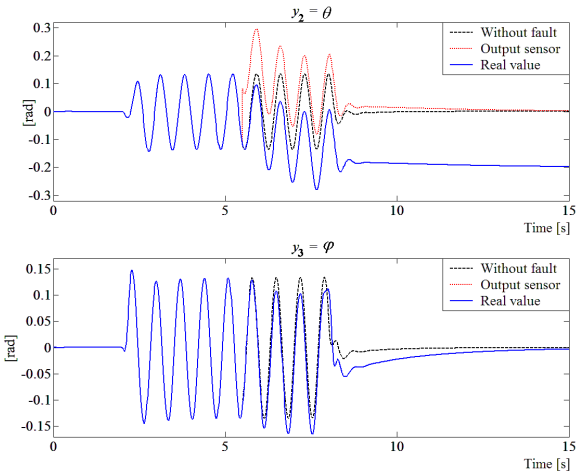


Figure 6: (a) y_2: Pitch angle; (b) y_3: Roll angle.

to 0.2 *rad*. Due to this fault, similar real value θ is shifted of - 0.2 *rad* from the output sensor which goes to its reference value. The real value of θ is far from its reference because the control input naturally reacts in the presence of the sensor fault. Figs. 5 and 6 show that the other outputs Z and φ are also affected by this fault but reach their reference value again.

The increase of the necessary force that should be applied to the actuators in front of this default is depicted in Figs. 7 and 8. Fig. 7(a) illustrates a saturation ($\pm$ 20000 $[N]$) reached in the actuator, in the presence of this fault. Figs. 7(b) and 8 show the other influence of this sensor fault on the control input.

5.2 Actuator Fault

Other experiments simulating actuator faults are performed. At t = 5.5 $[s]$, a reduction of 75% in the second actuator effectiveness (F_{fr}) is simulated. Figs. 9 to 12. illustrate simulation results. Figs. 9 and 10 show the permanent shift between the outputs with no fault and the outputs with fault. This shift is due to the fact that the other control inputs are affected by the fault due to the closed-loop and coupling between each other.The new necessary forces that should be applied in the actuators to compensate this loss of

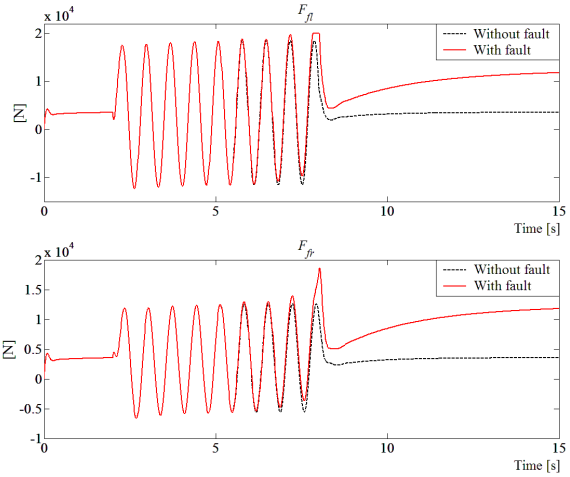


Figure 7: (a) F_{fl}: Force at the front-left suspension; (b) F_{fr}: Force at the front-right suspension.

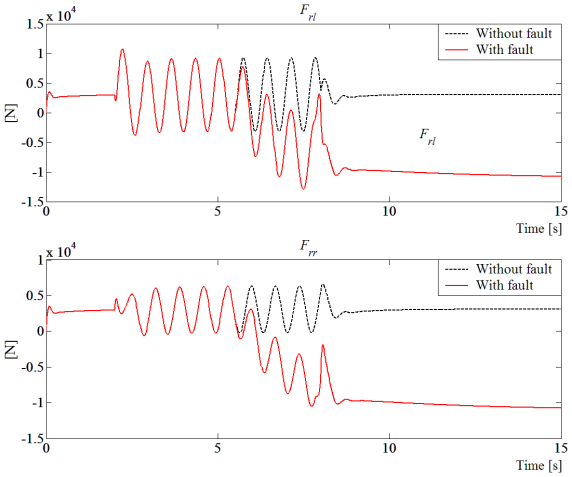


Figure 8: (a) F_{rl}: Force at the rear-left suspension; (b) F_{rr}: Force at the rear-right suspension.

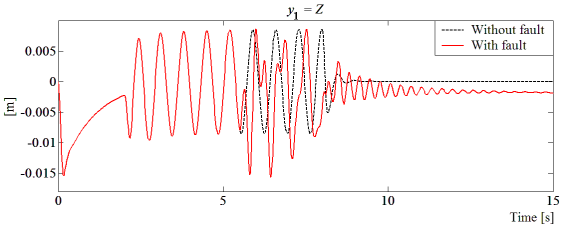


Figure 9: y_1: Heave position (ride height of sprung mass).

effectiveness are shown in Figs. 11 and 12. These force increases do not surpass the maximum allowable force limits in the actuators.

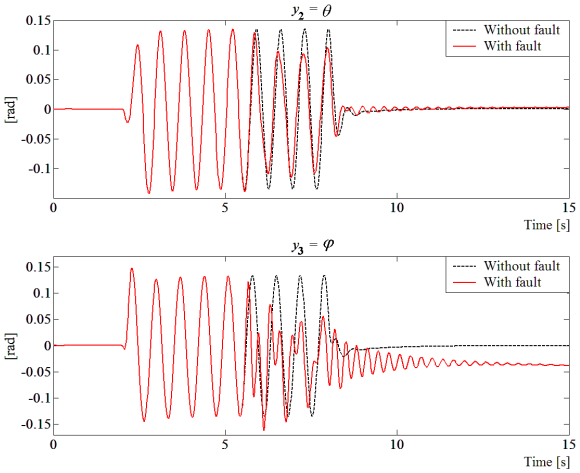


Figure 10: (a) y_2: Pitch angle; (b) y_3: Roll angle.

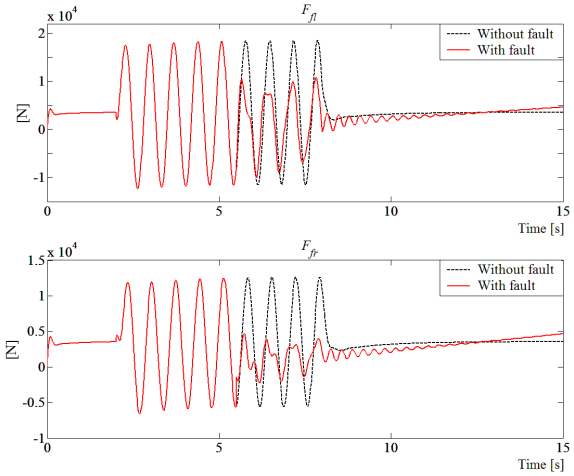


Figure 11: (a) F_{fl}: Force at the front-left suspension; (b) F_{fr}: Force at the front-right suspension.

6 CONCLUSIONS AND FUTURE WORK

In this paper, it was shown that active suspension can improve all three performance aspects *i.e.* passenger ride comfort, handling, and rattle space.

A nominal control law is designed for this active suspension system and the effect of the profile of the road is analyzed in the first part. The main aim of this work is the study of the influence of sensor and actuator faults on the control law.

The obtained results are realistic and show the importance of the design of a fault tolerant control system (FTCS) able to compensate this kind of fault. This work is just starting. Future work will also take into account the loss of a complete sensor or actuator in order to preserve safety and passengers' comfort.

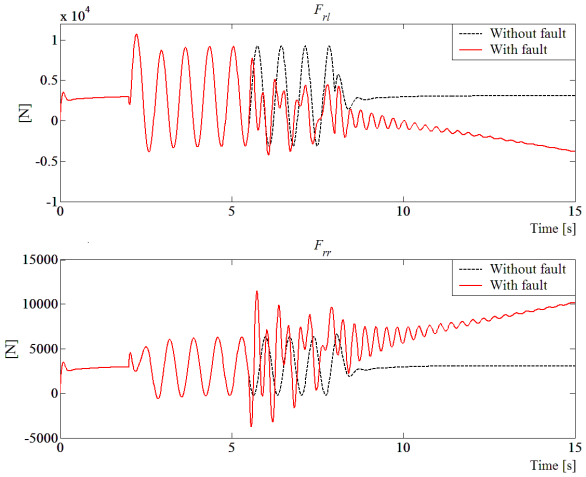


Figure 12: (a) F_{rl}: Force at the rear-left suspension; (b) F_{rr}: Force at the rear-right suspension.

ACKNOWLEDGEMENTS

This work was possible thanks to the support of DICYT – Universidad de Santiago de Chile, USACH, through Project 0607UO and Project 0607JD.

REFERENCES

Buckner G., Schuetze K., Beno J.: Active Vehicle Suspension Control Using Intelligent Feedback Linearization. American Control Conference, Chicago, Illinois, USA (2000) 4019–4024

Fukao T., Yamawaki A., Adachi N.: Nonlinear and H_∞ Control of Active Suspension Systems with Hydraulic Actuators. American Control Conference, Chicago, Illinois, USA (2000) 5125–5128

Giua A., Seatzu C., Usai G.: Active Axletree Suspension for Road Vehicles with Gain-Switching. 39th IEEE Conference on Decision and Control Sydney Convention and Exhibition Centre, Sydney, Australia (2000)

Lefebvre D., Chevrel P.,Richard S.: A Hinfinity-based control design methodology dedicated to active control of vehicle longitudinal oscillations. 40th IEEE Conference on Decision and Control Orlando, Florida, USA (2001) 99–104

Lakehal-Ayat M., Diop S., Fenaux E.: Development of a Full Active Suspension System. 15th Triennial World Congress IFAC, Barcelona, Spain (2002)

Karlsson N., Teely A., Hrovatz D.: A Backstepping Approach to Control of Active Suspensions. 40th IEEE Conference on Decision and Control Orlando, Florida, USA (2001) 4170–175

Alleyne A., Hedrick J.: Nonlinear Adaptive Control of Active Suspensions. IEEE Trans. Control Syst. Technol. **1** (1995) 94–101

Engelman G., Rizzon G.: Including the Force Generation Process in Active Suspension Control Formulation. American Control Conference, San Francisco, California, USA (1993)

Alleyne A., Liu R., Wright H.: On the Limitations of Force Tracking Control for Hydraulic Active Suspensions. American Control Conference, Philadelphia, USA (1998) 43–47

Noura H., Theilliol D., Sauter D.: Actuator Fault-Tolerant Control Design: Demostration on a Three-Tank-System. International Journal of Systems Science series 9 (2000) 1143–1155

Zhang Y., Jiang J.: Design of Restructurable Active Faut-Tolerant Control Systems. 15th Triennial World Congress IFAC, Barcelona, Spain (2002)

Ikenaga S., F. Lewis, Campos J., Davis L.: Active Suspension Control of Ground Vehicle Based on a Full-Vehicle Model. American Control Conference, Chicago, Illinois, USA (2000) 4019–4024

LEARNING DISCRETE PROBABILISTIC MODELS FOR APPLICATION IN MULTIPLE FAULTS DETECTION

Luis E. Garza Castañón
Department of Mechatronics and Automation, ITESM Monterrey Campus, Mexico
legarza@itesm.mx

Francisco J. Cantú Ortíz
Research and Graduate Programs Office, ITESM Monterrey Campus, Mexico
cantu@itesm.mx

Rubén Morales-Menéndez
Center of Innovation and Technology Design, ITESM Monterrey Campus, Mexico
rmm@itesm.mx

Keywords: Fault Detection, Bayesian Networks, Machine Learning, Power Networks.

Abstract: We present a framework to detect faults in processes or systems based on probabilistic discrete models learned from data. Our work is based on a residual generation scheme, where the prediction of a model for process normal behavior is compared against measured process values. The residuals may indicate the presence of a fault. The model consists of a general statistical inference engine operating on discrete spaces, and represents the maximum entropy joint probability mass function (pmf) consistent with arbitrary lower order probabilities. The joint pmf is a rich model that, once learned, allows us to address inference tasks, which can be used for prediction applications. In our case the model allows the one step-ahead prediction of process variable, given its past values. The relevant dependencies between the forecast variable and past values are learnt by applying an algorithm to discover discrete bayesian network structures from data. The parameters of the statistical engine are also learn by an approximate method proposed by Yan and Miller. We show the performance of the prediction models and their application in power systems fault detection.

1 INTRODUCTION

The problem of fault detection in processes has received great attention in last decades, and a wide variety of methods have been developed, most of them based on fault detection and isolation (FDI) techniques or in knowledge-based methods (Venkatasubramanian et al., 2003). FDI is based on the use of analytical redundancy rather than physical redundancy. In FDI the redundancy in static and dynamic relationships between process inputs and outputs is exploited (Frank, 1990). The methods used by FDI can be summarized in parity space approach, state estimation approach, fault detection filtering, and parameter identification approach. In every case, a mathematical model of process is required, either in state-space or input-output form, but most of the time these models are linear systems. Since many processes exhibits a nonlinear dynamics, several methods have been developed to deal with nonlinearities such as: decoupling approach, nonlinear observers and nolinear parity spaces (Zhang and Ding, 2005). These methods are limited to work well in a small region around the point of operation or are adequate just for a limited class of nonlinear systems.

In the other hand, Knowledege-based methods rely on qualitative model descriptions in the form of neural networks, Bayesian networks, fuzzy logic or qualitative reasoning. Neural networks are widely used in fault detection and diagnosis (Xu and Chow, 2005) but they represent black box models and can not deal with missing information. Fuzzy logic uses a database with IF-THEN rules which use linguistic variables. The problem with fuzzy logic is that can not deal with incomplete information in explicit form and the overall dimension of rules may blow up strongly even for small processes (Isermann, 1997). The methods based in qualitative reasoning require a set of qualitative differential equations between process variables not easy to obtain for complex processes. Other machine learning approaches used in fault detection can be found in (Sedighi et al.,

2005; Davy et al., 2006). Bayesian networks (BNs) have been lately used in fault detection and diagnosis (Yongli et al., 2006; Matsuura and Yoneyama, 2004), as they represent robust models for nonlinear systems able to deal with missing information and noise. A potential problem in BNs is the time for inference process in large domains.

A recent trend is the combination of methods to take advantage of the best aspects of every approach (Gentil et al., 2004). Our work is mainly focus in this direction.

Our fault detection method is based on a prediction model obtained from the process normal behavior time series. We can find in technical literature many approaches using machine learning techniques for time series prediction. For instance, in (Luque et al., 2007) an evolutionary approach is applied to learn a set of rules to predict local behavior of time series. In (Chen and Zhang, 2005) an adaptive network based fuzzy inference system (ANFIS) is used to predict chaotic and traffic flow time series. In (Vanajakshi and Rilett, 2007) a support vector machine (SVM) approach is used to predict traffic flow time series. In (Ma et al., 2007) evolving recurrent neural networks are presented which predict chaotic time series. None of these methods address the problem of missing information.

In our approach, we generate residuals by comparing actual measurements against a prediction given by a normal behavior model. The model structure and parameters are learned by applying machine learning techniques. The residuals behavior indicate the existence of a fault.

We test our approach by diagnosing multiple-faults events in a large power transmission network and show promising results.

2 OUR APPROACH

A general overview of the proposed approach is shown in Figure 1. Basically we generate residuals from the comparison between a process normal behavior model and the actual process values. We substitute the classical models of process normal behavior (eg. discrete linear models) with a discrete probabilistic function, whose parameters and structure are learned off-line from normal behavior process data. The probabilistic function is a general statistical inference engine, which allows inference to know the future value of a process variable, given its past values. In our case, we predict the one step-ahead value of the process variable given a set of past values. The set of relevant process variable values having direct influence on the forecast variable, are learned off-line by using an algorithm to learn discrete Bayesian networks. The output of this algorithm is a graphical causal structure, which is simplified by selecting the Markov blanket of the forecast process variable. This kind of compact probabilistic models are robust to noise, incomplete information and nonlinearities.

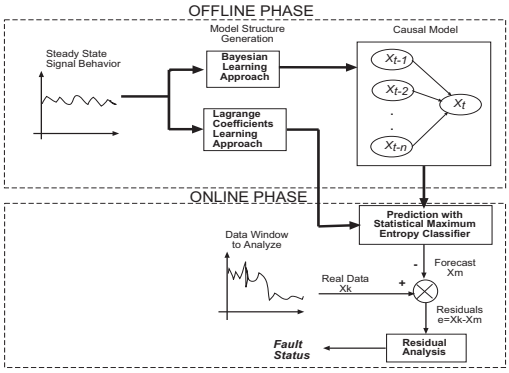


Figure 1: An overview of the fault detection approach based on machine learning models.

In the decision and isolation step, we generate residuals from the comparison between the output of the probabilistic model and actual process variable values. The identification of the fault is performed by a comparison of the residuals against a set of given thresholds.

The architecture of the method is split in two phases: the off-line phase and the online phase. The off-line phase learns the model structure and parameters, and the online phase take the decision regarding the presence of a fault.

2.1 The Off-line Phase

The off-line phase generates a discrete process normal behavior model from data, by applying machine learning techniques which learn both: the model structure and the parameters. The models can include several variables having an influence over the state of the process. The procedure to generate the models starts with the discretization of continuous variables, by using fixed bins or fuzzy clustering. The fixed interval width discretization, merely divides the range of observed values in equal sized bins. The general idea with multivariate discretization approach based on the *fuzzy C-means* algorithm (Wang, 1997), is that rather than discretizing independently each variable, we find the centroids of the c clusters defined by the user, and assign each instance of the multivariate series to the closest cluster [1].

[1]According to a defined metric. We use a simple Euclidean distance metric

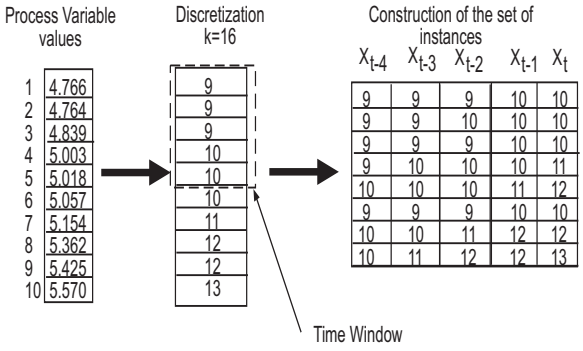


Figure 2: Selection of attributes with $M_d = 5$.

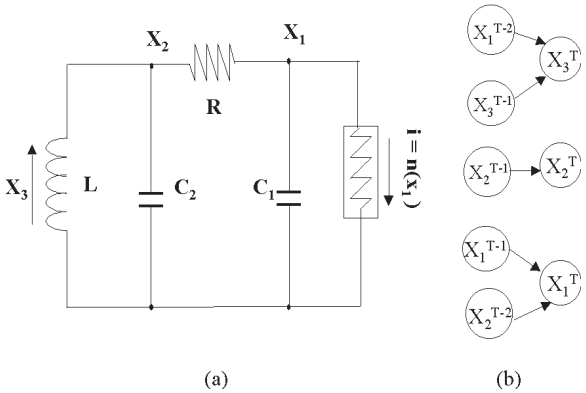


Figure 3: (a) Chua's electric circuit, (b) Learned graphical models from data.

The process of discretization allows the use of standard discrete Bayesian networks learning algorithms and the implementation of the algorithm to learn the general statistical inference engine parameters.
Once the discretization phase has been achieved, the next issue in the construction of the model, is the specification of the set of attributes and the instances, to be supplied to the algorithm that learns the discrete Bayesian network structure. This is not a trivial issue, since possibly we do not know anything about the lagged dependencies in the process variable dynamics. If we have observed a sample of N data for the variable X, the forecast or prediction variable X_t may depend on any of the past values $X_{t-1}, X_{t-2}, \ldots, X_{t-N}$. We solve this problem by selecting an initial set of attributes M_d [2] and keep adding attributes until a causal structure can be found. Although it is possible that different causal structures can be found, even a trivial structure with just two nodes, we can test each structure and select the more accurate. If a causal structure cannot be found with a discretization policy, then increase the number of bins, in fixed discretization policy, or increase the number of clusters, in the *fuzzy C-means* discretization policy, and again do the iterative selection of the size of attributes. An example of the selection of the attributes in a time series is shown in figure 2, with $M_d = 5$. The input to the discrete bayesian networks learning algorithm is thus a set of instances having the form $\{X_{t-M_d-1}, \ldots, X_t\}$. Notice we are not assuming beforehand anything regarding independence of variables or specific time dependencies. The algorithm that learns the Bayesian network structure tries to find such dependencies.

When the causal structure of the set of M_d attributes is found, we select our model from the Markov blanket of the prediction variable X_t. The Markov blanket in a BN consists of node's parents, its children and its children's parents. The Markov blanket forms a natural feature selection, as all features outside the Markov blanket can be safely deleted from the BN. We exploit this feature to produce a much smaller causal structure for our forecast model, without compromising the classification accuracy.
The prediction variable is the M_dth attribute, has $\mathcal{P}$ parents (variables influencing directly its value) and no children (other variables over which the forecast variable have an influence). We enforce this by specifying a variable ordering to the BN learning algorithm. For instance, Figure 3 shows the models obtained for an electrical circuit which behaves as a chaotic system. X_1 represents electrical current across the inductance L and X_2 and X_3 represent voltages at capacitors C_1 and C_2.

After we obtain the relevant past values for the forecast variable, we learn the parameters of the statistical inference engine based on the maximum entropy principle. This method can be stated as follows: Consider a random feature vector $\hat{F} = (\mathbf{F}, \mathbf{C})$, $\mathbf{F} = (F_1, F_2, \ldots, F_N)$, with $F_i \in \mathcal{A}_i$ and $\mathcal{A}_i$ the finite set $\{1, 2, 3, \ldots, |\mathcal{A}_i|\}$, and $\mathbf{C} \in \{1, 2, \ldots, K\}$. Denote the full discrete feature space by $\mathcal{G} \equiv \mathcal{A}_1 \times \mathcal{A}_2 \cdots \times \mathcal{A}_N \times \mathcal{C}$. Suppose we are given knowledge of all $(N(N-1)/2)$ pairwise pmf's $\{P[F_m, C], \forall m\}$ and wish to constrain the joint pmf $P[\mathbf{F}, \mathbf{C}]$ to agree with these. The pairwise probabilities typically are estimated from training set co-occurrence counts. The maximum entropy (ME) joint pmf consistent with these pairwise pmf's has the Gibbs form:

$$P[C = c | F = f] = \frac{exp\left(\sum_{i=1}^{N} \gamma(F_i = f_i, C = c)\right)}{\sum_{c'=1}^{K} exp\left(\sum_{i=1}^{N} \gamma(F_i = f_i, C = c')\right)} \quad (1)$$

[2] M_d is also the size of the time window, and the instances are formed sliding the time window through the complete time series. In a time series with N data we can have $N - M_d + 1$ instances

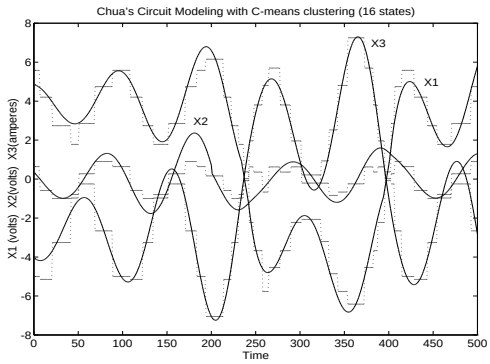


Figure 4: Modeling Chua's circut parameters with a C-Means clustering discretization method.

where

- F is the set of relevant past values for the forecast variable,
- C is the set of predicted variables.

The subset of model parameters (Lagrange multipliers) $\{\gamma(C_i = c_i, F = f), i = 1,\ldots,N, c_i = 1,\ldots,K, f = 1,\ldots,K\}$ are learned with a deterministic annealing algorithm. Where N is the number of relevant past values for the prediction variable, K is the number of discretization bins.
We need to supply following inputs to the Lagrange coefficients learning algorithm:

- A training set of $\mathcal{P}+1$ attributes with M instances,
- a training set support size $\mathcal{G}_S << \mathcal{G}$,
- an annealing parameter η,
- an annealing threshold ε,
- an annealing initial temperature T_{max} and final temperature T_{min}
- a ρ learning-rate parameter.

The inference engine provides a probability distribution of the forecast variable, given the evidence of relevant past values of forecast variable. We select the discrete state with highest probability and to make a comparison against the real data, we substitute the state by its correspondent real value. An example of modeling is shown in Figure 4.

2.2 The Online Phase

In order to perform process fault detection, the observations or measurements obtained from the process, have to be compared against the prediction given by the normal behavior model. From this comparison, the residuals are generated and then analyzed to give a decision about the behavior of the component.

If we denote X_t as the measurement of a component variable at time t, and $\hat{X}_t$ as the prediction of the component variable given by the steady state model, then the residual e_t is computed from:

$$e_t = X_t - \hat{X}_t \tag{2}$$

The differences between the steady-state model and the real data, e_t, are transformed to a filtered version of residuals, using the equation:

$$\bar{e_t} = \bar{e}_{t-1} + \lambda * (|e_t| - \bar{e}_{t-1})$$

The value of λ, between 0 and 1, represents the smoothing factor of the residuals. We refer to the average value of a set of filtered residuals as the error weighted moving average (EWMA) index. An example of EWMA residuals behavior in Chua's electrical circuit is shown in figure 5 under normal circumstances, and in figure 6 under an additive fault.

The fault decision is accomplished by comparing the actual filtered residuals against the limit thresholds of each fault mode. The limit thresholds are calculated previously from process data. In our case, we perform intensive simulations in a power transmission network which include single faults and different combinations of multiple faults.

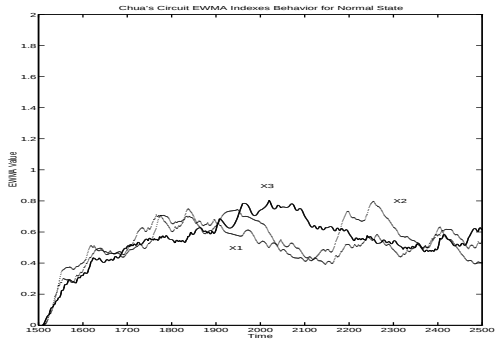


Figure 5: EWMA residuals behavior in normal operation of the three parameters in Chua's circuit.

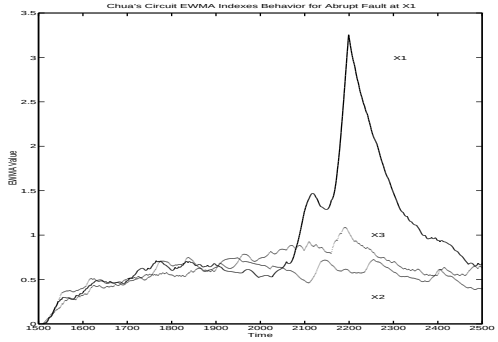


Figure 6: EWMA residuals behavior in an additive fault at X_1 in Chua's circuit.

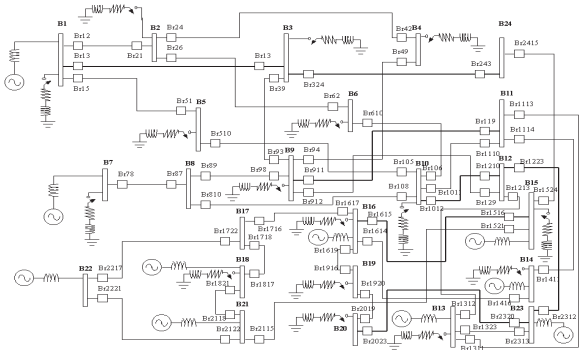


Figure 7: The electrical power network test system.

3 CASE STUDY

We illustrate the application of our approach in a simulated power transmission network, shown in fig. 7. The system consists of 24 nodes, 34 lines and 68 breakers. The electrical power network is supplied with the energy produced by three-phase generators. Ideally, the generators supply the energy to three-phase balanced loads, which means that every load has an identical impedance. In a balanced circuit, each phase has the same magnitude of voltage, but displaced 120 electrical degrees. In all simulations we include dynamic behavior by varying resistive-inductive loads in several nodes.

A fault in a electric network is any event that interfere with the normal flow of current. The faults in an electrical power network can be divided in two types: *symmetrical* faults and *unsymmetrical* faults. The symmetrical faults involve the three phases of the system, are relatively easy to evaluate, and represent about the 5 % of the fault cases. The unsymmetrical faults involve some kind of unbalance, and include line to ground faults and line to line faults. The line to ground faults represent about 70 % of the faults, and the line to line faults represent about 25 % of the cases (Grainger and Stevenson, 1994).
The diagnosis in large power networks is a difficult task, mainly due to overwhelming amount of data, the cascaded effect, and the uncertainty in the information. The main protection breakers of a node can be opened (as a secondary protection) by faults at neighbor nodes, giving rise to ambiguous diagnoses. The voltage measurements at a given node, are also perturbed by faults at neighbor nodes.
With our modeling approach, we represent the steady state dynamics of continuous signals (e.g. voltages) in every node, and detect different types of faults: symmetrical faults (e.g. a three-phase to ground fault) and unsymmetrical faults (e.g. a line-to-ground fault).

To evaluate the degree of success in the identification of the faulty components, we ran a set of 48 simulations in the power network. We randomly simulate simultaneous different types of faults in several nodes. The type of faults included *symmetrical* and *unsymmetrical* faults.

Table 1: Performance evaluation by type of fault.

Fault Type	*Correct*	*Wrong*	*% Accuracy*
A-B-C-GND	18	0	100.0
A-B-GND	12	0	100.0
A-GND	16	3	84.2
A-B	18	4	81.8
B-C	22	0	100.0
NO FAULT	20	7	74.0

The results obtained (see table 1) show that we were able to determine with great accuracy the *symmetrical* faults, but we have problems with false positive detections and line-to-line faults.
We also performed an evaluation with a level of 30 % of random missing information in the same test nodes data. The steady state models were learned with a training set of data with just 10% of random missing information. The computed EWMA indices remain almost in the same values ($\pm$2%) computed without missing information. The evaluation with missing information, delivered the same fault identification as the evaluation without missing information.

4 DISCUSSION

This approach is intended to work with data coming from multiple sources. The intention is to build, with this data, models which are robust to incomplete information and non-linearities. We have tested in some examples the capabilities of model to approximate nonlinear dynamics. The accuracy of the model, is related mainly to the level of discretization and the learning time of model's parameters. If we increase the level of discretization, we also need to increase the set support $\mathcal{G}_S$ of model's parameters learning algorithm, with the consequence of rising significatively the learning time. For instance, with 16 states and a set support size of 50 elements, learning time was 7.5 hours (using a desktop computer with a 1.3 GHz processor clock). If we increase the number of states to 32, the learning time was 12.5 hours. If we just increase the set support size for 16 states, from 50 to 80 elements, the learning time increases to 15 hours.
In summary, we do not think we have a restriction on the kind of applications we can tackle due to the accuracy of the model. All we need is a level of accuracy

enough to distinguish between normal operation and every type of fault. We think that a level of discretization of at most 32 states, will cover many of the fault detection applications.

5 CONCLUSIONS AND FUTURE WORK

We have presented a new approach to detect faults based on models learned by machine learning techniques. The model represents the process normal behavior and is used in a residual generation scheme where model output is compared against actual process values. The residuals generated from this comparison are used to indicate the existence of a fault. The compact learned models are robust to noise, missing information and nonlinearities. We apply our method in a very difficult domain, as it is an electrical power network. The noise in data, the cascaded effect, and the perturbation by neighbor nodes, makes the diagnosis task hard to achieve. We have shown good levels of accuracy in the determination of the real faulted components and the mode of fault, in multiple events, multiple mode fault scenarios, where missing information was given. We determine in experimental simulations that wrong node state identifications were mainly due to the overlapping between EWMA indices thresholds, giving rise to ambiguous fault decisions. We plan to reach higher levels of success with the help of more reliable signal change detection methods.

REFERENCES

Chen, D. and Zhang, J. (2005). Time series prediction based on ensemble anfis. In *Proceedings of the fourth International Conference on Machine Learning and Cybernetics*. IEEE.

Davy, M., Desorbry, F., Gretton, A., and Doncarli, C. (2006). An online support vector machine for abnormal events detection. In *Signal Processing 86 (2006)*. Elsevier.

Frank, P. (1990). Fault diagnosis in dynamic systems unisg analytical and knowledge based redundancy a survey and new results. In *Automatica*. Elsevier.

Gentil, S., Montmain, J., and Combastel, C. (2004). Combining fdi and ai approaches within causal-model-based diagnosis. In *IEEE Transactions on Systems, Man and Cybernetics, part B*. IEEE.

Grainger, W. and Stevenson, W. (1994). *Power Systems Analysis*. McGraw-Hill, USA.

Isermann, R. (1997). On fuzzy logic applications for automatic control, supervision, and fault diagnosis. In *IEEE Transactions on Systems, Man, and Cybernetics*. IEEE.

Luque, C., Valss, J., and Isasi, P. (2007). Time series forecasting by means of evolutionary algorithms. In *Proceedings of the Parallel and Distributed Processing Symposium 2007*. IEEE.

Ma, Q., Zheng, Q., Peng, H., Zhong, T., and Xu, L. (2007). Chaotic time series prediction based on evolving recurrent neural networks. In *Proceedings of the fourth International Conference on Machine Learning and Cybernetics*. IEEE.

Matsuura, J. P. and Yoneyama, T. (2004). Learning bayesian networks for fault detection. In *International Workshop on Machine Learning for Signal Processing*. IEEE.

Sedighi, A., Haghifam, M., and Malik, O. (2005). Soft computing applications in high impedance fault detection in distribution systems. In *Electric Power Systems Research 76 (2005)*. Elsevier.

Vanajakshi, L. and Rilett, L. (2007). Support vector machine technique for the short term prediction of travel time. In *Proceedings of the 2007 Intelligent Vehicles Symposium*. IEEE.

Venkatasubramanian, V., Rengaswamy, R., k. Yin, and Kavuri, S. (2003). A review of process fault detection and diagnosis part 1, part 2 and part 3. In *Computers and Chemical Engineering*. Elsevier.

Wang, L. (1997). *A Course in Fuzzy Systems and Control*. Prentice Hall, USA.

Xu, L. and Chow, M. (2005). Power distribution systems fault case identification using logistic regression and artificial neural network. In *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*.

Yongli, Z., Limin, H., and Jinling, L. (2006). Bayesian networks-based approach for power systems fault diagnosis. In *IEEE Transactions on Power Delivery*. IEEE.

Zhang, P. and Ding, S. X. (2005). A simple fault detection scheme for nonlinear systems. In *Proceedings of the 2005 IEEE International Symposium on Intelligent Control*. IEEE.

AUTOMATED SIZING OF ANALOG CIRCUITS BASED ON GENETIC ALGORITHM WITH PARAMETER ORTHOGONALIZATION PROCEDURE

Masanori Natsui and Yoshiaki Tadokorot
Dept. of Information and Computer Sciences, Toyohashi University of Technology
1-1 Hibarigaoka, Tempaku-cho, Toyohashi-shi, Aichi 441-8580, Japan
{*natsui, tadokoro*}*@signal.ics.tut.ac.jp*

Keywords: Evolutionary computation, Circuit synthesis, Analog circuits, Genetic algorithms, Principal component analysis.

Abstract: This paper presents a method for the automated sizing of analog circuits using genetic algorithm (GA). For the rapid and efficient exploration of GA, we introduce the idea of search space sphering and dimension reduction with principal component analysis (PCA). The potential capability of the system is demonstrated through the automated sizing of wide-swing current mirror circuit. Experimental results show that the system with PCA successfuly generates higher-performance circuits under given evaluation function on average, and the dimensionally reduction method further improves the performance of solutions.

1 INTRODUCTION

Electronic design automation (EDA) has matured as a technology to be universally accepted for producing highly integrated VLSI systems. However, there still remain many difficult design problems to be addressed in the advanced SoC (System on Chip) era.

In this paper, we propose an automated analog circuit design method based on evolutionary computation. An analog circuit inherently involves tradeoffs among a large number of performance metrics and the performance is determined by the complex and nonlinear nature of relations between the topology and parameter values (e.g. device sizes) of a circuit. Therefore, the analog circuit design often requires simultaneous optimization of circuit topology and parameter values. It requires the knowledge and experience of experts who had trained in a particular way to understand circuit/device technologies. This is the major motivation to introduce evolutionary optimization techniques for the design automation.

There are already some approaches to the evolutionary design of analog circuit structures(Koza et al., 1997; Lohn and Colombano, 1999; Shibata and Fujii, 2001). The reference (Koza et al., 1997) describes various analog circuit design including analog filters and amplifiers by means of *genetic programming* (GP). The reference (Lohn and Colombano, 1999) proposes a method of representing electronic circuit structures by "linear representation" and its application to analog circuit synthesis using *genetic algorithm* (GA). The reference (Shibata and Fujii, 2001) also applies GA to the synthesis of analog circuits based on current path based coding. These approaches appear to perform both topology and parameter optimization equally. Both approaches have the possibilities of finding best possible solutions under given conditions and constraints since both topology and parameter optimization are performed simultaneously. However, the simultaneous optimization often leads to an explosion of search space, which requires much computation time for finding a solution, and sometimes converges into a local optimum.

To make our method suitable for practical use, we have proposed a two-stage optimization process of analog circuits as follows: (i) topology-oriented optimization using evolutionary computation and (ii) detailed optimization of parameter values (e.g. aspect ratios of transistors) using a commercially available automatic circuit sizing software(Homma et al., 2000; Homma et al., 2002; Natsui et al., 2005; Natsui et al., 2007). This storategy have achieved a reduction in the size of search space and an improvement in the search efficiency.

In this paper, we mainly focus on an automated analog circuit sizing for a given circuit topology by

using GA. In order to accelerate the search efficiency of the circuit sizing system, we propose the idea of applying the parameter orthogonalization procedure based on *principal component analysis* (PCA) to the system. In the analog circuit design, the relative scale between circuit parameters such as the ratio of the length and width of a transistor or that of the widths of two transistors has an important role to determine circuit performance. If the system can consider the correlations between parameters that are specific to higher-fitness individuals and periodically reduce the search space, it can increase the search efficiency of the optimization process.

From this viewpoint, we introduce a search space reduction operation based on PCA. The principal component-based exploration realizes efficient parameter optimization, and additionally, dimensionality reduction method further reduces the search space. As a result, we can realize the high-performance optimization system which finds the optimal parameter set with less computation time.

The potential capability of the proposed system is demonstrated through the automated circuit sizing of a wide-swing current mirror, which is one of the most popular building block in analog circuit design. This paper is organized as follows: Section 2 presents the basic concept of the GA-based automated circuit sizing system and its extension using PCA. Section 3 demonstrates the experimental result. Section 4 is the conclusion and future prospects.

2 GA-BASED AUTOMATED CIRCUIT SIZING SYSTEM AND ITS EXTENSION USING PCA

2.1 Genetic Algorithm

Genetic algorithm (GA) can be regarded as a unique variation of evolutionary computation techniques (Back et al., 1997; Holland, 1975; Goldberg, 1989). In general, evolutionary methods mimic the process of natural evolution, the driving process for emergence of complex structures well-adapted to the given environment. The better an individual performs under the conditions the greater is the chance for the individual to live for a longer while and generate offspring. As a result, the individuals are transformed to the suitable forms on the designer's defined constraint.

Figure 1 shows the overall procedure of GA. At first, GA generates embryonic individuals randomly to form the initial population $P(0)$. Note that usually each individual is represented by a fixed-length bit string in the traditional GA, while our system applies *real-coded genetic algorithm* (RCGA) (Eshelman and Schaffer, 1993) which uses a real number vector representation of chromosomes.

```
program Genetic Algorithm;
 begin
  t := 0;
  {t: Number of generations.}
  initialize(P(t));
  {P(t): Population.}
  evaluate(P(t));
  while t ≤ Max. num. of gen. do
   begin
    C(t) := crossover(P(t));
    M(t) := mutation(P(t));
    evaluate(C(t) ∪ M(t));
    P(t+1) := select(C(t) ∪ M(t) ∪ P(t));
    t := t + 1;
   end
 end.
```

Figure 1: Typical flow of GA.

The next step is to evaluate a fitness function of all individuals in $P(t)$. A value for fitness is assigned to each individual depending on how close it actually is to solving the problem. Since typical analog circuits have various performance metrics, an analog circuit optimization problem is usually formulated as (1) a single-objective optimization problem where different performance objectives are combined to form a scalar objective, or (2) a multiobjective optimization problem based on Pareto-optimality(Somani et al., 2007). In our system, we apply the single-objective approach for the system simplicity and reasonable computational effort. Thus a synthetic fitness function for a circuit is defined by the combination of fitness functions for various performance objectives.

After the evaluation, the system selects a set of individuals having higher fitness values to perform evolutionary operations: *crossover* and *mutation*. The crossover recombines two individuals into two new offsprings. The mutation operation, on the other hand, changes the values of chosen genes randomly. There are many ways how to do crossover and mutation. In our system, we apply blend crossover (BLX-α)(Fig. 2) and uniform mutation. Blend crossover generates the offspring vectors $X_1 = (x_{11}, x_{12}, \cdots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \cdots, x_{2n})$ from parant vectors $P_1 = (p_{11}, p_{12}, \cdots, p_{1n})$ and $P_2 = (p_{21}, p_{22}, \cdots, p_{2n})$ by $x_{1i} = \gamma p_{1i} + (1-\gamma) p_{2i}$ and $x_{2i} = (1-\gamma) p_{1i} + \gamma p_{2i}$, where $\gamma = (1+2\alpha)u_i - \alpha$, $u_i \in [0,1]$ is a random value. Uniform mutation, on the other hand, changes the value of the element to a value chosen from the uniform distribution on the specified interval.

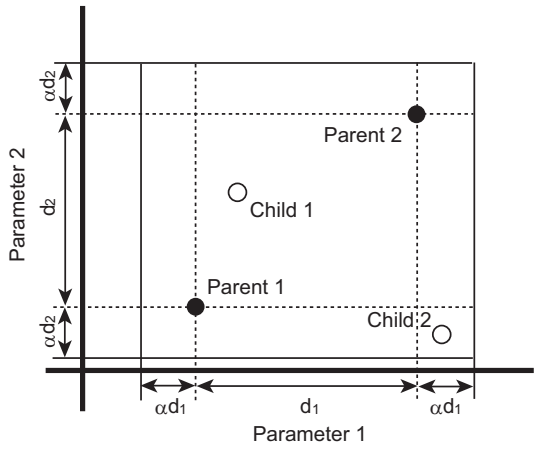


Figure 2: Blend crossover for two parameter vectors.

2.2 Search Space Reduction using PCA

Let us consider the search space reduction method of the automated analog circuit sizing on the basis of common characteristics of higher-fitness individuals. Given individuals which have averagely higher fitness values after a certain generations, one of the simplest way is to narrow down the search space of each parameter independently according to its mean and variance. If the parameters are orthogonalized and noncorrelated, this approach may work well. However in the case of analog circuit sizing, it sometimes fails and the exploration process converges to a local minimum since these parameters usually have strong correlations mutually. In order to perform effective search space reduction, we should apply an appropriate reduction method considering its relationship.

From this viewpoint, we utilize a coordinate transform procedure based on principal component analysis (PCA) to find a rotated orthogonal system such that the elements in the new coordinates are uncorrelated. PCA is a well known method for reducing the dimensionality by extracting components which are uncorrelated with each other, and has been used in a wide range of research including signal processing, statistics, and neural computing, as the powerful tool of data analysis and compression (Jolliffe, 2002). The exploration and search space correction based on the principal components and their variances have a possibility of enhancing the efficiency of the parameter optimization of GA.

The parameter orthogonalization procedure by PCA can be performed by the following steps:

1. Selects a set of higher-fitness individuals, where the parameter set corresponding to the ith individual is represented by $x_i = (x_{i1}, x_{i2}, \cdots, x_{iM})$.
2. Generates a parameter matrix
$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NM} \end{pmatrix},$$
where each row corresponds a parameter set of an individual.
3. Generates a matrix $\tilde{X} = (\tilde{x_{ij}})$ by normalizing values of the matrix X so as to the mean $\mu_j = \frac{1}{N}\sum_{i=1}^{N} \tilde{x_{ij}} = 0$ and the variance $\sigma_j^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x_{ij}} - \mu_j)^2 = 1$ for each column.
4. Calculates a variance-covariance matrix, eigenvalues and eigenvectors of $\tilde{X}$.
5. Calculates an orthogonalized parameter matrix $X' = \tilde{X}A = (x'_{ij})$ where A is a matrix such that each column contains coefficients for one principal component.
6. Applies $x'_i = (x'_{i1}, x'_{i1}, \cdots, x'_{iM})$ to the parameter set corresponding to the ith individual.

The orthogonalization procedure is performed periodically with a specific interval during the evolution process by GA. This means that the system always use the vectors of real number elements with $\mu = 0$ and $\sigma^2 = 1$ for the individual representation, while the search space of circuit parameters is periodically corrected depending on the distribution of higher-fitness individuals. Additionally, if the variation of the last principal components are sufficiently small, we can eliminate these components from an exploration and reduce the length of individuals since they contribute relatively little extra information of individuals. This dimensionality reduction further improves the search efficiency, and reduces the total computational time.

When the system evaluates the evolved individuals, corresponding circuit parameters are restored by performing the above operation in reverse order (Fig. 3). Moreover, the restored parameters are quantized with a certain step depending on a variance of each parameter and the minimum step size of a target fabrication process. In our system, the parameters are quantized with step size $\sigma_j/10$, where σ_j is a variance of the jth parameter. If $\sigma_j/10$ is less than the minimum step size, the system applies the latter value. It realizes a coarse-to-fine search strategy, which achieves an effective exploration due to the prevention of the search space explosion of GA.

3 EXPERIMENT

We implement a GA-based automated analog circuit sizing system. The system applies GA to search the

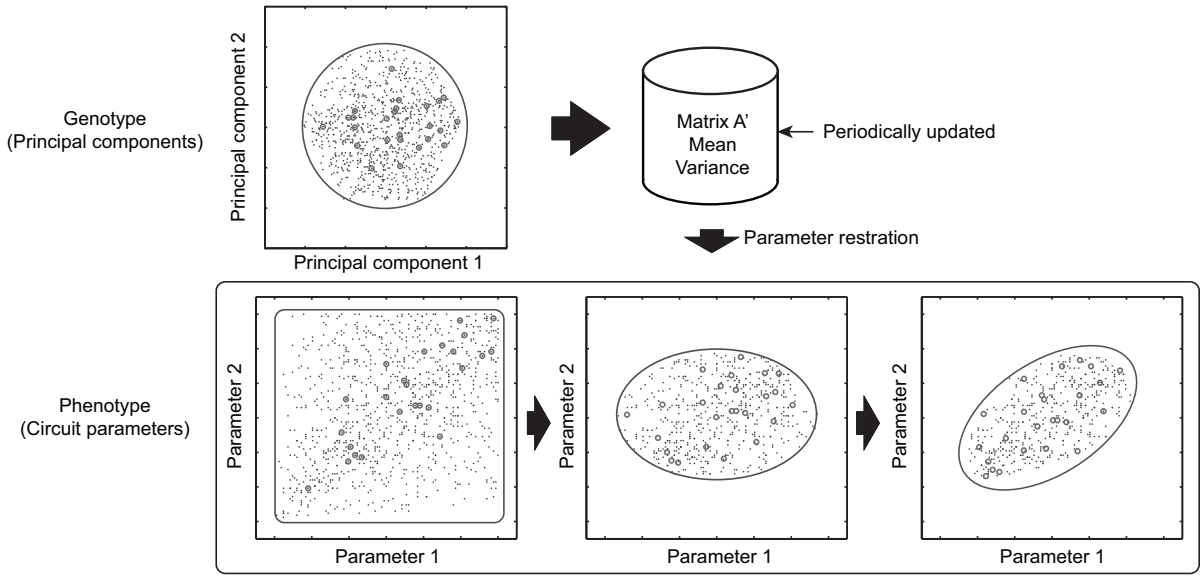


Figure 3: Parameter restoration (A' is a matrix such that $AA' = I$).

Table 1: Main parameter values for GA.

Parameter	Value
Maximum number of generations	500
Population size	200
Crossover method	BLX-0.5
Crossover rate	0.7
Mutation rate	0.1
Mutation method	Uniform
Selection method	Rank & Elitist
Interval of PCA execution	20
Num. of individuals for PCA	20

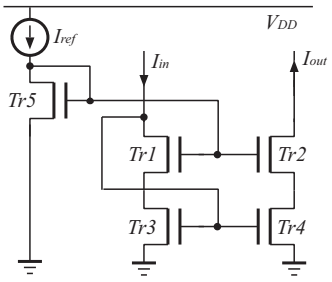


Figure 4: Wide-swing nMOS current mirror.

optimal set of circuit parameters (transistor length L and width W) which achieves higher performance on a given circuit topology, and periodically the orthogonalization procedure using PCA is performed to correct the search space of the optimization. After updating a characteristic vector matrix, the system destroys the lower-fitness individuals and creates randomly generated ones to form a population of the next generation.

The design specification considered here is a wide-swing nMOS current mirror (Fig. 4), which is widely used in analog integrated circuits both as a biasing element and as a load device for amplifier stage(Gray et al., 2001). The basic function of current mirrors is to produce an output current i_O equal to the input current i_I multiplied by a desired current gain B, that is, $i_O = -B \times i_I$. Ideally the gain B is independent of the input frequency, the output current i_O is independent of the output voltage v_O, the input voltage drop v_I equals zero, and the circuit area is as small as possible. In practice, the characteristics of real current mirrors deviate from those of ideal ones. There are various possible choices for a current mirror structure due to the trade-offs among the characteristics.

Design parameters of this circuit structure are (W_1,W_2,$\cdots$,W_5,L_1,L_2,$\cdots$,L_5), where W_i and L_i are the width and length of the MOS transistor Tr_i, respectively. The generated individual is translated into the corresponding netlist, which is simulated to analyze its electrical behavior and characteristics with SPICE simulations. Traditionally a weighted sum of the fitness functions has been used for the function aggregation. However, for nonlinear optimization problems like analog circuit sizing where the different objectives have nonlinear dependence upon each other, this approach sometimes causes some difficulties and leads to the premature convergence of the evolution process. For example, if one objective is easier to attain a target performance than other objectives, the

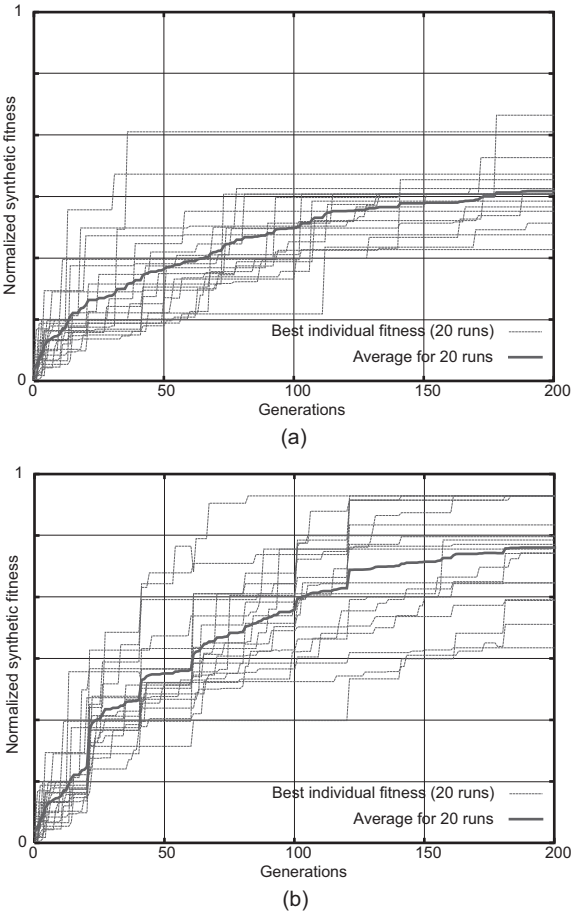


Figure 5: Fitness transition: (a) the conventional system, (b) the proposed system with PCA (The synthetic fitness values are normalized into $0 \leq F \leq 1$).

population sometimes converges to the specific search space, which leads to the evolutionary stasis. For satisfying all the objectives at the end of the optimization run, all the different fitness functions must be normalized and weighted equally.

In order to prevent the system from the premature convergnce, we define the synthetic evaluation function F as follows:

$$F = \prod_{i=1}^{3} F_i,$$

where F_1 evaluates the accuracy in i_I-i_O characteristic, F_2 evaluates the output saturation voltage investigated through DC transfer analysis, and F_3 evaluates the input impedance. The value of F_i is determined according to the rank of the ith characteristic in a population and is normalized into $0 \leq F_i \leq 1$. Thus the optimization process advances towards searching a parameter set which realizes higher performance synthetically. Note that only when the ith characteristic of an individual cannot be simulated by SPICE, F_i is set to 0 and the synthetic fitness value of the individual is also set to 0, which will be eliminated in the selection operation.

For more effecient exploration, we need further investigation of finding more proper fitness function. The result would be reported in the future.

In order to demonstrate the specific advantage of the search space reduction using PCA, we carry out experiments on generating an optimal parameter set of the given current mirror circuit structure. Table 1 summarizes the system parameters in this experiment. We set $0.6\mu m \leq W_i \leq 60\mu m$ and $0.6\mu m \leq W_i \leq 60\mu m$, and the minimum step size of these parameters is $0.1\mu m$. Thus the size of the possible choices is $(\frac{60-0.6}{0.1})^{10} \simeq 5.47 \times 10^{27}$, which needs 93 bits for an individual representation in traditional GA. By applying parameter quantizetion method described above, on the other hand, the proposed system limits the size to 10^{10} at most, which needs 34 bits, and the possible choices of circuit parameters are changed depending on the result of PCA.

Figure 5 shows the effect of PCA-based search space reduction by comparing the average fitness transitions. The system with PCA shows the tendency to improve the fitness more rapidly. It clearly shows that the proposed system explores a search space and finds better implementations more effectively compared with the conventional system.

Next, we evaluate the effect of applying PCA on the performance of evolved solutions. Figure 6 compares the performance of evolved circuits on

(a) the conventional system,

(b) the system with the search space reduction according to their means and variances independently,

(c) the system with the parameter orthogonalization procedure by PCA,

(d) the system with the parameter orthogonalization procedure and dimensionally reduction using PCA.

We performed 20 distinct evolutionary runs for each condition. We can see that the system with PCA generated higher-performance circuits under given evaluation function on average, and the dimensionally reduction method further improved the performance of evolved solutions.

In the following, we analyze one successful evolutionary process in detail. Figure 7 depicts an example snapshot of the evolved best fitness individual on each generation. The transistor sizes of them are described graphically. Given the initial random population, the system searches the set of parameters having higher fitness value for a while. At the 31st

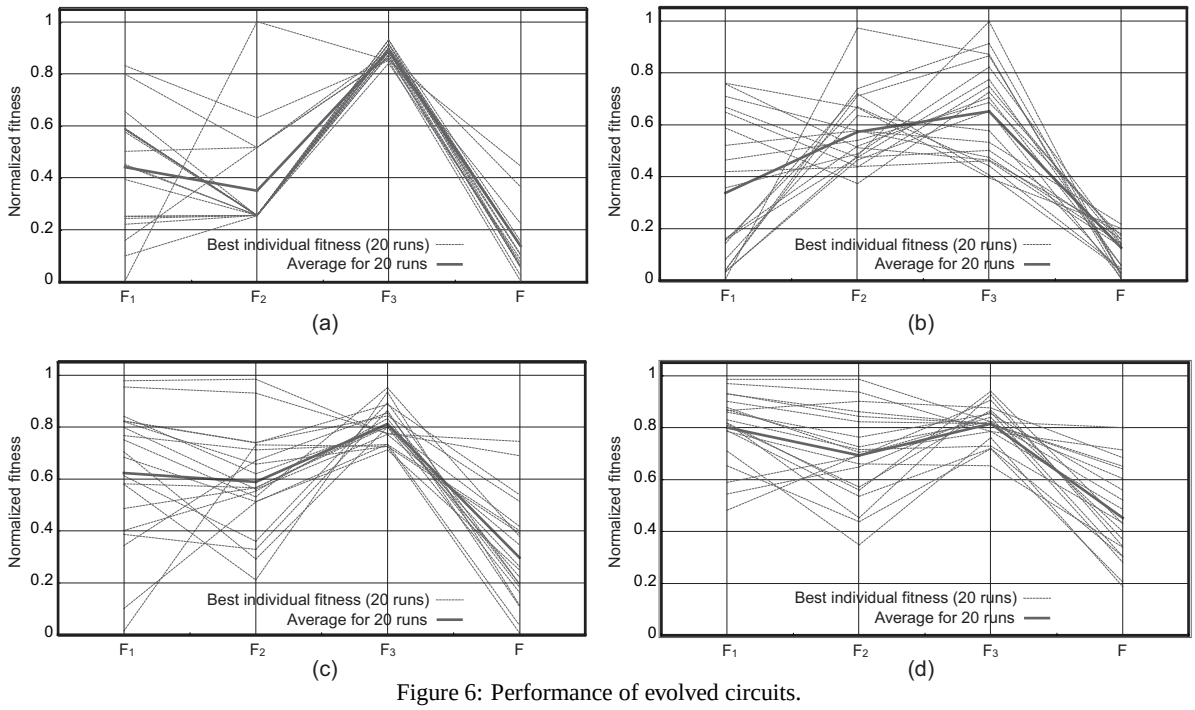


Figure 6: Performance of evolved circuits.

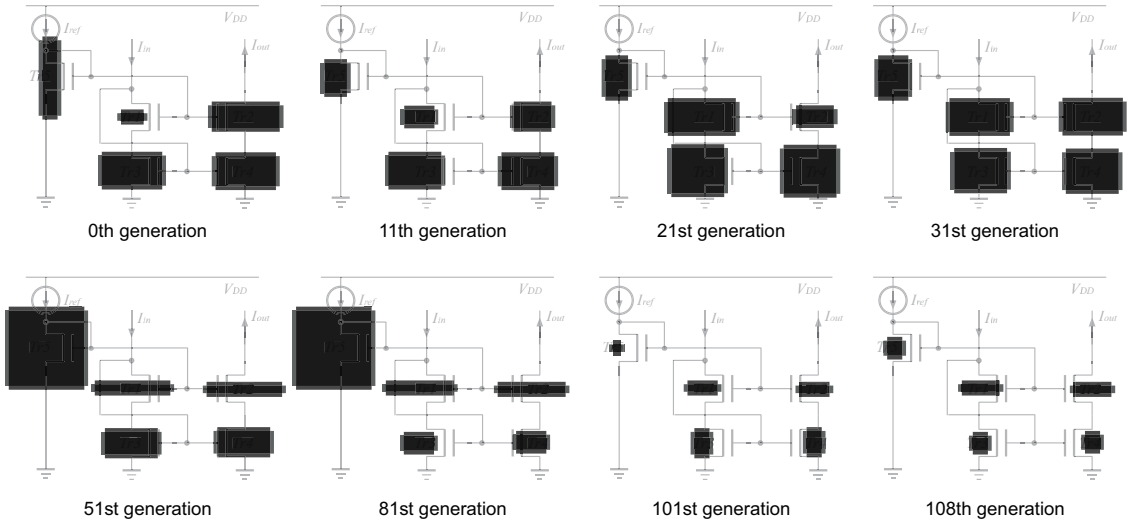


Figure 7: Best fitness individual transition.

generation, the system finds a set of parameters such that $(W1, L1, W3, L3) = (W2, L2, W4, L4)$, which is known as the proper setting of this structure. After that, the evolution process further improves the performance by changing a set of parameters. We can see that the best fitness individuals after the 31st generation have a tendency of keeping the above ratio since the parameter orthogonalization procedure is periodically performed to correct the search space. And finally, the system reaches the optimal solution at the 108th generation. This result shows that the PCA-based search space correction method works well and the system can find the optimal solutions effectively.

4 CONCLUSIONS

In this paper, we have presented a possibility of the GA-based analog circuit sizing system and its performance improvement using PCA-based orthogonalization procedure. An experimental synthesis of a wide-swing current mirror circuit demonstrates the poten-

tial capability of the proposed system to optimize analog-digital-mixed architectures without using expert knowledge of transistor-level circuits.

The experimental result also suggests a possibility of constructing intelligent circuit synthesis systems, which may evolve its ability by accumulating design experience. By introducing the guidelines for reusing suitable individuals, the proposed system could achieve more significant results.

ACKNOWLEDGEMENTS

This work was supported by Grant-in-Aid for Young Scientists (B) (No. 18700044) from The Ministry of Education,Culture,Sports,Science and Technology.

REFERENCES

Back, T., Hammel, U., and Schwefel, P. H. (1997). Evolutionary computation: Comments on the history and current state. *IEEE Trans. Evolutionary Computation*, 1(1):3 – 13.

Eshelman, L. J. and Schaffer, J. D. (1993). Real-coded genetic algorithms and interval schemata. *Foundations of genetic algorithms 2*, pages 187 – 202.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wasley Publishing Company.

Gray, P. R., Hurst, P. J., Lewis, S. H., and Meyer, R. G. (2001). *Analysis and Design of Analog Integrated Circuits (Fourth Edition)*. John Wiley & Sons.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Homma, N., Aoki, T., and Higuchi, T. (2000). Evolutionary synthesis of fast constant-coefficient multipliers. *IEICE Trans. Fundamentals*, E83-A(9):1767–1777.

Homma, N., Aoki, T., and Higuchi, T. (2002). Evolutionary graph generation system with transmigration capability and its application to arithmetic circuit synthesis. *IEE Proc. Circuits Devices Syst.*, 149(2):97 – 104.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer.

Koza, R. J., III, Bennett, H. F., Andre, D., Keane, A. M., and Dunlap, F. (1997). Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE Trans. Evolutionary Computation*, 1(2):109 – 128.

Lohn, D. J. and Colombano, S. P. (1999). A circuit representation technique for automated circuit design. *IEEE Trans. Evolutionary Computation*, 3(3):205 – 219.

Natsui, M., Homma, N., Aoki, T., and Higuchi, T. (2005). Design of multiple-valued logic circuits using graph-based evolutionary synthesis. *Journal of Multiple-Valued Logic and Soft Computing*, 11(5-6):519 – 544.

Natsui, M., Tadokoro, Y., Homma, N., Aoki, T., and Higuchi, T. (2007). Synthesis of current mirrors based on evolutionary graph generation with transmigration capability. *IEICE Electronics Express*, 4(3):88 – 93.

Shibata, H. and Fujii, N. (2001). An evolutionary synthesis of analog active circuits using current path based coding. *IEICE Trans. on Fundamentals*, E84-A(10):2561–2568.

Somani, A., Chakrabarti, P., and Patra, A. (2007). An evolutionary algorithm-based approach to automated design of analog and rf circuits using adaptive normalized cost functions. *IEEE Trans. Evolutionary Computation*, 11(3):336 – 353.

ADAPTIVE RESOURCES CONSUMPTION IN A DYNAMIC AND UNCERTAIN ENVIRONMENT

An Autonomous Rover Control Technique using Progressive Processing

Simon Le Gloannec, Abdel Illah Mouaddib
GREYC UMR 6072, Université de Caen Basse-Normandie, Campus Côte de Nacre
bd Maréchal Juin, BP5186, 14032 Caen Cedex, France
{*slegloan, mouaddib*}*@info.unicaen.fr*

François Charpillet
MAIA team, LORIA, Campus Scientifique, BP 239, 54506 Vandœuvre-lès-Nancy Cedex, France
charp@loria.fr

Keywords: MDP, Resource-bounded Reasoning, Hierarchical control, Autonomous agents, Planning and Scheduling.

Abstract: This paper address the problem of an autonomous rover that have limited consumable resources to accomplish a mission. The robot has to cope with limited resources: it must decide the resource among to spent at each mission step. The resource consumption is also uncertain. Progressive processing is a meta level reasoning model particulary adapted for this kind of mission. Previous works have shown how to obtain an optimal resource consumption policy using a Markov decision process (MDP). Here, we make the assumption that the mission can dynamically change during execution time. Therefore, the agent must adapt to the current situation, in order to save resources for the most interesting future tasks. Because of the dynamic environment, the agent cannot calculate a new optimal policy online. However, it is possible to compute an approximate value function. We will show that the robot will behave as good as if it knew the optimal policy.

1 INTRODUCTION

Resources consumption control is crucial in the autonomous rover context. For example, a rover must know when it has to go back to a docking station before having no energy left. The question for the agent is to know where, when, and the amount of resources it has to spend for a particular task. We want this agent to be adaptive : it has to choose the amount of resources to spend in each task by taking into account the expected value. This is the reason why we use progressive processing to model and to control the mission. This model of reasoning describes tasks that can be performed progressively, with several options at each step. The resources consumption is probabilistic. Progressive processing provides a meta-level resources consumption model. Then, we obtain an optimal policy control thanks to a MDP.

For now, progressive processing does not cope with dynamic environments. In a real robot application, it is not uncommon to see new tasks incoming during the mission. When it happens, the agent has to completely recalculate the resource control policy to behave optimally. Most of the time, it doesn't have sufficient time to do it. For these reasons, we propose here a value function approximation method that allows the agent to compute a local near optimal policy very quickly in order to cope with the dynamic changes. We will bring experimental result to validate this method.

The paper is divided into five main sections. The third section introduces progressive processing and its use for an autonomous rover mission application. In the fourth section, we present the dynamic environment and propose a way to cope with changes during the mission. This theory is experimentally validated in the last section.

2 RELATED WORK

- **Control of Progressive Processing:** Progressive processing has been introduced in (Mouaddib and Zilberstein, 1998), and has been used in a real rover application (Zilberstein et al., 2002) and in a retrieval information engine (Arnt et al., 2004). Concerning the retrieval information engine, users send requests to a server that can respond very

precisely and use lots of resources, or approximatively and use less resources. Progressive processing permits to model the server tasks. It also permits to adapt the resources dedicated to each task by taking the number of requests into account. We will briefly present the rover application in this paper.

- **MDP Decomposition:** an MDP is used to formalise the mission resource consumption control. We can either compute an optimal policy or an approximate policy. Our approximation is based on MDP decomposition. This technique has been largely studied in the literature. Dean et al (Dean and Lin, 1995) investigated methods that decompose global planning problems into a number of local problem. In (Parr, 1998), the method is to build a cache of policies for each part of the problem independently, and then to combine the pieces in separate steps. (Meuleau et al., 1998) also proposed a decomposition technique for a problem where tasks have indepedent utilities, and are weakly coupled.
- **Value Function Approximation:** our work also deals with value function approximation. In (Feng et al., 2004), the state space is dynamically partitioned into regions where the value function is the same throughout the region. Authors make piecewise constant and piecewise linear approximations. (Pineau et al., 2003) introduced point base value iteration. It approximates an exact value iteration solution by selecting a small set of representative belief points and by tracking the value and its derivative for those points only.

3 MODEL A MISSION WITH PROGRESSIVE PROCESSING

The problem formalism will be presented hierarchically: the mission is divided into progressive processing units (PRU), which are separated into levels.

3.1 Formalism

A mission is a set of tasks, and each task is an acyclic graph vertex (see Figure 1). To facilitate access to the formalism, we assume that this acyclic graph is an ordered sequence. This involves no loss of generality. You can see this sequence as a particular path in the graph (for example A, B, E, F). There are **P** tasks in the sequence. Each task is modelled by a progressive processing unit (PRU). It is structured hierarchically. Each $\mathsf{PRU}_\mathtt{p}$, $\mathtt{p} \in \{1,\ldots,\mathbf{P}\}$ is a level ordered finite sequence $[\mathsf{L}_{\mathtt{p},1},\ldots,\mathsf{L}_{\mathtt{p},\mathtt{L}}]$. An agent can process a level only if the preceding level is finished. The process can be interrupted after each level. It means that the agent can stop the PRU execution, but it receives no reward for it. Other situations have been proposed when the agent can have a reward at each level. However, in our case, only a complete task accomplishment provides a reward.

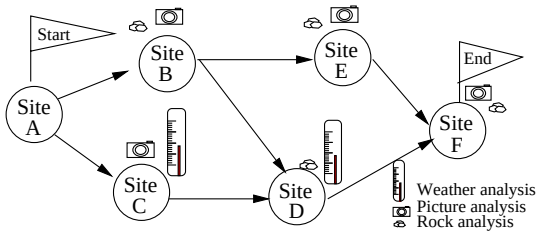


Figure 1: A mission.

Each level $\mathsf{L}_{\mathtt{p},\ell}$ contains one or more modules $[\mathsf{m}_{\mathtt{p},\ell,1},\ldots,\mathsf{m}_{\mathtt{p},\ell,\mathtt{M}}]$. A module is a specific way to execute a level. The agent can only execute one module per level. The execution of a module produces a quality **Q** and consumes resources. A progressive processing unit definition is illustrated on Figure 2 for a picture task. Firstly, the rover has to aim its camera, then it chooses the picture resolution. At the end, it saves the picture. The execution is performed from bottom to top.

The quality $\mathbf{Q}_{\mathtt{p},\ell,\mathtt{m}} \in \mathbb{R}^+$ is a criterion to measure the module execution impact. There is no immediate reward after each level processing. The agent receives the sum of all the $\mathbf{Q}_{\mathtt{p},\ell,\mathtt{m}}$ only when the last level is performed.

The resource consumption in a module $\mathsf{m}_{\mathtt{p},\ell,\mathtt{m}}$ is probabilistic. We denote as $\mathcal{P}r_{\mathtt{p},\ell,\mathtt{m}}$ the probability distribution of resources consumption.

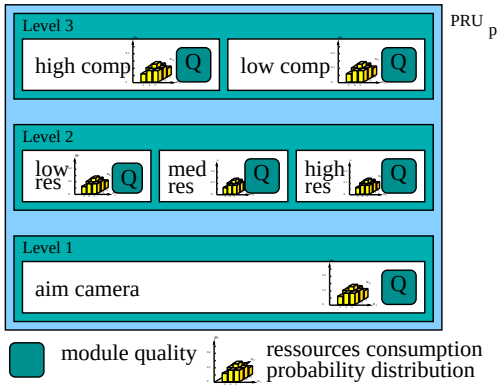


Figure 2: PRU.

3.2 Mission Control

The problem of control we address in this paper consists of a robot exploring an area where it has sites to visit and performs exploration tasks. The problem is that the robot cannot know in advance its resource consumption. We use then a MDP to control the mission. The agent is supposed to be rational. It must maximise the mathematical expected value. It computes a policy that correspond to this criterion before executing the mission. The on-line mission control process consists in following this policy. In the next section, we present the MDP model and the control policy calculation. At each site the agent must take a decision to **stay for continuing** the exploration or to **move to another site**.

3.2.1 Modelling the Mission as an Markov Decision Process

Formally, a MDP is tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}r, \mathcal{R}\}$ where :

- $\mathcal{S}$ is a finite set of states,
- $\mathcal{A}$ is a finite set of actions,
- $\mathcal{P}r$ is a mapping $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$,
- $\mathcal{R} : \mathcal{S} \rightarrow \mathbb{R}$ is a reward function.

Given a particular rational criterion, algorithms for solving MDPs can return a policy π, that maps from $\mathcal{S}$ to $\mathcal{A}$, and a real-valued function $V : \mathcal{S} \rightarrow \mathbb{R}$. Here, the criterion is to maximise the expected reward sum.

3.2.2 States

In the progressive processing model, a state is a tuple $\langle \mathtt{r}, \mathbf{Q}, \mathtt{p}, \ell \rangle$. $\mathtt{r}$ is a reel number that indicates the amount of remaining resources. $\mathbf{Q}$ is the cumulated quality since the beginning of the current PRU. $\mathtt{p}$ and ℓ indicates the last executed level $\mathsf{L}_{\mathtt{p},\ell}$ done. The failure state, denoted as $\mathbf{s}_{\mathtt{failure}}$ is reached when the resource quantity is negative. The reward is assigned to the agent only when the last level of the PRU has been successfully executed. Then, it is necessary to store the cumulated quality $\mathbf{Q}$ in the state description. We have introduced level 0 in order to represent the situation where the agent begins the PRU execution.

3.2.3 Actions

There are two kinds of actions in the progressive processing model. An agent can execute only one module in the next level or move to the next PRU. These two actions are depicted in Figure 3. Formally, $\mathcal{A} = \{\mathbf{E}_{\mathtt{m}}, \mathbf{M}\}$. When the agent reaches the last level in a $\mathsf{PRU}_{\mathtt{p}}$, it directly moves to the next $\mathsf{PRU}_{\mathtt{p}+1}$.

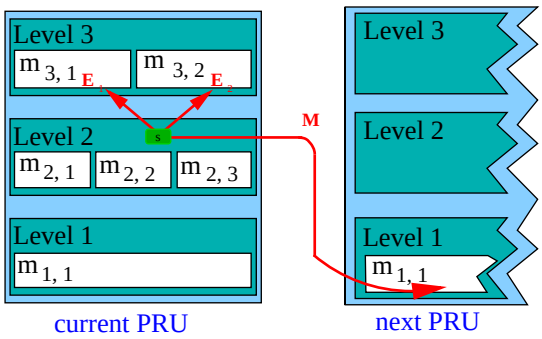


Figure 3: Actions.

Actually, $\mathbf{E}_{\mathtt{m}}$ is an improvement of the the current PRU whereas $\mathbf{M}$ is an interruption. The agent can progressively improve the PRU execution, it can also stop it at any moment.

3.2.4 Transitions

$\mathbf{M}$ is a deterministic action whereas $\mathbf{E}_{\mathtt{m}}$ is not. Indeed, the module execution consumes resources and this consumption is probabilistic. After executing a given module, the agent always cumulates a fixed quality. The uncertainty is related to the resources consumption probability distribution. Thus,

$$\mathcal{P}r(\langle \mathtt{r}, \mathbf{Q}, \mathtt{p}, \ell \rangle, \mathbf{M}, \langle \mathtt{r}, 0, \mathtt{p}+1, 0 \rangle) = 1$$

$$\mathcal{P}r(\langle \mathtt{r}, \mathtt{p}, \mathbf{Q}, \ell \rangle, \mathbf{E}_{\mathtt{m}}, \langle \mathtt{r}', \mathbf{Q}', \mathtt{p}, \ell+1 \rangle) = \mathcal{P}r(\Delta\mathtt{r} | \mathsf{m}_{\mathtt{p},\ell,\mathtt{m}})$$
$$\text{where} \quad \mathbf{Q}' = \mathbf{Q} + \mathbf{Q}_{\mathtt{p},\ell,\mathtt{m}}$$
$$\text{and} \quad \mathtt{r}' = \mathtt{r} - \Delta\mathtt{r} \tag{1}$$

3.2.5 Reward

A reward is given to the agent as soon as it finishes a PRU. This reward corresponds to the cumulated quality through the modules path. If the agent leaves a PRU without finishing it, it receives no reward. This makes a sense for exploration task where the robot has no reward if the task is not completely finished.

$$\mathcal{R}(\langle \mathtt{r}, \mathbf{Q}, \mathtt{p}, \mathsf{L}_{\mathtt{p}} \rangle) = \mathbf{Q} \tag{2}$$
$$\mathcal{R}(\langle \mathtt{r}, \mathbf{Q}, \mathtt{p}, \ell < \mathsf{L}_{\mathtt{p}} \rangle) = 0 \tag{3}$$

where $\mathsf{L}_{\mathtt{p}}$ is the number of levels in $\mathsf{PRU}_{\mathtt{p}}$.

3.2.6 Value Function

The control policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ depends on a value function that is calculated thanks to the Bellman equation 4. We assume that $V(\mathbf{s}_{\texttt{failure}}) = 0$.

$$
\begin{array}{l}
V(\langle \mathtt{r}, \mathbf{Q}, \mathtt{p}, \ell \rangle) = \\
\left\{ \begin{array}{ll} 0 & \text{if } \mathtt{r} < 0 \quad \text{(failure)} \\ \mathcal{R}(\langle \mathtt{r}, \mathbf{Q}, \mathtt{p}, \ell \rangle) + \max(V_{\mathbf{M}}, V_{\mathbf{E}}) & \text{otherwise} \end{array} \right. \\
\\
V_{\mathbf{M}}(\langle \mathtt{r}, \mathbf{Q}, \mathtt{p}, \ell \rangle) = \\
\\
\left\{ \begin{array}{ll} 0 & \text{if } \mathtt{p} = \mathbf{P} \\ V(\langle \mathtt{r}, \mathbf{0}, \mathtt{p}+1, 0 \rangle) & \text{otherwise} \end{array} \right. \\
\\
V_{\mathbf{E}}(\langle \mathtt{r}, \mathbf{Q}, \mathtt{p}, \ell \rangle) = \\
\\
\left\{ \begin{array}{l} 0 \text{ if } \ell = \mathtt{L_p} \\ \max\limits_{\mathbf{E}_{\mathtt{m}}} \sum\limits_{\Delta \mathtt{r}} \mathcal{P}\, r(\Delta \mathtt{r} | \mathtt{m}_{\mathtt{p},\ell,\mathtt{m}}).V(\langle \mathtt{r}', \mathbf{Q}', \mathtt{p}, \ell+1 \rangle) \\ \text{where} \quad \mathbf{Q}' = \mathbf{Q} + \mathbf{Q}_{\mathtt{p},\ell,\mathtt{m}} \\ \text{and} \quad \mathtt{r}' = \mathtt{r} - \Delta \mathtt{r} \end{array} \right.
\end{array}
\tag{4}
$$

Action **E** consumes some resources whereas **M** does not. The agent will execute the module that gives it the best expected value.

3.3 Control Policy

The main objective is to compute a policy that the robot will follow in order to optimise the resource consumption during the mission. It has been proven that we could calculate an optimal policy (Mouaddib and Zilberstein, 1998). When the mission is fixed before execution time, we can get this policy with a backward chaining algorithm. With this policy, the robot can choose at any moment the decision that maximise the global utility function for the mission i.e. the obtained reward sum. But, this policy is fixed for a given mission. If the mission changes during execution time, this policy is no longer up to date.

Even if the policy algorithm is linear in the state space, the time needed to compute the optimal policy is often more than a module execution time. This occurs especially when the mission (and then the state space) is large. This is the reason why we propose an other way to calculate the policy. We will not calculate an optimal policy, but a near-optimal policy which could be used to take good decision when the mission changes.

4 THE DYNAMIC ENVIRONMENT

We suppose in this paper that tasks can come or disappear during execution time. This assumption is realistic for an explorer rover.

Instead of seeing the control as a global policy for the mission, we decompose it in two parts, the current PRU and the rest of the mission. The policy is only calculated for the current PRU. It only dependson the expected value function for the rest of the mission.

We could yet obtain the optimal policy with this method by simply calculating the expected value function for the rest of the mission without any approximation. Thus, we will naturally not save time. Then, we present here a method to calculate the expected value function very quickly.

5 VALUE FUNCTION APPROXIMATION

The quick value function approximation is based on a decomposition technique. Decomposition techniques are often use in large Markov decision processes. Here, we will re-compose an approximate expected value function in two times. Firstly, we calculate an optimal local value function for each PRU in the mission. These local functions are splitted into function pieces. Secondly, we re-compose the value function with all the function pieces as soon as a change occurs at run-time. This section is divided in two parts, the decomposition, and the recomposition. We denote as V^* the optimal value function and $V^{\sim}$ its approximation. The main objective of the recomposition is to fit V^* as good as possible.

5.1 Decomposition

The decomposition consists in the calculation of the value functions for each $\mathsf{PRU}_\mathtt{p}$ in the mission. These functions are indeed performance profiles (denoted as $f_\mathtt{p}$): they indicate the expected value for this PRU if $\mathtt{r}$ are allocated. Three performance profiles examples are depicted in Figure 4. The PRU γ can not consume more than 31 resource units, and give a maximum expected value of 25.

The main idea is to use these functions to recompose the global approximate value function. The performance profiles receive a preliminary treatment. The progressive processing provides value functions that are similar to anytime reasoning value functions :

- they increase with the allocated resource amount,

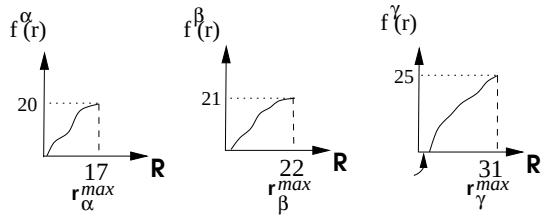


Figure 4: 3 PRU performance profiles.

- the best improvements are made with few resources, the growth of V^* is higher at its beginning than at its end.

For these reasons we split f_p into i_p pieces $g_{p,1},\ldots,g_{p,i_p}$. During the recomposition, we will take the best pieces to re-compose the beginning of $V^{\sim}$, and the worst for the end. We keep the part of f_p with the best growth, i.e. the piece between $(0,0)$ and $(r_{p,1}^{max}, f_p(r_{p,1}^{max}))$. $r_{p,1}^{max}$ is the resource amount that provides the best tradeoff between the resources spent and the local expected value.

$$r_{p,1}^{max} = \underset{p,r}{\operatorname{argmax}}\left(\frac{f_p(r)}{r}\right) \quad (5)$$

$$\forall i>1, r_{p,i}^{max} = \underset{p,r}{\operatorname{argmax}}\left(\frac{f_{p,i}(r)}{r}\right) \quad (6)$$

When the first $r_{p,1}^{max}$ is found, we save the first piece of f_p as $g_{p,1}$. We split f_p in $g_{p,1}$ and $f_{p,2}$. We continue to search the second $r_{p,2}^{max}$ in $f_{p,2}$; we save $g_{p,2}$ etc... until f_{p,i_p} is empty.

$$g_{p,1}: \left| \begin{array}{l} [0\ldots r_{p,1}^{max}] \rightarrow \mathbb{R} \\ r \rightarrow f_p(r) \end{array} \right.$$

$$f_{p,1}: \left| \begin{array}{l} [0\ldots r_p^{max} - r_{p,1}^{max}] \rightarrow \mathbb{R} \\ r \rightarrow f_p(r - r_{p,1}^{max}) - f_p(r_{p,1}^{max}) \end{array} \right.$$

$$\forall i>1,\ g_{p,i}: \left| \begin{array}{l} [0\ldots r_{p,i}^{max}] \rightarrow \mathbb{R} \\ r \rightarrow f_{p,i}(r) \end{array} \right.$$

$$\forall i \geq 1,\ f_{p,i+1}: \left| \begin{array}{l} \left[0\ldots r_p^{max} - \sum_{j=1}^{i} r_{p,j}^{max}\right] \rightarrow \mathbb{R} \\ r \rightarrow f_{p,i}(r - r_{p,i}^{max}) - f_{p,i}(r_{p,i}^{max}) \end{array} \right. \quad (7)$$

Figure 5 illustrates the decomposition method for one PRU performance profile f_p.

All PRU are treated in the same way, such that we obtain a list of performance profile pieces $\{g_{p,i}, 1 \leq p \leq \mathbf{P}, 1 \leq i \leq i_p\}$. We sort this set with the best tradeoff criterion : $g_{p,i}(r_{p,i}^{max})/r_{p,i}^{max}$. When this set is sorted, the agent is ready to re-compose $V^{\sim}$.

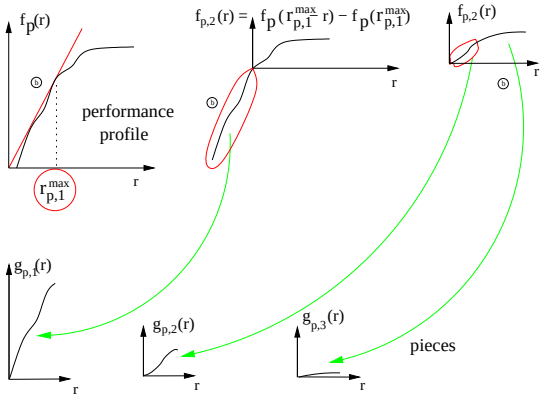


Figure 5: Decomposition.

5.2 Recomposition

When the mission changes during execution time, the agent re-composes $V^{\sim}$ by assembling all the pieces. The first g element is the one with the best growth, and so on. The method is depicted on Figure 6. $V^{\sim}$ and V^* have the same support $[0\ldots r^{max}]$. In order to know the expected value for a given amount r, the agent check the value $V^{\sim}(r)$. Then, it will take its local decision according to this approximate value.

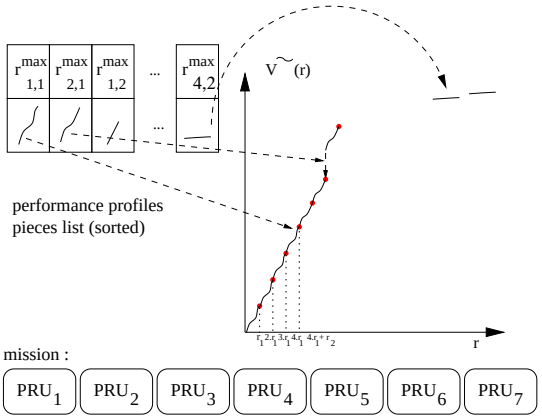


Figure 6: $V^{\sim}$ recomposition.

6 VALIDATION

In the previous sections, we present how to obtain an approximation of the value function. The rover will use this function in order to take decision if the mission change during execution time. Now, we must prove that :

- the time needed to obtain $V^{\sim}$ is short enough,

- the rover can take good decisions by using $V^{\sim}$.

Concerning the time needed to calculate $V^{\sim}$, we obtain very good results : it takes just few milliseconds. Tabular 1 give computation time for different mission sizes.

Table 1.

# PRUs	20	40	60	80	100	120	140
V^* (in s)	3	7	16	35	63	112	187
$V^{\sim}$ (in ms)	0,4	0,9	1,2	1,7	2,2	2,6	3,5

To validate our approach, we will show that the agent can take good decisions by using $V^{\sim}$ as an expected value. The decision quality evaluation is made throw the Q-value function comparison for each pair $(state, action)$ between the policies locally obtained with $V^{\sim}$ and V^*.

The validation consists in four steps :

1. we calculate both $V^{\sim}$ and V^*,
2. we calculate both policies (π_0^* and $\pi_0^{\sim}$) for the current PRU_0,
3. we calculate Q_0^*, the local optimal Q-value function (for PRU_0),
4. compare the loss of value while the agent is using $\pi_0^{\sim}$ instead of π_0^*.

Note that point 2 and 3 are done together at the same time. As the MDP for PRU_0 has few states, the calculation of π_0^*, Q_0^* and $\pi_0^{\sim}$ is quick. We did some experiments with mission composed of different PRU kinds and number. Figure 7 is an example of $V^{\sim}$ and V^* for 20 PRU with 4 PRU kinds.

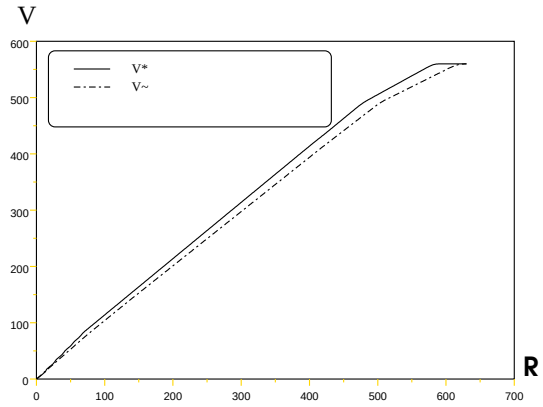


Figure 7: $V^{\sim}$ and V^*.

On this Figure 7 we see that $V^{\sim}$ is a good approximation of V^*. We have done several experiments and have obtain some good results. Q_0^* is given by the following equation.

$$Q_0^*(\langle \mathtt{r}, \mathbf{Q}, 0, \ell \rangle, \mathbf{E}_{\mathtt{m}}) = V_{\mathbf{E}_{\mathtt{m}}}(\langle \mathtt{r}, \mathbf{Q}, 0, \ell \rangle) \quad (8)$$

$$Q_0^*(\langle \mathtt{r}, \mathbf{Q}, 0, \ell \rangle, \mathbf{M}) = V^*(\mathtt{r}) \quad (9)$$

It makes no sense to calculate the Q-value for $\pi_0^{\sim}$. When the agent takes its decisions using $\pi_0^{\sim}$ instead of π_0^*, there may be a loss of value. The error value is given by :

$$e(\mathbf{s}) = Q_0^*(\mathbf{s}, \pi_0^*(\mathbf{s})) - Q_0^*(\mathbf{s}, \pi_0^{\sim}(\mathbf{s})) \quad (10)$$

Indeed, if for a given state $\mathbf{s}$, $\pi_0^*(\mathbf{s}) = \pi_0^{\sim}(\mathbf{s})$, then the error is zero. Otherwise, $e(\mathbf{s})$ measure the loss of value. The objective is to obtain as few error as possible. When an error exists, the better is to have a small value.

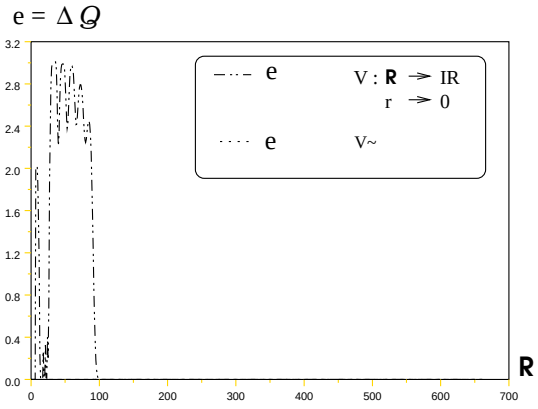


Figure 8: $V^{\sim}$ and V^*.

Figure 8 is an error measure with $V^{\sim}$ and V^* depicted on Figure 7. When we use our approximation method, there is no error. It explains why we do not see any dotted curve on the Figure. The other curve corresponds is the error measure if the agent considers that the expected value is null ($V(\mathtt{r}) = 0$). When the robot has few resources, it does not always take an optimal decision. However, when the agent follows our approximative policy, it never make mistakes.

Of course, this example seems to be chosen because it is the best. But it is not the case. We made some experiments with two different PRU kinds. The first PRU kind is composed of tightened modules : near qualities, near resource consumption. On the contrary, the second PRU kind is composed of distant modules : separate qualities, distant resource consumption.

When the agent calculates $\pi_0^{\sim}$ on a PRU of the first type, there are lots of errors. Indeed, the modules are almost the same, so the Q-value difference between two actions is low. Then, it is very difficult for the

agent to take the right decision. However, in this case, the error is also low. When the agent chooses a module that is near of the optimal one, it does not make the right decision, but this decision is good.

When the agent calculates $\pi_0^{\sim}$ on a PRU of the second type, there are few errors. For a given state, Q-values are separate because the module qualities are separate. Then, most of the time $\pi_0^{\sim}$ is equal to π_0^*, the robot chooses the right decision.

To conclude, if modules are clearly separate, the robot takes the right decision. If modules are near, the robot takes good decisions, with a low Q-value error level.

6.1 Limits of this Approximation Method

The resource consumption probability distribution follows a normal distribution law in all the modules we use for our experiments. Most of the time, it represents modules that can be found a real application. We have also tried to make some experiments with modules in which the resource consumption probability distribution is not normal. We made a risky module $\mathtt{m}$: the resource consumption is 4 or 16, with $\mathcal{P}r(4|\mathtt{m}) = 0.5$ and $\mathcal{P}r(16|\mathtt{m}) = 0.5$. In this case the module can only consume 4 or 16 units, but not 8. In this kind of particular case V^* is not smooth. As a result, $V^{\sim}$ is not a good approximation. Then, $\pi_0^{\sim}$ and π_0^* are different, there is a lot of error. But this case does not represent a realistic resource consumption module.

7 CONCLUSIONS

Resource consumption is crucial for an autonomous rover. Here, this rover has to cope with limited resources to executed a mission composed of hierarchical tasks. These tasks are progressive processing units (PRU). It is possible to compute an optimal resource control for the entire mission by modelling it with an MDP. In the case where the mission changes at execution time, the rover has to recompute online a new global policy. We propose a way to quickly compute an approximate value function that can be used to calculate a local policy on the current PRU. MDP Decomposition and value function approximation techniques are used to calculate $V^{\sim}$. We have shown in the last section that the agent takes good decisions when it uses $V^{\sim}$ to compute its local policy $\pi_0^{\sim}$. In a near future, we intend to complete our demonstration on real robots by considering dynamic situations where missions can change online.

REFERENCES

Arnt, A., Zilberstein, S., Allan, J., and Mouaddib, A. (2004). Dynamic composition of information retrieval techniques. *Journal of Intelligent Information Systems*, 23(1):67–97.

Dean, T. and Lin, S. H. (1995). Decomposition techniques for planning in stochastic domains. In *proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1121–1127, Montreal, Quebec, Canada.

Feng, Z., Dearden, R., Meuleau, N., and Washington, R. (2004). Dynamic programming for structured continuous markov decision problems. In *proceedings of UAI 2004*, pages 154–161.

Meuleau, N., Hauskrecht, M., Kim, K., Peshkin, L., Kealbling, L., Dean, T., and Boutilier, C. (1998). Solving very large weakly coupled markov decision processes. In *proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Madison, WI*.

Mouaddib, A. I. and Zilberstein, S. (1998). Optimal scheduling for dynamic progressive processing. In *proceedings of ECAI*, pages 499–503.

Parr, R. (1998). Flexible decomposition algorithms for weakly coupled markov decision process. In *proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Madison, WI*.

Pineau, J., Gordon, G. J., and Thrun, S. (2003). Point-based value iteration: An anytime algorithm for pomdps. In *proceedings of the 18th International Joint Conference on Artificial Intelligence IJCAI*, pages 1025–1032.

Zilberstein, S., Washington, R., Berstein, D., and Mouaddib, A. (2002). Decision-theoretic control of planetary rovers. *LNAI*, 2466(1):270–289.

CONTROLLING INVESTMENT PROPORTION IN CYCLIC CHANGING ENVIRONMENTS

J.-Emeterio Navarro-Barrientos
Department of Computer Science, Humboldt-Universität zu Berlin, Unter den Linden 6 10099, Berlin, Germany
navarro@informatik.hu-berlin.de

Keywords: Genetic algorithms, investment strategies, pattern recognition.

Abstract: In this paper, we present an investment strategy to control investment proportions for environments with cyclic changing returns on investment. For this, we consider an investment model where the agent decides at every time step the proportion of wealth to invest in a risky asset, keeping the rest of the budget in a risk-free asset. Every investment is evaluated in the market modeled by stylized returns on investment (RoI). For comparison reasons, we present two reference strategies which represent agents with zero-knowledge and complete-knowledge of the dynamics of the RoI, and we consider an investment strategy based on technical analysis. To account for the performance of the strategies, we perform some computer experiments to calculate the average budget that can be obtained over a certain number of time steps. To assure for fair comparisons, we first tune the parameters of each strategy. Afterwards, we compare their performance for RoIs with fixed periodicity (stationary scenario) and for RoIs with changing periodicities (non-stationary scenario).

1 INTRODUCTION

Finding a proper method to control investment proportion is a problem that has been addressed by many researchers (Kelly, 1956; Kahneman and Tversky, 1979). Many of the proposed methods are based on *machine learning (ML)*. For example, in (Magdon-Ismail et al., 2001) the authors use neural networks to find patterns in financial time series and in (Geibel and Wysotzki, 2005), the authors propose a risk-sensitive reinforcement learning algorithm to find a policy for controlling under constraints. Other techniques from ML that are also frequently used are those based on *evolutionary computation*, like *genetic programming (GP)* and *genetic algorithms (GA)*. For a general introduction to these techniques for portfolio management and bankruptcy prediction see (Dawid, 1999). Some researchers have shown that investment strategies based on GP techniques may be profitable; however, they usually find strategies which can't be easily funded (Schulenburg and Ross, 2001). Controlling strategies that are based on a standard GA may be also difficult to explain, however, we believe that they are easier to understand than those using GP.

Genetic algorithms (GA) are stochastic search algorithms based on evolution that explore progressively from a large number of possible solutions finding after some generations the best solution for the problem. Inspired by natural selection, these powerful techniques are based on some defined evolution operators, like selection, crossover and mutation (Holland, 1975). Moreover, some researches have extended the use of GA for solving stochastic dynamic optimization problems online (Grefenstette, 1992), where most of the algorithms for changing environments are tested in problems like the knapsack problem (Yang, 2005). However, to our knowledge, no-one has applied GAs specifically to the problem of controlling the proportion of investment in environments with cyclic changing returns on investment.

This paper is organized as follows: Sec. 2 describes the investment model and Sec. 3 presents a novel approach to control investment proportions based on a GA for environments with cyclic changing time series. In Sec. 4 we present the dynamics for the risky asset and we compare the performance of the adaptive strategy with other strategies for stationary and non-stationary environments.

2 INVESTMENT MODEL

We consider an investment model (Navarro and Schweitzer, 2003) where an agent is characterized by

two individual variables: (i) its *budget* $x(t)$, and (ii) its *investment proportion* $q(t)$. The budget, $x(t)$, changes in the course of time t by means of the following dynamic:

$$x(t+1) = x(t)\left[1 + r(t)\,q(t)\right] \qquad (1)$$

This means that the agent at time t invests a portion $q(t)x(t)$ of its total budget. And this investment yields a gain or loss on the market, expressed by $r(t)$, the return on investment, *RoI*. Some authors assume that returns are obtained by means of continuous double auction mechanisms (LeBaron, 2001), however, in this paper we consider that the returns are not being influenced by agent's actions, this approach plays a role in more physics-inspired investment models, (Richmond, 2001; Navarro-Barrientos et al., 2008). Since $q(t)$ always represents a portion of the total budget $x(t)$, and it is bound to $q(t) \in [0,1]$. For completeness, we assume that the minimal and maximal investment proportions are described by $q_{\min}$ and $q_{\max}$, respectively.

Thus, in this paper we present an adaptive strategy to control proportions of investment, expressed by a method to find the most proper $q(t)$. We assume a simple dynamic for the returns allowing us to focus in the feedback of these market returns on the investment strategy (and not on the feedback of the strategies on the market). Moreover, we assume that the agent invests independently in the market, i.e. there is no direct interaction with other agents.

3 ADAPTIVE INVESTMENT STRATEGY

In this section, we present an adaptive investment strategy based on a GA for controlling proportions of investment in cyclic changing environments. For simplicity, we call this strategy *Genetic Algorithm for Changing Environments* (GACE), and we show on the following the specifications for the GA.

3.1 Encoding Scheme

We consider a population of chromosomes $j = 1,...,C$, where each chromosome j has an array of genes, g_{jk}, where $k = 0,...,G_j - 1$, and G_j is the length of the chromosome j. The length of a chromosome is assumed to be in the range $G_j \in (1, G_{\max})$, where $G_{\max}$ is a parameter that specifies the maximal allowed number of genes in a chromosome. The values of the genes could be binary, but for programming reasons we use real values (Michalewiçz, 1999). Moreover, each chromosome j represents a *set of possible strategies* of an agent, where each g_{jk} is an investment proportion.

3.2 Fitness Evaluation

Each chromosome j is evaluated after a given number of time steps by a *fitness function* $f_j(\tau)$ defined as follows:

$$f_j(\tau) = \sum_{k=0}^{G_j-1} r(t)\,g_{jk}\,; \quad k \equiv t \bmod G_j, \qquad (2)$$

where τ is a further time scale in terms of generations. When a generation is completed, the chromosomes' population is replaced by a new population of better fitting chromosomes with the same population size C. Since the fitness of a chromosome tends to be maximized, negative $r(t)$ should lead to small values of g_{jk}, and positive $r(t)$ should lead to larger values of g_{jk}. Because of this, we consider the product of $r(t)g_{jk}$ as a performance measure, which is in accordance with our investment model, Eq. (1). Noteworthy, in this approach the GA tries to find the chromosomes leading to larger profits. A different approach would be to implement a GA to find the chromosomes that minimize the loss, in which case, we would have a different fitness function. Note that we treat directly returns on investment and not price movements, because our goal is to evaluate the fitness of the strategies for the RoI and not the accuracy of the prediction of the next RoI.

3.3 Selection of a New Population

If we assume that chromosomes have fixed length, $G_j = G_{\max}$, then the most proper number of time steps that have to elapse in order to evaluate all chromosomes' genes is $t_{\text{eval}} = G_{\max}$. However, this previous assumption corresponds to the ideal case where the agent knows the periodicity of the returns and sets the length of all chromosomes to this value. In this paper, we assume that the agent **doesn't know** neither the **periodicity** nor the **dynamics** of the RoI. Thus, we assume that the chromosomes have different length. Different approaches may be proposed to know after how many time steps a new generation of chromosomes should be obtained, however, we find that the best approach was to choose the number of time steps for evaluation accordingly to the length of the best chromosome in the population.

3.3.1 Elitist and Tournament Selection

After calculating the fitness of each chromosome according to Eq. (2), we first find the best chromosomes

from the current population by applying elitist selection, which copies directly the best s percentage to the new population. Afterwards, a tournament selection of size two is done by randomly choosing two pairs of chromosomes from the current population and then selecting from each pair the one with the higher fitness. These two chromosomes are not simply transferred to the new population, but undergo a transformation based on the genetic operators' crossover and mutation.

3.3.2 Crossover and Mutation Operators

The limitations of conventional crossover in GA with variable length has already been addressed by some authors, where neural networks or hierarchical tree-structures are used to determine which genes should be exchanged between the chromosomes (Harvey, 1992). For simplicity, we propose a modification of the standard GA crossover operator that better suits our demands. Thus, we propose the use of a crossover operator called *Proportional Exchange Crossover* (PEC) operator, which randomly selects the range of genetic information to be exchanged between two chromosomes and contracts(extends) the genetic information from the largest(shortest) to the shortest(largest) chromosome, respectively. Algorithm 1 shows the PEC algorithm for all pair of parent-chromosomes being selected via tournament selection. Note that a chromosome j is saved in an array with indexes in the range $[0,..,G_j-1]$. The shortest and largest parent-chromosomes are denoted by pa_s with size G_s and pa_l with size G_l, resp., and $R \in \mathbb{N}$ is the size proportion between these two parent-chromosomes. The cross-points for the shortest and largest parent-chromosomes are denoted by $cp_s \in \mathbb{N}$ and $cp_l \in \mathbb{N}$, respectively. The breeding between the two parent-chromosomes results in a short and a large children-chromosomes denoted by ch_s and ch_l, respectively.

Note, that different functions could be considered for the transformation of the genetic material between chromosomes with different length. For simplicity, we consider in our implementation of the Algorithm 1 in line 9 the function $extend(pa,m,R) = pa[m]$, which simple copies the genes from the short parent-chromosome to the large child-chromosome; and in line 13 the function $contract(pa,m,R) = 1/R \sum_{i=m}^{m+R} pa[i]$, which performs an average over the genetic material.

Now, to make sure that a population with chromosomes of diverse lengths is present, we introduce a mutation operator for the length of the chromosome. For this, with probability p_l a new length is drawn randomly and the genetic information of the chromosome is proportionally scaled to the new length, leading to a new enlarged or stretched chromosome. The algorithm used for the mutation of the length of the chromosome is based on the same principle as the PEC operator. After the crossover and length-mutation operators are applied, the typical gene-mutation operator is applied. This means that with a given mutation probability $p_m \in U(0,1)$, a gene is to be mutated by replacing its value by a random number from a uniform distribution $U(q_{\min}, q_{\max})$.

Algorithm 1: Proportional Exchange Crossover (PEC) operator.

```
1  foreach pair of parent-chromosomes do
2      create ch_s and ch_l with sizes G_s and G_l
3      find the cross-points:
       cp_s ~ U(0, G_s − 1);   cp_l = (G_l cp_s) / G_s
4      determine size proportion: R = cp_l / cp_s
5      with p = 0.5 choose side for crossover
6      if crossover on the left side then
7          extend genes from pa_s to ch_l:
           for m = 0 to cp_s − 1 do
8              for n = 0 to R − 1 do
9                  ch_l[m · R + n] ←
                   extend(pa_s, m, R)
10             end
11         end
12         contract genes from pa_l to ch_s:
           foreach m = 0 to cp_s − 1 do
13             ch_s[m] ← contract(pa_l, m, R)
14         end
15     else
16         extend as in 9 but for m = cp_s to G_s − 1
17         contract as in 13 but for m = cp_s to
           G_s − 1
18     end
19     copy remaining genes in pa_s and pa_l into
       same positions in ch_s and ch_l, respectively.
20 end
```

3.3.3 Strategy Selection and Initialization

For every new generation, the agent takes the set of strategies g_{jk} from the chromosome j with the largest fitness in the previous generation.

$$q(t) = g_{lk} \text{ with } l = \arg \max_{j=1,..,C} f_j;\ k \equiv t \bmod G_l \quad (3)$$

For the initialization, each g_{jk} is assigned a random value drawn from a Uniform distribution: $g_{jk} \sim U(q_{\min}, q_{\max})$. And for the length of the chromosomes, each G_j is initialized randomly from a Uniform distribution of integers: $G_j \sim (1, G_{\max})$, where $G_{\max}$ is the maximal allowed chromosome length.

4 EXPERIMENTAL RESULTS

In this section, we present the environment for the agent and we analyze the performance of the adaptive strategy presented above.

4.1 Artificial Returns

We consider artificially generated returns which are driven by the following dynamics:

$$r(t) = (1-\sigma)\sin\left(\frac{2\pi}{T}t\right) + \sigma\xi, \qquad (4)$$

where the amplitude of the returns depends on the amplitude noise level $\sigma \in (0,1)$, and ξ corresponds to a random number drawn from a Uniform distribution, $\xi \in U(-1,1)$. The periodicity of the returns is drawn randomly $T \sim U(1, T_{\max})$ and would be present for a number of $t' \sim U(1, t_{\max})$ time steps. Thus, σ accounts for the fluctuations in the market dynamics on the amplitude of the RoI; $T_{\max}$ accounts for the largest possible periodicity and $t_{\max}$ accounts for the maximal number of time steps a periodicity can elapse. Fig

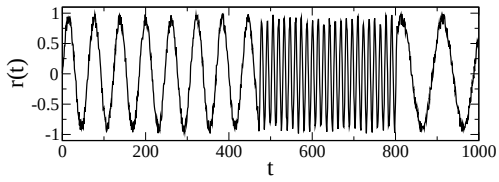


Figure 1: Periodic RoI, $r(t)$, Eq. (4) for noise level $\sigma = 0.1$, $T_{\max} = 100$ and $t_{\max} = 1000$.

4.2 Reference Strategies

For comparison purposes, we present in this section different strategies which are used as a reference to account for the performance of the adaptive strategy. Note that we could have considered other type of strategies which may lead to a more complete study. However, our main goal is to show the performance of GACE comparing it against the performance of other strategies for the same investment scenario.

4.2.1 Strategies with Zero/Complete Knowledge

For comparison reasons, we present in this section two strategies representing two simple behaviors for an agent: the first one, called *Constant-Investment-Proportion* (CP), represents the agent with zero knowledge and zero-intelligence; the second one, called *Square-Wave* (SW), represents the agent with complete knowledge of the environment.

Constant Investment Proportion (CP). The simplest strategy for an agent would be to take a constant investment proportion for every time step:

$$q(t) = q_{\min} = \text{const.} \qquad (5)$$

Square Wave Strategy (SW). An agent using this strategy invests $q_{\max}$ during the positive cycle of the periodic return and invests $q_{\min}$ otherwise:

$$q(t) = \begin{cases} q_{\max} & t \bmod T < T/2 \\ q_{\min} & \text{otherwise.} \end{cases} \qquad (6)$$

Notice that this reference strategy assumes that the agent **knows** in advance the **periodicity**, T, and the **dynamics** of the returns.

4.2.2 Strategy based on Technical Analysis

We include in our study a strategy based on *technical analysis* methods, which are frequently used by traders to forecast returns. For simplicity, we chose the *Moving Least Squares* (MLS) technique and considered an agent with a memory size M to store previous received returns. This strategy fits a linear trendline to the previous M returns, to estimate the next return, $\hat{r}(t)$. Noteworthy, once the next return has been estimated, the agent still needs to perform the corresponding adjustment of the investment proportion. For this, we consider that the agent has a *risk-neutral* behavior as follows:

$$q(t) = \begin{cases} q_{\min} & \hat{r}(t) \le q_{\min} \\ \hat{r}(t) & q_{\min} < \hat{r}(t) < q_{\max} \\ q_{\max} & \hat{r}(t) \ge q_{\max} \end{cases} \qquad (7)$$

4.3 Results for RoI with Fixed Periodicity

To elucidate the performance of the adaptive strategy proposed in this paper and the reference strategies previously presented, we start with a simple scenario where returns have a fixed periodicity.

First, we assume that the parameters of a strategy lead to an optimal performance, if it leads to the *maximum total budget* that can be reached with this strategy during a complete period of the returns. When evaluating the strategies, we have to consider that their performance is also influenced by stochastic effects. In the case of the strategy GACE we also have to account for the different possible strategies that may evolve. This means that we have to average the simulation over a large number of trials, N, where each trial simulates an agent acting independently with the same set of parameter values. For convenience, we reinitialize the budget after each cycle of

the RoI. This is done, because if the strategy performs well, the budget of the agent may reach very high values, which would lead to numerical overflows.

4.3.1 GACE Parameter Tuning

The configuration of most meta-heuristic algorithms requires both complex experimental designs and high computational efforts. Thus, for finding the best parameters for the GA, a software called *+CARPS (Multiagent System for Configuring Algorithms in Real Problem Solving)* (Monett, 2004) was used. It consists of autonomous, distributed, cooperative agents that search for solutions to a configuration problem, thereby fine-tuning the meta-heuristic's parameters.

The *GA* was configured for periodic returns with $T = 100$ and different level of noise: $\sigma = 0.1$, and $\sigma = 0.5$. Four GA parameters were optimized: the population size C, the crossover probability p_c, the mutation probability p_m, and the elitism size s. Their intervals of definition were set as follows: $C \in \{50, 100, 200, 500, 1000\}$, $p_c \in [0.0, 1.0]$, $p_m \in [0.0, 1.0]$, and $s \in [0.0, 0.5]$. We show in Table 1 the best obtained configuration for the GA in the periodic returns previously mentioned. For clarity, we considered in these experiments chromosomes with fixed length, i.e. $G_j = T$, and no probability of length mutation, i.e $p_l = 0$.

Table 1: GACE's best pars. for RoI with fixed T.

C	p_c	p_m	s
1000	0.7	0.01	0.3

Now, to better illustrate the set of investment strategies that are being obtained using GACE, we show in Fig. 2 the RoI and the investment proportions obtained after a number of time steps. For the reader with background in signal processing techniques, Fig. 2 may sound familiar as it resembles to those figures obtained when using matched filters for signal recovery (Turing, 1960).

4.3.2 Performance Comparison

In order to assure fair comparison between the strategies, we need to find the most proper parameter values for the strategies. Note that for both strategies CP, Eq. (5) and SW, Eq. (6), we don't need to tune any parameters. However, for the strategy MLS, Eq. (7), we assume that the agent **knows** the **periodicity** T of the returns. This means that the agent needs to determine the most proper memory size, M, based on the known periodicity of the returns. For this, we performed some computer experiments using MLS with

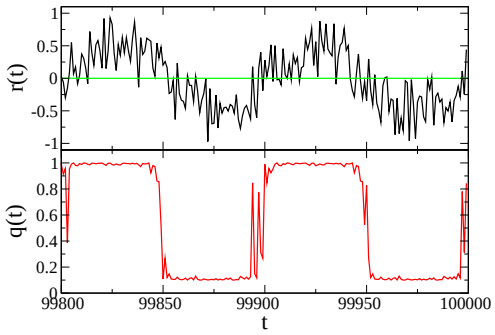


Figure 2: (Top)Return $r(t)$ and (bottom) investment control strategy $q(t)$ using GACE after $t = 10^5$ time steps, for returns with $T = 100$ and $\sigma = 0.5$.

different memory sizes for returns with different fixed periodicities, T, and no noise, finding that the most proper memory size, M, and the periodicity, T, are proportionally related by $M/T \approx 0.37$. Now, if we assume returns with no noise, we can find analytically the memory size $M^\star$ that maximizes the profits. For this, it can be shown that for periodic returns as in Eq. (4) with $\sigma = 0$, the strategy MLS, Eq. (7), estimates the next return $\hat{r}(t+1)$ as follows:

$$\hat{r}(t+1) = \frac{M+1}{M}\left[\sin(\omega t) - \sin(\omega t - \omega M)\right], \quad (8)$$

where $\omega = 2\pi/T$. Now, by calculating the average profits $\langle r\,q\rangle$ for the positive cycle of the returns:

$$\langle r\,q\rangle = \frac{T\,(M+1-\cos(\omega M))}{4M}. \quad (9)$$

And obtaining the derivative of $\langle r\,q\rangle$ w.r.t M:

$$\partial_M \langle r(t)\,q(t)\rangle = \frac{-T\sin\left(\frac{\omega}{2}M\right)^2 + \pi M \sin(\omega M)}{2M^2}. \quad (10)$$

It can be shown that by solving $\partial_M \langle r(t)\,q(t)\rangle = 0$ and using Taylor expansion to the sixth order for the sinusoidal functions, the memory size $M^\star$ that maximizes the profits corresponds to:

$$M^\star = \frac{\sqrt{\frac{3}{2}}}{\pi}T. \quad (11)$$

Consequently, the proportion $M/T \approx 0.37$ found by means of computer simulations approximates pretty well the proportion found analytically $M/T = \sqrt{\frac{3}{2}}/\pi = 0.389$.

Now, we compare the performance of the adaptive investment strategy GACE, presented in Sec. 3, with respect to the reference strategies presented in Sec. 4.2. For clarity, we assume for the moment that the strategy GACE uses fixed chromosome length, i.e. $G_j = G_{\max}$. For all strategies we consider $q_{\min} = 0.1$

and $q_{\max} = 1.0$ in our experiments. These parameter values describe the behavior of the strategies CP, Eq. (5), and SW, Eq. (6). For the strategy MLS, Eq. (7), we use Eq. (11) to determine the optimal memory size and for the strategy GACE we use the parameters in Table 1.

In our experiments we assume that the agent invests in returns with periodicity $T = 100$ for different noise levels. We consider here that the length of the chromosomes is fixed to $G_j = 100$ and a new generation of chromosomes is being obtained after a number of time steps $t_{\text{eval}} = 100$. For the computer experiments, we let the agent to use one of the strategies to invest during a number of $t = 10^5$ time steps. In order to account for the randomness of the scenario, we perform the experiment for a number of $N = 100$ trials, gathering the average budget obtained for each strategy at every 100 time steps.

Fig. 3 shows in a log-log plot the average budget, $\langle x \rangle$, for all strategies in the course of GACE's generations, τ. Except for the GACE strategy, all other strategies have a constant budget in average over each generation. This occurs because the average of the budget and the time steps to evaluate the population of chromosomes were taken at every 100 time steps, which corresponds to the periodicity of the returns $T = 100$. Noteworthy, after 4 and 300 generations GACE over-performs the strategies CP and MLS and after 400 generations it performs almost as well as the strategy SW.

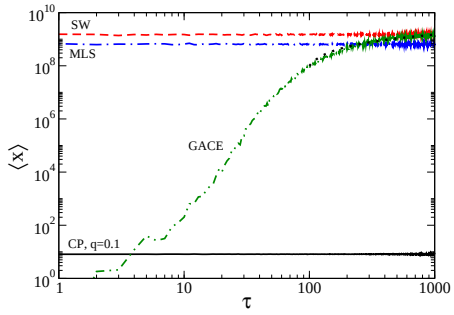


Figure 3: Average budget, $\langle x \rangle$, for different investment strategies in the course of generations τ, for returns with periodicity $T = 100$ and noise $\sigma = 0.1$.

4.4 Results for RoI with Changing Periodicity

In the previous section, we showed results for a stationary environment, now in this section we tackle a non-stationary environment.

4.4.1 GA Parameter Tuning

We used again the program +CARPS to find the best parameters for GACE now for returns with changing periodicity. The *GA* was configured for returns with a maximal periodicity of $T_{\max} = 100$ and maximal elapsing time steps $t_{\max} = 10^4$ for different level of noise: $\sigma = 0.1$, and $\sigma = 0.5$. In this process, we used the same intervals of definition as in Sec. 4.3.1, with the inclusion of the interval: $p_l \in [0.0, 1.0]$. The resulting best parameter values are shown in Table 2.

Table 2: GACE's best pars. for RoI with changing T.

C	p_c	p_m	s	p_l
1000	0.5	0.001	0.3	0.5

4.4.2 Performance Comparison

In this section we investigate the performance of the adaptive strategy with respect to the reference strategies in a non-stationary scenario. For this, we performed some computer experiments for returns with changing periodicity. As we did in the previous sections, we assumed for all strategies the parameter values $q_{\min} = 0.1$ and $q_{\max} = 1.0$ and for the strategy MLS we used Eq. (11) to calculate the memory size, M. For the strategy GACE, we used the parameter values listed in Table 2 and the length of a chromosome in the range $G_j \in (1, G_{\max})$, with $G_{\max} = 200$.

We show in Fig. 4 (top) the evolution of budget for different investment strategies, and (bottom) the corresponding periodicity of the returns, Eq.(4), both in the course of time. Thus, the best strategy is the strategy SW, following the strategy MLS; however, note that both strategies have total and partial knowledge about the dynamics of the returns, respectively. This previous knowledge gives some advantage to these strategies over the strategy GACE, which only needs the specification of $G_{\max}$. We note that the strategy GACE evolves quite fast, yielding a set of investment strategies with a clear tendency to lead more gains than losses.

5 CONCLUSIONS

In this paper, we presented a simple investment model and some investment strategies to control the proportion of investment in cyclic changing environments. The novelty of this paper is in the adaptive investment strategy here proposed, called *Genetic Algorithm for Changing Environments* (GACE), which is a new approach based on evolution for the correct mapping of

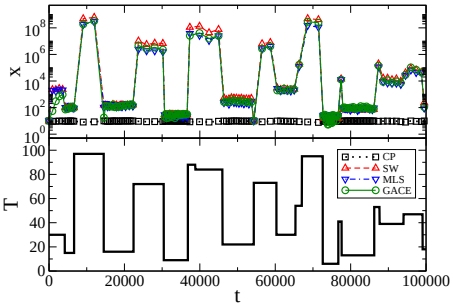


Figure 4: (Top) Budget for different strategies and (bottom) Periodicity of the returns, both in the course of time for RoI with parameters $T_{\max} = 100$, $t_{\max} = 10^4$ and $\sigma = 0.1$.

investment proportions to patterns that may be present in the returns. We analyzed the performance of GACE for different scenarios, and compared its performance in the course of time against other strategies used here as a reference. We showed that even though the strategy GACE has no knowledge of the dynamics of the returns, after a given number of time steps it may lead to large gains, performing as well as other strategies with some knowledge. This particularly is shown for long-lasting periodicities, where an ever increasing growth of budget was observed. Further work includes the analysis of the performance of the strategy GACE for real returns, and to compare the performance of GACE against other approaches from Machine Learning.

ACKNOWLEDGEMENTS

We thank to Dr. Dagmar Monett for providing us the program *+CARPS*. We also thank Frank Schweitzer and H.-D. Burkhard for helpful advice.

REFERENCES

Dawid, H. (1999). *Adaptive learning by genetic algorithms: Analytical results and applications to economic models*. Springer, Berlin.

Geibel, P. and Wysotzki, F. (2005). Risk-sensitive reinforcement learning applied to control under constraints. *J. of Artif. Intell. Res.*, 24:81–108.

Grefenstette, J. J. (1992). Genetic algorithms for changing environments. In Manner, R. and Manderick, B., editors, *Parallel Problem Solving from Nature 2.*, pages 137–144. Elsevier, Amsterdam.

Harvey, I. (1992). The sage cross: The mechanics of recombination for species with variable-length genotypes. In *Parallel Problem Solving from Nature*, volume 2, pages 269–278. Elsevier, Amsterdam.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press.

Kahneman, D. and Tversky, A. (1979). Prospect theory of decisions under risk. *Econometrica*, 47:263–291.

Kelly, J. L. (1956). A new interpretation of information rate. *AT&T Tech. J.*, 35:917–926.

LeBaron, B. (2001). Empirical regularities from interacting long and short horizon investors in an agent-based stock market. *IEEE Transactions on Evolutionary Computation*, 5:442–455.

Magdon-Ismail, M., Nicholson, A., and Abu-Mostafa, Y. (2001). Learning in the presence of noise. In Haykin, S. and Kosko, B., editors, *Intelligent Signal Processing*, pages 120–127. IEEE Press.

Michalewiçz, Z. (1999). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin.

Monett, D. (2004). +CARPS: Configuration of Metaheuristics Based on Cooperative Agents. In Blum, C., Roli, A., and Sampels, M., editors, *Proc. of the 1st Int. Workshop on Hybrid Metaheuristics, at the 16th European Conference on Artificial Intelligence, (ECAI '04)*, pages 115–125, Valencia, Spain. IOS Press.

Navarro, J. E. and Schweitzer, F. (2003). The investors game: A model for coalition formation. In Czaja, L., editor, *Proc. of the Workshop on Concurrency, Specification & Programming, CS&P '03*, volume 2, pages 369–381, Czarna, Poland. Warsaw University Press.

Navarro-Barrientos, J. E., Cantero-Alvarez, R., Rodrigues, J. F. M., and Schweitzer, F. (2008). Investments in random environments. *Physica A*, 387(8-9):2035–2046.

Richmond, P. (2001). Power law distributions and dynamic behaviour of stock markets. *The European Physical Journal B*, 20(4):523–526.

Schulenburg, S. and Ross, P. (2001). Strength and money: An LCS approach to increasing returns. In *Lecture Notes in Computer Science*, volume 1996, pages 114–137. Springer.

Turing, G. (1960). An introduction to matched filters. *IEEE T. Inform. Theory.*, 6(3):311–329.

Yang, S. (2005). Experimental study on population-based incremental learning algorithms for dynamics optimization problems. *Soft Computing*, 9(11):815–834.

PATH PLANNING FOR MULTIPLE FEATURES BASED LOCALIZATION

Francis Celeste, Frederic Dambreville
Dept. EORD FAS Team, DGA, 16 bis av. Prieur de la cte d'Or, 94114 Arcueil, France
francis.celeste@etca.fr, frederic.dambreville@etca.fr

Jean-Pierre Le Cadre
IRISA/CNRS, Campus de Beaulieu, 35042 Rennes, France
lecadre@irisa.fr

Keywords: Path planning, Cramèr Rao Bound, map-based localization, dynamic programming.

Abstract: In surveillance or exploration mission in a known environment, the localization of the dedicated sensor is of main importance. In this paper, we discuss the path planning problem for the localization algorithm which correlates range and bearing measurements and a map composed of several features. The sensor motion is designed from an information measure derived from the Fisher Information Matrix. It is shown that a closed form expression of the cost can be obtained. The optimal features location can be neatly geometrically interpreted. An integral cost which includes the sensor perception limitation is then formulated. It is used in a dynamic programming framework to tackle the path optimization problem.

1 INTRODUCTION

The path planning problem for map-based localization consists in designing the best trajectory for a mobile in a known environment, which guarantees the highest performance of positioning during its execution. Data collected from sensors are "matched" to a prior map to estimate the state (e.g., position and heading). Depending on the sensor dynamic and the observation models, different localization algorithms can be used. When the system is linear or near linear with Gaussian noises, Kalman-based approaches are relevant (Thrun et al., 2005; S. Thrun and Dellaert, 2000). In this paper, we introduce a framework to compute "optimal" path for a moving vehicle which collects range and bearing data from 2D features. One of the main challenges is to choose an appropriate measure to be optimized. In random estimation, the Fisher Information Matrix (FIM) can be used. We considered a D-optimal design (Paris and Le Cadre, 2002). The first interesting result of this work is the derivation of a closed form expression for the FIM determinant. It is shown that it depends on groups of two or three features. Then, a geometric analysis of the optimal features placement can be done. By exploiting this measure, we introduce an integral cost functional for a path space, which is composed of elementary moves with constant velocity and constant heading. Moreover, the sensor field of view limitations are included to the cost computation. At last, we formulate the problem as finding an optimal path on a graph by means of dynamic programming. The paper ends with one illustrative example.

2 PROBLEM FORMULATION

We consider a moving sensor evolving according to the dynamic model

$$\begin{aligned} \dot{x}_t &= v_t \cos\varphi_t, \\ \dot{y}_t &= v_t \sin\varphi_t, \\ \dot{\varphi}_t &= \omega_t. \end{aligned} \tag{1}$$

where its state $\mathbf{X}_t \triangleq [x_t, y_t, \varphi_t]$ is composed of its 2-D position and its orientation. A feature map of its environment is available for localization purpose. In equation 2, we assume that the known control $\mathbf{u}_t \triangleq [v_t, \omega_t] \in \mathcal{U} \subset \mathbb{R}^2$. During its displacement, the mobile gets sensor measurements from detected features which are in the embedded map. Let us denote $\mathbf{f}_t \triangleq \{f_1, \ldots, f_{m_t}\}$ the set of m_t features visible and used in the localization process at time t. Each feature is defined by its 2D position in a global frame

$\mathcal{R}_g \triangleq (O, \overrightarrow{u}, \overrightarrow{v})$:

$$f_i \leftrightarrow \left(x^i, y^i\right) \in \mathcal{D} \subset \mathbb{R}^2. \tag{2}$$

and the "sensor-feature" vector $\delta p_i(t) \triangleq \left[x^i - x_t, y^i - y_t\right]^*$. The measurements vector is the stacked vector $\mathbf{Z}_t = \left[z_1^t, \ldots, z_{m_t}^t\right]$ where z_i^t is the range and bearing measurement for feature f_i. So, the observation model stands as follows :

$$\mathbf{Z}_t = \mathbf{H_t}(\mathbf{X}_t, \mathbf{f}_t) + \mathbf{W_t}. \tag{3}$$

where the $2 \times \mathrm{i}^{\text{th}}$ and $2 \times \mathrm{i} + 1^{\text{th}}$ elements of $\mathbf{H_t}(\mathbf{X}_t, \mathbf{f}_t)$ are the components of the two dimensional vector $h(\mathbf{X}_t, f_i)$ given by

$$z_i^t = h(\mathbf{X}_t, f_i) + w_t^i. \tag{4}$$

$$h(\mathbf{X}_t, f_i) \triangleq \begin{cases} \sqrt{(x_t - x^i)^2 + (y_t - y^i)^2} \\ \operatorname{atan}_2(\frac{y^i - y_t}{x^i - x_t}) - \varphi_t \end{cases} \tag{5}$$

The noise vector w_t^i is modelled by an i.i.d. Gaussian process with zero mean and covariance matrix Σ_t^i. Moreover, we suppose that $\Sigma_t^i = \Sigma, \forall i$ and

$$\Sigma = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\varphi^2 \end{pmatrix}. \tag{6}$$

We also consider that w_t^j and w_t^l are independent for $l \neq j$. So in light of (2), the likelihood function is given by

$$p(\mathbf{Z}_t | \mathbf{X}_t) \propto \exp\left(-\frac{1}{2}\sum_{l=1}^{m_t} \|z_l - h(\mathbf{X}_t, f_l)\|_\Sigma^2\right). \tag{7}$$

If $\hat{\mathbf{X}}_t$ is one estimate based on the measurement $\mathbf{Z}_t$ (e.g., the maximum likelihood estimate), the covariance error $e_{\mathbf{X}_t} = \mathbf{X}_t - \hat{\mathbf{X}}_t$ is lower bounded by the *Cramer Rao Bound* (CRB) (Van Trees, 1968).

$$Cov(e_{\mathbf{X}_t}) \succ F^{-1}(t). \tag{8}$$

The calculation of the FIM $F(t)$ is given in our case by,

$$F = \sum_{i=1}^{m_t} \left(\frac{\partial h(\mathbf{X}_t, f_i)}{\partial \mathbf{X}_t}\right)^* \Sigma^{-1} \left(\frac{\partial h(\mathbf{X}_t, f_i)}{\partial \mathbf{X}_t}\right). \tag{9}$$

The elementary gradient vector can be derived straightforwardly

$$\frac{\partial h(\mathbf{X}_t, f_i)}{\partial \mathbf{X}_t} = \begin{pmatrix} c_i & s_i & 0 \\ -\frac{s_i}{\rho_i} & \frac{c_i}{\rho_i} & -1 \end{pmatrix}. \tag{10}$$

where $\alpha_i(t) \triangleq \angle \overrightarrow{u}\, \delta p_i(t)$, $\rho_i \triangleq ||\delta p_i(t)||$, $c_i \triangleq \cos\alpha_i$ and $s_i \triangleq \sin\alpha_i$. Let us also introduce the following notations :

- $\vec{c} \triangleq [c_1 \cdots c_{m_t}]^*$, $\vec{s} \triangleq [s_1 \cdots s_{m_t}]^*$,
- $\vec{c}_\rho \triangleq [\frac{c_1}{\rho_1} \cdots \frac{c_{m_t}}{\rho_{m_t}}]^*$, $\vec{s}_\rho \triangleq [\frac{s_1}{\rho_1} \cdots \frac{s_{m_t}}{\rho_{m_t}}]^*$.
- $1_{m_t} \triangleq [1 \cdots 1]^*$, $0_{m_t} \triangleq [0 \cdots 0]^*$

Without loss of generality, we set $\sigma_d = \sigma_\varphi = 1$ then we can rewrite[1]

$$F(t) = G(t)G(t)^*. \tag{11}$$

with

$$G(t) = \begin{pmatrix} \overbrace{\vec{c}^t}^{\mathcal{G}_1(t)} & \overbrace{\vec{s}_\rho^{\,t}}^{\mathcal{G}_2(t)} \\ \vec{s}^t & -\vec{c}_\rho^{\,t} \\ 0_{m_t} & 1_{m_t} \end{pmatrix}. \tag{12}$$

$G(t)$ is a $3 \times 2\mathrm{m_t}$ matrix with columns G_i are part of the subset $\mathcal{G}_1(t)$ or $\mathcal{G}_2(t)$:

$$\mathcal{G}_1(t) = \left\{G_{i_1}, 1 \leq i_1 \leq m_t | G_{i_1} = \begin{pmatrix} c_{i_1} & s_{i_1} & 0 \end{pmatrix}^*\right\},$$
$$\mathcal{G}_2(t) = \left\{G_{i_2}, 1 \leq i_2 \leq m_t | G_{i_2} = \begin{pmatrix} \frac{s_{i_2}}{\rho_{i_2}} & -\frac{c_{i_2}}{\rho_{i_2}} & 1 \end{pmatrix}^*\right\}.$$

In this paper, we are dealing with the optimization of the sequence of displacement which provides the "best" estimate of the state. This can be achieved using an appropriate measure of information gain. We adopt here a D-optimal design considering the determinant of the FIM[2]. In the next section, we show that this measure is a function implying the estimated bearings angles $(\alpha_i(t))_{i=1}^{m_t}$ and relative ranges $(\rho_i(t))_{i=1}^{m_t}$.

3 DERIVATION OF $\det(F)$

Let us define $\mathcal{L}(t)$ as the determinant of the FIM at time t in position $\mathbf{X}_t$. From (11), we have

$$\mathcal{L}(t) = \det(G(t)G(t)^*). \tag{13}$$

Using the Binet-Cauchy formula[3], we can notice that

$$\mathcal{L}(t) = \sum_{1 \leq i < j < k \leq 2m_t} \left\{\det(G_i, G_j, G_k)\right\}^2. \tag{14}$$

hence to compute $\mathcal{L}(t)$, we have to enumerate the different cases in accordance with the column vectors (G_i, G_j, G_k) are in $\mathcal{G}_1$ or $\mathcal{G}_2$. In the following, we denote $d_{ijk} \triangleq \det(G_i, G_j, G_k)$. If all columns are in $\mathcal{G}_1$, d_{ijk} is trivially equal to zero. Using determinant computation properties and relations betweeen trigonometric functions, we get

[1] $*$ is the transpose operator

[2] other matrix operator can be used, such as the *trace*

[3] $\det(AB) = \sum_S \det(A_s)\det(B_s)$, $S = \{1, \cdots, n\}$, if $A \in \mathcal{M}_\mathbb{K}(m,n)$ et $B \in \mathcal{M}_\mathbb{K}(n,m)$, A_s is the $m \times n$ matrix whose columns are those of A with in S

a) $G_i, G_j \in \mathcal{G}_1$ and $G_k \in \mathcal{G}_2$

$$d^1_{ijk} = \sin(\alpha_i - \alpha_j).$$

b) $G_i \in \mathcal{G}_1$ and $G_k,\ G_j \in \mathcal{G}_2$

$$d^2_{ijk} = \frac{\cos(\alpha_i - \alpha_k)}{\rho_k} - \frac{\cos(\alpha_i - \alpha_j)}{\rho_j}.$$

c) $G_i \in \mathcal{G}_1,\ G_j$ and $G_k \in \mathcal{G}_2$

$$d^3_{ijk} = \frac{\sin(\alpha_i - \alpha_k)}{\rho_i \rho_k} + \frac{\sin(\alpha_i - \alpha_j)}{\rho_i \rho_j} + \frac{\sin(\alpha_j - \alpha_k)}{\rho_j \rho_k}.$$

In conclusion, we notice that $\mathcal{L}(t)$ is the sum of three terms $\mathcal{L}_1(t)$, $\mathcal{L}_2(t)$ and $\mathcal{L}_3(t)$ which characterize interactions between pairs and triplets of visible features.

$$\mathcal{L}(t) = a_1\mathcal{L}_1(t) + a_2\mathcal{L}_2(t) + a_3\mathcal{L}_3(t). \tag{15}$$

with $\mathcal{L}_1(t) = \sum_{i=1}^{m_t}\sum_{j>i}^{m_t} g_1(f_i, f_j)$, $\mathcal{L}_2(t) = \sum_{i=1}^{m_t}\sum_{j=1}^{m_t}\sum_{k>j}^{m_t} g_2(f_i, f_j, f_k)$ and $\mathcal{L}_3(t) = \sum_{i=1}^{m_t}\sum_{j>i}^{m_t}\sum_{k>j}^{m_t} g_3(f_i, f_j, f_k)$ where $(g_l)_{l\in\{1,2,3\}}$ are respectively given by the square of d^l_{ijk} in the above cases. Coefficients $(a_l)_{1\le l\le 3}$ depend on σ_r and σ_φ.

4 THE OPTIMAL PLACEMENT OF THE FEATURES

We now study the location of the features which provides the best performance of estimation around a given mean state $\mathbf{X}$. The analysis takes into account the sensor field of view and only consider $\mathcal{L}_1(t)$ (pairs interaction). Such an approximation is valid when $\frac{\sigma_d}{\rho} \ll \sigma_\varphi$. Let $(f_i)_{1\le i\le n}$ be visible from state $\mathbf{X}$. We introduce $P = (\bar{x};\bar{y})$, $(\vec{v}_i)_{1\le i\le n}$, D_m, $\vec{v}_-$ and $\vec{v}_+$ (see figure 1). D_m is the angular aperture of the sensor field of view. An analogy can be made with the reasoning

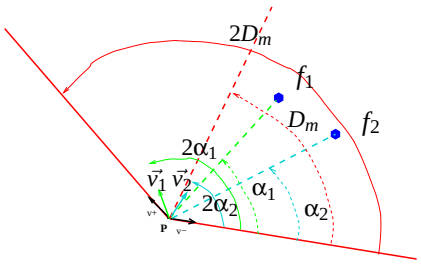


Figure 1: Sensor features spatial configuration.

in (Gu et al., 2006) for multiple UAVs cooperation for sensing. The derivation made here is nevertheless simpler and more geometrically intuitive.

Proposition 1. *Maximizing $\mathcal{L}_1(t)$ is equivalent to find the configuration $(\vec{v_1}^*, \dots, \vec{v_n}^*)$ which minimizes $||\vec{v_T}|| = ||\sum_{i=1}^n \vec{v_i}||^2$.*

Indeed, using classic trigonometric properties[4] we can show that $\mathcal{L}_1 = \frac{1}{4}\left(1 - ||\sum_{i=1}^n \vec{v_i}||^2\right)$.

4.1 Optimal Placement for $D_m < \frac{\pi}{2}$

In this context, the value of the angle made by vectors $\vec{v_i}$ and $\vec{v_j}$ is strictly smaller than π. So $||\vec{v_T}|| > 0$. Let $i_0 \in \{1, \cdots, n\}$ and $\theta_{i_0} = \angle\vec{v_-}\vec{v_{i_0}}$. We also denote $\vec{v_{\underline{i_0}}} \triangleq \sum_{j\neq i_0}\vec{v_j}$ and $\theta_{\underline{i_0}} = \angle\vec{v_-}\vec{v_{\underline{i_0}}}$

$$\begin{aligned} ||\vec{v_T}||^2 &= ||\vec{v_{i_0}} + \vec{v_{\underline{i_0}}}||^2 \\ &= 1 + ||\vec{v_{\underline{i_0}}}||^2 + 2||\vec{v_{\underline{i_0}}}||\cos\left(\theta_{\underline{i_0}} - \theta_{i_0}\right). \end{aligned}$$

As $D_m < \frac{\pi}{2}$, $\vec{v_{\underline{i_0}}}$ is also between $\vec{v_-}$ and $\vec{v_+}$. So, for a given placement of vectors $\{\vec{v_i}\}_{i\neq i_0}$, $||\vec{v_T}||$ is minimized for $\theta^*_{i_0}$ which makes $g(\theta_{i_0}) = \cos\left(\theta_{\underline{i_0}} - \theta_{i_0}\right)$ minimum.

Proposition 2. *In the optimal configuration, each vector $\vec{v_i}$ is on the frontier of the visibility cone.*

Proof. $0 \le \theta_{\underline{i}}, \theta_{i_0} \le 2D_m \Rightarrow \theta_{\underline{i_0}} - 2D_m \le \theta_{\underline{i_0}} - \theta_{i_0} \le \theta_{\underline{i_0}}$. Moreover, $\theta_{\underline{i_0}} - 2D_m > -\pi$ et $\theta_{\underline{i_0}} < \pi$. We can easily deduce that

$$\theta^*_{i_0} = \begin{cases} 2D_m & \text{if } |\theta_{\underline{i_0}} - 2D_m| > \theta_{\underline{i_0}} \\ 0 & \text{if } |\theta_{\underline{i_0}} - 2D_m| < \theta_{\underline{i_0}} \end{cases}.$$

which proves that either $\vec{v_{i_0}} = \vec{v_-}$ or $\vec{v_{i_0}} = \vec{v_+}$. Let us denote n_- and n_+ the number of vectors $\vec{v_i}$ respectively equal to $\vec{v_-}$ and $\vec{v_+}$ ($n_- + n_+ = n$). n_- must verify the relation

$$||\vec{v_T}||^2 = 2(1-a)n_-^2 - 2(1-a)nn_- + n^2 \triangleq f(n_-).$$

with $a = \cos(2D_m)$ ($a < 1$). f is minimal for $n_- = \frac{n}{2}$, so

- if n is even, $n_- = n_+ = \frac{n}{2}$ and which provides $\boxed{\mathcal{L}_1 = \frac{n^2}{4}\sin^2(D_m)}$.
- else we can set $n_- = \frac{n-1}{2}$ and $n_+ = \frac{n+1}{2}$, then $\boxed{\mathcal{L}_1 = \frac{n^2-1}{4}\sin^2(D_m)}$.

[4] $\sin^2 a = \frac{1}{2}(1 - \cos 2a)$ and $\cos(a-b) = \cos a\cos b + \sin a \sin b$

4.2 Optimal Placement for $D_m > \frac{\pi}{2}$

In this case, we have to make a different reasoning according to the parity of n. When n is even, the optimal solution is obvious as we can place the features so that $\vec{v_T} = \vec{0}$. Indeed, it is enough to choose $\{\vec{v_1}, \cdots, \vec{v_n}\}$ pairwise such that their difference angle is equal to π (i.e. orthogonal assignment of features). We can notice that, there are plenty of such configurations and the cost is $\boxed{\mathcal{L}_1 = \frac{n^2}{4}}$. Otherwise, if n is odd, it is more difficult to find a placement which gives $\vec{v_T} = \vec{0}$. Nevertheless, we can search among a particular class of configurations with $\vec{v_{i_0}} = -\vec{v_{\underline{i_0}}}$. Assuming $i_0 = n$, one way to obtain $\vec{v_{\underline{n}}}$ collinear and opposite to $\vec{v_n}$, is to choose $\{\vec{v_1}, \cdots, \vec{v_{n-1}}\}$ where

$$\exists \varphi \in \left]\frac{\pi}{2}, \pi\right[, \begin{cases} \angle \vec{v_i} \vec{v_n} = \varphi, & \forall i \in \{1, \cdots, \frac{n-1}{2}\}, \\ \angle \vec{v_n} \vec{v_{j_i}} = \varphi, & \forall j_i = i + \frac{n-1}{2}. \end{cases}$$

Given $\angle \vec{v_-} \vec{v_i} = \theta_p$, $\forall i \in \{1, \cdots, \frac{n-1}{2}\}$ and supposing $\vec{v_-} = \vec{u}$, then

$$\begin{array}{rcl} \vec{v_n} & = & \cos(\varphi + \theta_p)\vec{u} + \sin(\varphi + \theta_p)\vec{v}, \\ \vec{v_i} & = & \cos(\theta_p)\vec{u} + \sin(\theta_p)\vec{v}, \ \forall i, \\ \vec{v_{j_i}} & = & \cos(2\varphi + \theta_p)\vec{u} + \sin(2\varphi + \theta_p)\vec{v}, \ \forall j_i. \end{array}$$

and $\forall i \in \{1, \cdots, \frac{n-1}{2}\}$

$$\begin{array}{rcl} \vec{v_i} + \vec{v_{j_i}} & = & \cos(\theta_p) + \cos(2\varphi + \theta_p)\vec{u} \\ & & + \sin(\theta_p) + \sin(2\varphi + \theta_p)\vec{v}. \end{array} \quad (16)$$

Using trigonometric properties, we get that:

$$\begin{array}{rcl} \vec{v_i} + \vec{v_{j_i}} & = & 2\cos(\varphi)\left(\cos(\varphi + \theta_p)\vec{u} + \sin(\varphi + \theta_p)\vec{v}\right) \\ & = & 2\cos(\varphi)\vec{v_n}. \end{array}$$

To make $\vec{v_T} = \vec{0}$, we must force

$$\vec{v_n} + \sum_{i=1}^{\frac{n-1}{2}} \vec{v_i} + \vec{v_{j_i}} = \vec{0},$$

which is equivalent to the following condition on φ.

$$l(\varphi) \stackrel{\Delta}{=} 1 + (n-1)\cos(\varphi) = 0, \ \varphi \in \left[\frac{\pi}{2}, \pi\right[. \quad (17)$$

As the field of view is limited, we have to satisfy $\varphi \leq D_m$. Therefore, if such an angle exists, the cost value is again $\mathcal{L}_1 = \frac{n^2}{4}$. In particular, if $D_m > \frac{2\pi}{3}$, we can always find an optimal placement. Indeed, it is sufficient to choose $n-3$ vectors as in the even case (orthogonal assignment) and to use the last three with $\varphi = \frac{2\pi}{3}$. When exists φ solution of (17) with $D_m < \varphi < \frac{2\pi}{3}$, it seems difficult to find a configuration which allows to attain the maximum cost. But, we propose a suboptimal solution which minimizes $l(\varphi)$. l is decreasing on $\left[\frac{\pi}{2}, D_m\right]$ ($\frac{\partial l}{\partial \varphi} \propto -\sin(\varphi) < 0$) so its maximum is given for $\varphi = D_m$. This leads to the cost value

$$\boxed{\mathcal{L}_1 = \frac{1}{4}\left(n^2 - (1 + (n-1)\cos(D_m))^2\right)}.$$

In this section, we made a geometric analysis to determine the optimal placement of the features to maximize the cost $\mathcal{L}_1$. Making the same kind of reasoning for the complete cost $\mathcal{L}(t)$ is much more challenging. After this static analysis, we deals with the path planning problem in the next section. For the sake of brevity, we only detail the approach for $\mathcal{L}_1(t)$ but it can be generalized to $\mathcal{L}_2(t)$ and $\mathcal{L}_3(t)$.

5 PATH PLANNING

We consider the evolution of the sensor between $[t_0, t_f]$ with $0 < t_f \leq T*$ from position $q_s \in \mathcal{D}$ to position $q_t \in \mathcal{D}$. We look for paths $(\mathbf{X}_t)_{t \in [t_0, t_f]}$ which maximizes the cost

$$\Psi([t_0, t_f]) = \int_{t_0}^{t_f} \mathcal{L}_1(t)\mathrm{d}t. \quad (18)$$

The problem can be formalize in the optimal control framework with two boundaries constraints. Unfortunately, due to the cost expression and the sensor field of view (FOV) limitations, no analytic formulation of the optimal path can be derived. An approximated approach based on the discretization of the state and control space seems more tractable.

5.1 Path Description

As in (Celeste et al., 2007), We formalize here the problem as a discrete path planning. A regular grid is considered and one path is a sequence of elementary displacements with constant heading $\left(\varphi \in \{\varphi_i = \frac{i*\pi}{4}, \ i \in \{-3, ..., 4\}\}\right)$ and constant velocity v (a leg). For a path τ with n_τ legs, the cost is as follows:

$$\Psi([t_0, t_f]) = \sum_{i=0}^{n_\tau - 1} \int_{t_i}^{t_{i+1}} \mathcal{L}_1(t)\mathrm{d}t. \quad (19)$$

$\mathbf{X}_{t_0} = q_s$ and $\mathbf{X}_{t_{n_\tau - 1}} = q_t$ are supposed to be on the grid. Some constraints on the maneuvers can be imposed to avoid chaotic behavior (e.g. bang-bang effect)(Paris and Le Cadre, 2002). To solve the planning task we need to compute the cost associated with each leg. First of all, it is necessary to determine the part of the leg where each feature is visible due to the sensor FOV.

5.2 Cost for One Leg

For a FOV model with an aperture 2Δ and a maximum range detection R_d, the area $\mathcal{Z}$ visible from the leg e is composed of three regions $\mathcal{Z}_1$, $\mathcal{Z}_2$ and $\mathcal{Z}_3$ (see Figure 3). A pair of features $(f_i, f_j) \in \mathcal{Z}^2$ are visible from $P_-^{ij}(x_-^{ij}, y_-^{ij})$ and $P_+^{ij}(x_+^{ij}, y_+^{ij})$. These limits can be derived using a simple geometric reasoning. Moreover,

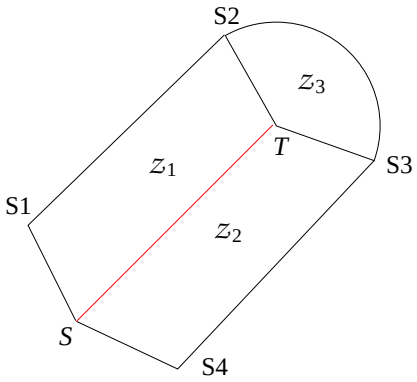


Figure 2: The visible region for one leg.

we have a relation between an elementary displacement and the associated duration ($dt \propto dx$ if $\varphi \neq \frac{\pi}{2}[\pi]$, $dt \propto dy$ else). and the leg can be reparametrized as follows:

- $y(x) = \beta + \gamma x, \forall x \in [x_S, x_T]$ if $\varphi \neq \frac{\pi}{2}[\pi]$ (non vertical motion),
- $x = x_S, y_S \leq y \leq y_T$ else (vertical motion),

The total cost for a leg e can then be computed using relevant change of variable.

For non vertical displacement, the cost due to a pair of features (f_i, f_j) is the integral of a rational function:

$$K_{ij}(x) = \frac{\left[(x-x^i)(y(x)-y^j) - (x-x^j)(y(x)-y^i)\right]^2}{p_i(x)p_j(x)}.$$

where $p_l(x) = (x-x^l)^2 + (y(x)-y^l)^2 \stackrel{\Delta}{=} a_l x^2 + b_l x + c_l$, $l \in \{j,i\}$ is the respective square range of f_i, f_j to the sensor. Therefore, these polynomials are irreducible whatever the sensor position in $\mathcal{D} \setminus \{f_i, f_j\}$. We can rewrite

$$K_{ij}(x) = \frac{(A_{ij}x + B_{ij})^2}{p_i(x)p_j(x)}. \quad (20)$$

So, we have to compute:

$$c_{ij}^{nv}(e) \propto \int_{x_-^{ij}}^{x_+^{ij}} K_{ij}(x)\mathrm{d}x. \quad (21)$$

which can be done with a relevant partial expansion of the rational function. Nevertheless, we have to pay attention to the position of the leg relatively to the features.

case (1) e is on the perpendicular bisector of $[f_i\, f_j]$, then $p_j(x) = p_i(x), \forall x$ and

$$\frac{(A_{ij}x + B_{ij})^2}{p_i(x)p_j(x)} = \frac{r_1 x + s_1}{p_i(x)} + \frac{r_2 x + s_2}{p_i^2(x)}. \quad (22)$$

case (2) e is not on the perpendicular bisector of $[f_i\, f_j]$, then

$$\frac{(A_{ij}x + B_{ij})^2}{p_i(x)p_j(x)} = \frac{r_1 x + s_1}{p_i(x)} + \frac{r_2 x + s_2}{p_j(x)}. \quad (23)$$

Identification of the numerators yields in both cases to a linear system to deduce $\chi = [r_1\, r_2\, s_1\, s_2]^*$,

$$M_{ij}^{(c)}\chi = \mathcal{B}_{ij}, \text{ for cases } c = 1,2 \quad (24)$$

$$M_{ij}^{(1)} = \begin{pmatrix} a_i & 0 & 0 & 0 \\ b_i & 0 & a_i & 0 \\ c_i & 1 & b_i & 0 \\ 0 & 0 & c_i & 1 \end{pmatrix}, \mathcal{B}_{ij} = \begin{pmatrix} 0 \\ A_{ij}^2 \\ 2A_{ij}B_{ij} \\ B_{ij}^2 \end{pmatrix} \quad (25)$$

and

$$M_{ij}^{(2)} = \begin{pmatrix} a_i & a_j & 0 & 0 \\ b_i & b_j & a_i & a_j \\ c_i & c_j & b_i & b_j \\ 0 & 0 & c_i & c_j \end{pmatrix} \quad (26)$$

For vertical displacements, it is more appropriate to consider integration with the variable y. The same reasoning leads to the integration of a rational function to get the cost expression

$$c_{ij}^{v}(e) \propto \int_{y_-^{ij}}^{y_+^{ij}} K_{ij}(y)\mathrm{d}y. \quad (27)$$

5.2.1 Closed Form Expression for the Cost

Whatever the leg orientation, we have to deals with the computation of integrals of the form ($n \in \{1,2\}$, $l \in \{i,j\}$):

$$H^{(n)}(l,u,v,x_-,x+) = \int_{x_-}^{x_+} \frac{ux+v}{(ax^2+bx+c)^n}\mathrm{d}x \quad (28)$$

Using specific changes of variable and classic primitives, the closed form expression for the cost (21), (27) can be derived. For instance,

$$H^{(1)}(l,u,v,x_-,x+) = \nu_l^{(1)} \ln\left(\frac{|p_l(x_+)|}{|p_l(x_-)|}\right) +$$

$$\lambda_l^{(1)}\left(\tan^{-1}(q_l(x_+ + \frac{b_l}{2a_l})) - \tan^{-1}(q_l(x_- + \frac{b_l}{2a_l}))\right)$$

where $q_l = \sqrt{\frac{4a_l^2}{4a_l c_l - b_l^2}}$, $\nu_l^{(1)} = \frac{u}{2a_l}$ and $\lambda_l^{(1)} = \frac{2va_l - ub_l}{2a_l^2} q_l$.
The expressions of the costs are finally

$$c_{ij}(e) = H^{(1)}(i, r_1, s_1, x_-^{ij}, x_+^{ij}) + H^{(n)}(j, r_2, s_2, x_-^{ij}, x_+^{ij})$$

where $n \in \{1,2\}$ depends on the leg orientation according to $[f_i, f_j]$. Given the contribution of each visible pair of features, the complete cost of the leg is given by $c(e) = \sum_{i,j} c_{ij}(e)$ Therefore, the cost associated to a path $\tau = \{e_1, \cdots, e_n\}$ of length $n = n_\tau - 1$ is $c(\tau) = \sum_{i=1}^{n} c(e_i)$. The optimization can then be solved via dynamic programming.

6 EXPERIMENT

In this experiment, we consider an embedded map composed of ten features organised on the border of $\mathcal{D} = [0;200;0;200]$. The sensor FOV is characterized by a maximum range detection $R_{max} = 70m$ and a half aperture angle $D_m = 120$ deg.. Moreover, the authorized difference angle between two following time steps must be bounded by $\pi/4$ and the path length smaller than $l_{max} = 98$ legs from $q_s = (20;20)$ to $q_t = (170;20)$. The grid resolutions are $\delta x = \delta y = 10$. The algorithm seems to behave well. The sensor

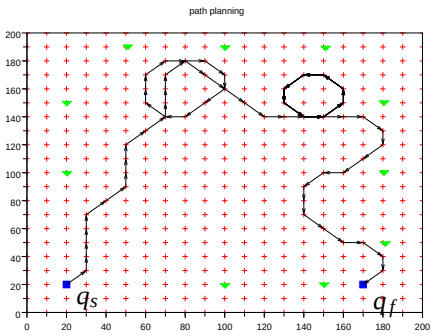


Figure 3: Optimal path, features(green), q_s and q_f (blue).

moves in order to be as soon as possible on the perpendicular bisector of pairs of features and to increase the number of visible pairs. The proposed path allows to provide better triangulation conditions which improves the estimation process. Moreover some interesting behaviour like cycles can also be observed.

7 CONCLUSIONS AND PERSPECTIVES

In this paper, we introduced a path planning algorithm for map based localization. First of all, we derived an information gain as the determinant of the Fisher Information Matrix adapted to multiple features. A geometric interpretation of this measure was made. Then, to determine the optimal path, we considered the integral cost of this function. It is important to notice that the cost computation take into account the sensor field of view model. Finally, we applied the approach on a scenario and illustrate the behaviour of the algorithm. We detailed the approach for only the first part of the total cost, but it can be generalized to the others. Now, we plan to take into account noisy feature positions which will yields to a path planning problem with uncertain cost. Then, the next challenge is to find optimal paths which tackle also those uncertainties on the given map.

REFERENCES

Celeste, F., Dambreville, F., and Le Cadre, J.-P. (2007). Optimal strategies based on the cross entropy method. In *Fusion 2007*, Quebec (Canada).

Gu, G., Chandler, P., Schumacher, C., Sparks, A., and Pachter, M. (2006). Optimal cooperating sensing using a team of uavs. *IEEE Trans. on Aerospace and Electronic Systems*, 42(4).

Paris, S. and Le Cadre, J.-P. (2002). Trajectory planning for terrain-aided navigation. In *Fusion 2002*.

S. Thrun, D. Fox, W. B. and Dellaert, F. (2000). Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 28(1-2):99–141.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT press.

Van Trees, H. (1968). *Detection, Estimation and Modulation Theory*. New York Wiley.

NEURAL NETWORK AND GENETIC ALGORITHMS FOR COMPOSITION ESTIMATION AND CONTROL OF A HIGH PURITY DISTILLATION COLUMN

J. Fernandez de Cañete, P. del Saz-Orozco and S. Gonzalez-Perez
System Engineering and Automation Dpt., University of Malaga, Malaga, Spain
canete@isa.uma.es, *delsaz@isa.uma.es*, *sgp@isa.uma.es*

Keywords: Distillation control, software sensors, neural networks, genetic algorithms.

Abstract: Many industrial processes are difficult to control because the product quality cannot be measured rapidly and reliably. One solution to this problem is neural network based control, which uses an inferential estimator (software sensor) to infer primary process outputs from secondary measurements, and control these outputs. This paper proposes the use of adaptive neural networks applied both to the prediction of product composition from temperature measurements, and to the dual control of distillate and bottom composition for a continuous high purity distillation column. Genetic algorithms are used to automatically choice of the optimum control law based on the neural network model of the plant. The results obtained have shown the proposed method gives better or equal performances over other methods such fuzzy, or adaptive control.

1 INTRODUCTION

Nowadays, advanced control systems are playing a fundamental role in plant operations because they allow for effective plant management. Typically, advanced control systems rely heavily on real-time process modelling, and this puts strong demands on developing effective process models that, as a prime requirement, have to exhibit real-time responses. Because in many instances detailed process modelling is not viable, efforts have been devoted towards the development of approximate dynamic models.

Approximate process models are based either on first principles, and thus require good understanding of the process physics, or on some sort of black-box modelling. Neural network modelling represents an effective framework to develop models when relying on an incomplete knowledge of the process under examination. Because of the simplicity of neural models, they exhibit great potentials in all those model-based control applications that require real-time solutions of dynamic process models. The better understanding acquired on neural network modelling has driven its exploitation in many chemical engineering applications.

For many reasons, distillation remains the most important separation technique in chemical process industries around the world. Therefore, improved distillation control can have a significant impact on reducing energy consumption, improving product quality and protecting environmental resources. However, both distillation modelling and control are difficult task because it is usually a nonlinear, non-stationary, interactive, and subject to constraints and disturbances process. Nevertheless, process identification and optimization (Bhat and McAvoy 1990) (Bulsari 1995), software sensor development (Zamprogna et al 2001), fault analysis and process control (Hussain 1999) (Xiong and Jutan 2002) works have been successfully reported in this field.

Genetic algorithms (GA) are model machine learning methodologies, which derive their behaviour from a metaphor of the processes of evolution in nature and are able to overcome complex non-linear optimization tasks like non-convex problems, non-continuous objective functions, etc. (Michalewitz 1992). They are based on an initial random population of solutions and an iterative procedure, which improves the characteristics of the population and produces solutions that are closer to the global optimum. This

is achieved by applying a number of genetic operators to the population, in order to produce the next generation of solutions. GAs have been used successfully in combinations with neural and fuzzy systems (Fleming and Purshouse 2002).

In this paper we describe the application of adaptive neural networks to the estimation of the product compositions in a binary methanol-water continuous distillation column from available on-line temperature measurements. This software sensor is then applied to train a neural network model so that a GA performs the searching for the optimal dual control law applied to the distillation column. The performance of the developed neural network based estimator is further tested by observing the performance of the designed neural network based control system for both set point tracking and disturbance rejection cases.

2 PROCESS DESCRIPTION

The distillation column used in this study is designed to separate a binary mixture of methanol and water, which enters as a feed stream with flow rate F_{vol} and composition X_F between the rectifying and the stripping section, obtaining both a distillate product stream D_{vol} with composition X_D and a bottom product stream B_{vol} with composition X_B. The column consists of 40 bubble cap trays. The overhead vapour is totally condensed in a water cooled condenser (tray 41) which is open at atmospheric pressure. The process inputs that are available for control purposes are the heat input to the boiler Q and the reflux flowrate L_{vol}. Liquid heights in the column bottom and the receiver drum (tray 1) dynamics are not considered for control since flow dynamics are significantly faster than composition dynamics and pressure control is not necessary since the condenser is opened to atmospheric pressure.

The model of the distillation column used throughout the paper is developed by (Diehl et al 2001), composed by the mass, component mass and enthalpy balance equations used as basis to implement a SIMULINK model (figure 1) which describes the nonlinear column dynamics as a 2 inputs (Q , L_{vol}) and 2 output (X_D , X_B). Implementations details for the overall column dynamics are given in (Fernandez de Canete et al 2007).

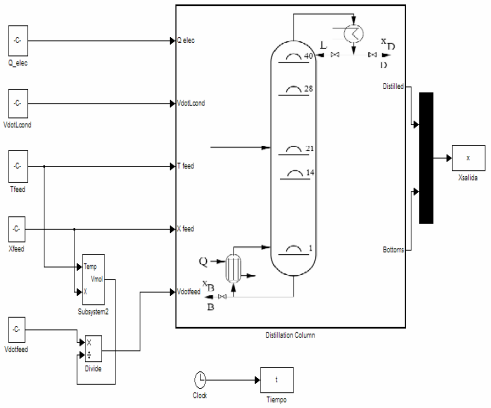


Figure 1: Schematic of the SIMULINK model of the distillation.

3 NEURAL ESTIMATOR AND CONTROLLER

The complete neural network based estimation and control system is described below (Figure 2).

3.1 Neural Composition Estimator

In order to infer the composition from temperature a neural network is used. Previously, a sensitivity study is performed in order to choose the correct set of temperatures to infer top and bottom compositions, resulting a three temperature vector $T[k] = [T_{41}[k], T_{21}[k], T_1[k]]$ selected as input to the neural network predictor which outputs the predicted values of composition vector $\hat{y}[k] = [\hat{X}_D[k], \hat{X}_B[k]]$. Normally, in a plant operation, both real values are measured off-line in the laboratory. In this study, the neural network parameter update is made accepting the simulation results as same with the actual plant data. Training set for a 3-layer net (3-15-2 units) is generated by selecting 1200 temperature data points obtained by during column open loop operation with range for L_{Vol} (0-5E-06 m3/h) and heat flow Q (0-2000 J/s) for fixed feed rate conditions F_{Vol} = 1 E-06 m3/h, X_F = 0.3, where the Levenberg-Marquardt training algorithm has been applied.

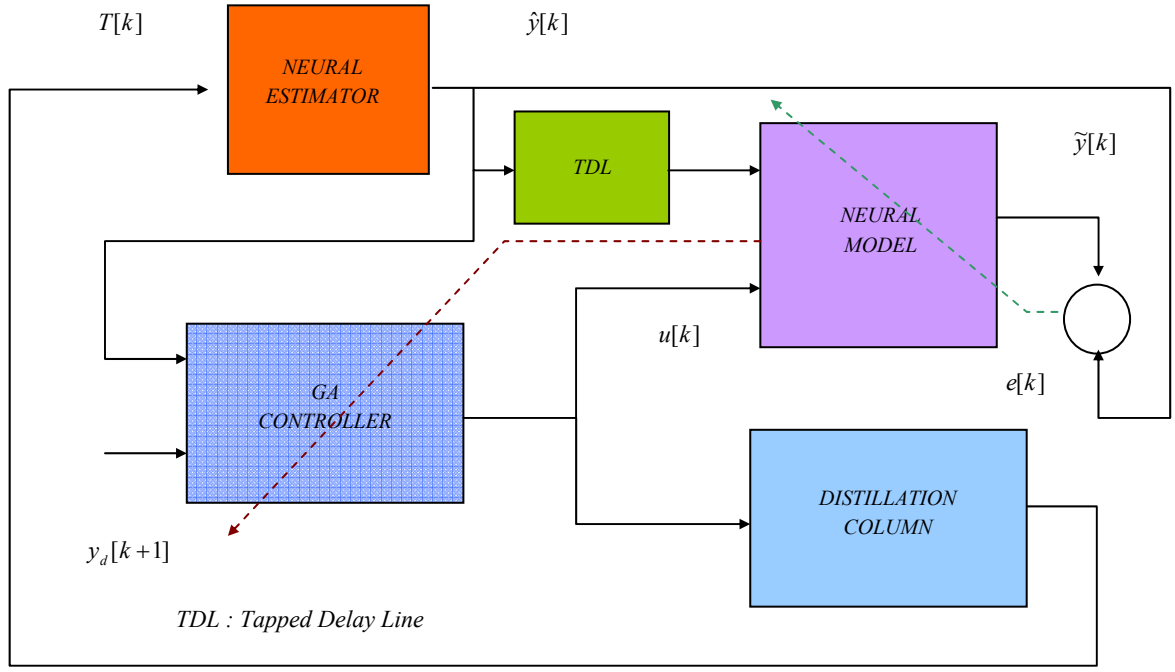


Figure 2: Estimation and control neural network based structure.

An additional temperature data set consisting of 150 data points was used to test the neural predictor afterwards. The error in the training phase is under 0.001% and 0.002% in the validation phase. For training pattern generation we assume an initial steady state for the column after a start-up process.

3.2 Neural Model

Prior to the design of the controller, a neural network has been used as an identification model of the distillation column dynamics. To obtain representative training data, varying feed flows, initial liquid composition values both in the column, boiler and condenser along with input values for the control actions were imposed on the model. The a 3-layer net (4-10-2 units) with vector input $\tilde{y}[k] = f(\hat{y}[k], \hat{y}[k-1], \hat{y}[k-2], u[k])$ and vector output $u[k] = [L_{Vol}[k], Q[k]]$ with $\hat{y}[k]$ regularly spaced covering the same range as defined in the former. As the model's dynamic will be modified with unknown perturbations, this neural network based model will be updated with the real plant response.

3.3 Neural GA Controller

As the estimation of the composition vector $\tilde{y}[k]$ in the next simulation step according the present and previous states of $\hat{y}[k]$ and the input to the system $u[k]$ can be achieved using the neural net, the control problem can be implemented as an optimization problem in which the function to minimize is the difference between the desired output $y_d[k]$ and the estimated one $\tilde{y}[k]$ in the next step. As a result, the optimum control law $u[k]$ is elicited for the distillation control problem.

In order to search for the optimum for the highly non-linear function a genetic algorithm is used with 75 members fixed population, 75 generations and random mutation. If an error under 0.01% is achieved, the algorithm is stopped in order to accelerate the simulations.

4 RESULTS

The aim in the design of the composition neural estimator is to use together with Neural-GA controller for dual composition control of the distillation column. Therefore, the composition

estimator is tested by using the SIMULINK model before it is used for control (Figure 3).

Changes in the reflux and heat flow are determined by the neural network based controller for the column (Figure 4). The performance of the control structure is checked for a 95% to 98% (5% to 7%) pulse change in the distillate (bottom) composition set-point at t = 4600 s together with a 40% to 30% change in feed composition X_F at t = 9050 s, this variable taken as a disturbance (Figure 5). The results obtained demonstrate the potential use of this control strategy in this field.

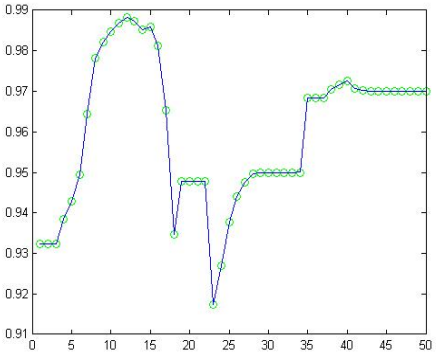


Figure 3: Composition estimation for the neural predictor.

5 CONCLUSIONS

We have proposed a neural network design methodology to dual composition control in a multivariable binary distillation column. A neural network has been employed both for prediction of composition profiles from temperatures and design of optimum control law using a GA search technique, by using a neural model based fitness function. The results obtained point to the potential

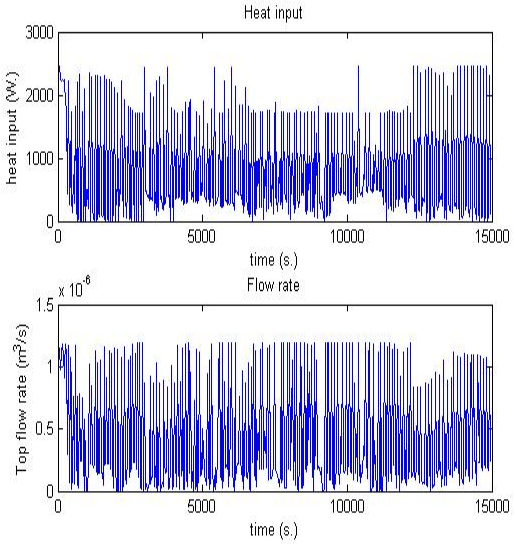


Figure 4: Heat flow and reflux flow rate for a pulse set-point change in top (bottom) product purity and disturbance in X_F.

use of this control strategy in areas of design related to operability and control in process engineering.

Future works are actually directed towards the application of the proposed methodology to a real small scale pilot plant DELTALAB DC-SP (web http//www.isa.uma.es/C4/Control%20Neuroborroso/Document%20Library/index.htm)

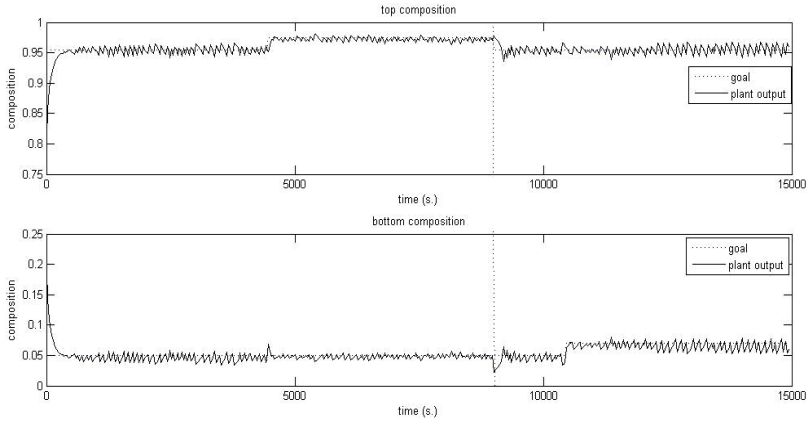


Figure 5: Performance of the Neural-GA controller for a pulse set-point change in top (bottom) product purity and disturbance in X_F

REFERENCES

Bhat N, McAvoy T., 1990. Use of neural nets for dynamic modeling and control of chemical process systems. *Computers and Chemical Eng* , v. 14, n. 4, pp. 573–583.

Bulsari A.,1995. *Neural networks for chemical engineers.* Elsevier, Amsterdam.

Zamprogna, E.; Barolo, M.; Seborg, D.E., 2001. Composition estimations in a middle-vessel batch distillation column using artificial neural networks. *Chemical Engineering Research and Design*, v.79, n.6, pp. 689-696.

Hussain, M. A., 1999. Review of the applications of neural networks in chemical process control. Simulation and on-line implementations, *Artificial Intelligence in Eng*., v. 13, pp. 55-68.

Xiong, Q., Jutan, A., 2002. Grey-box modelling and control of chemical processes", *Chem. Eng. Science*, v. 57, pp. 1027-1039.

Michalewitz, Z., 1992. *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin, Germany.

Fleming, P., Purshouse, R., 2002. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, v. 10, pp. 1223–1241.

Diehl, M., Uslun I., Findeisen, R., 2001. Real-time optimization for large scale processes: Nonlinear predictive control of a high purity distillation column, *On Line Optimization of Large Scale System: State of the Art*, Springer-Verlag, 2001

Fernandez de Canete, J., Gonzalez-Perez, S., Del Saz-Orozco, P., 2007. A development of tools for monitoring and control of multivariable neurocontrolled systems with application to distillation columns*, Proeedings of the EANN 2007 International Conference*, Thesaloniki, Greece, pp. 296-305.

A NOVEL PARTICLE SWARM OPTIMIZATION APPROACH FOR MULTIOBJECTIVE FLEXIBLE JOB SHOP SCHEDULING PROBLEM

Souad Mekni, Besma Fayech Char and Mekki Ksouri
ACS, Ecole Nationale d.Ingnieurs de Tunis, Tunisia
msouadpop@yahoo.fr, {besmafayechchaar, Mekki.ksouri}@insat.rnu.tn

Keywords: Flexible Job Shop Scheduling, Multiobjective Optimization, Particle Swarm Optimization, Smallest Position Value.

Abstract: Because of the intractable nature of the .exible job shop scheduling problem and its importance in both .elds of production management and combinatorial optimization, it is desirable to employ e cient metaheuristics in order to obtain a better solution quality for the problem. In this paper, a novel approach based on the vector evaluated particle swarm optimization and the weighted average ranking is presented to solve .exible job shop scheduling problem (FJSP) with three objectives (i) minimize the makespan, (ii) minimize the total workload of machines and (iii) minimize the workload of critical machine. To convert the continuous position values to the discrete job sequences, we used the heuristic rule the Smallest Position Value (SPV). Experimental results in this work are very encouraging since that relevent solutions were provided in a reasonable computational time.

1 INTRODUCTION

Solving a NP-hard scheduling problem with only one objective is a difficult task. Adding more objectives obviously makes this problem more difficult to solve. In fact, while in single objective optimization the optimal solution is usually clearly defined, this does not hold for multiobjective optimization problems. Instead of a single optimum, there is rather a set of good compromises solutions, generally known as Pareto optimal solutions from which the decision maker will select one. These solutions are optimal in the wider sense that no other solution in the search space is superior when all objectives are considered. Recently, it was recognized that Particle Swarm Optimization (PSO) was well suited to multiobjective optimization mainly because of its fast convergence.

The Particle Swarm Optimization (PSO) is a population based search algorithm developed by Kennedy and Eberhart in 1995 (Kennedy, 1995) (Abraham, 2006, pp. 3-15) (Clerc, 2005) inspired by social behaviour of bird flocking or fish schooling. Unlike Genetic Algorithms (GA), PSO has no evolution operators such as crossover and mutation. In PSO, the population is initialized randomly and the potential solutions, called particles (Hu, 2004) fly through the search space with velocities which are dynamically adjusted according to their historical behaviors. In PSO, each particle is influenced by both the best solution that it has discovered so far and the best particle in its neighbors (local variant of PSO) or in the entire population (global variant of PSO).

Figure 1 shows the general flow chart of PSO. At each time step, the behaviour of a given particle is a compromise between three possible choices:

- to follow its own way
- to go towards its best previous position
- to go towards the best neighbour

This compromise is formalized by equations (1) and (2) and illustrated by figure 2.

$$\begin{aligned} \vec{v_i}(t+1) &= c_1 * \vec{v_i}(t) + c_2 * (\vec{p_i}(t) - \vec{x_i}(t))1 \quad (1) \\ &\quad + c_3 * (\vec{p_g}(t) - \vec{x_i}(t)) \quad (2) \end{aligned}$$

$$\vec{x_i}(t+1) = \vec{x_i}(t) + \vec{v_i}(t+1) \quad (3)$$

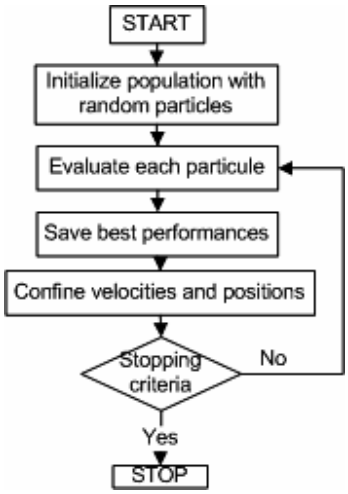


Figure 1: The mapping between particle and FJSP.

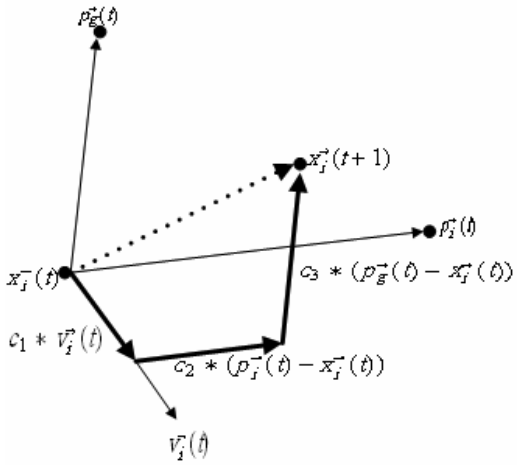


Figure 2: An illustration of particle's move.

with

- $v_i^{\rightarrow}(t)$: velocity of particle i at iteration t
- $p_i^{\rightarrow}(t)$: best previous position of particle i at iteration t
- $p_g^{\rightarrow}(t)$: best neighbour of particle i at iteration t
- $x_i^{\rightarrow}(t)$: position of particle i at iteration t
- c_1, c_2, c_3: positive random numbers. These numbers are social-confidents coefficients

Although PSO is still new in evolutionary computation field, it has been applied to a plethora of problems in science and engineering. Multiobjective optimization problem (MOO) has been one of the most studied application areas of PSO algorithms (Coello, 2002), (Coello, 2004), (Hu, 2002), (Parsopoulos, 2002), (Hu, 2003). Number of approaches have been used and/or designed to manage MOO problems using PSO. A straight forward approach is to convert MOO to a single objective optimization problem. One simple implementation of the conversion is the so-called weighted aggregation approach which sums all the objectives to form a weighted combination (Shi, 2004) (Mendes, 2004). Weights can be either fixed or adapted dynamically during the optimization.

Other approaches combine Pareto dominance with PSO in order to identify Pareto fronts. Most of the research studies developed in this field used two dimensional objectives. It may seem that using only two objectives oversimplifies the problem (Mendes, 2004). In this paper, an application of the particle swarm optimization algorithm to the flexible job shop scheduling with three objectives is reported. The main goal of our research is to design mecanisms to extend PSO such that it can generate solutions of "good quality" either for the individual optimization of criteria or for the compromise between the different objectives.

2 MATHEMATICAL FORMULATION

The flexible job shop scheduling problem was studied in (Chetouane, 1995), (Mesghouni, 1999), (Kacem, 2002), (Dupas, 2004), (Xia, 2005), (Abraham, 2006), (Liu, 2006), (Liu, 2007). FJSP belongs to the *NP*-hard family (Sakarovitch, 1984). It presents two difficulties. The first one is the assignment of each operation to a machine, and the second is the scheduling of this set of operations in order to optimize our criteria. The result of a scheduling algorithm must be a schedule that contains a start time and a resource assignment to each operation.

The data, constraints and objectives of our problem are as follows:

2.1 Data

- M represents a set of m machines. A machine is called M_k $(k = 1, ..., m)$, each M_k has a load called W_k.
- N represents a set of n jobs. A job is called J_i $(i = 1, ..., n)$, each job has a linear sequence of n_i operations.
- $O_{i,j}$ represents the operation number j of the job number i. The realization of each operation $O_{i,j}$ requires a machine M_k and a processing time

$p_{i,j,k}$. The starting time of $O_{i,j}$ is $t_{i,j}$ and the ending time is $t_{fi,j}$.

2.2 Constraints

- Machines are independent of one another.
- Each machine can perform operations one after another.
- Each machine is available during the scheduling.
- A started operation runs to completion.
- Jobs are independent of each another.

In our work, we suppose that:

- Machines are available since the date $t = 0$.
- Each job j_i can start at the date $t = 0$.
- The total number of operations to perform is greater than the number of machines.

2.3 Criteria

We have to minimize $Cr1$, $Cr2$ and $Cr3$:

- The makespan:

$$Cr1 = \max_{1\leq i\leq n}(\max_{1\leq j\leq n_i}(t_{fi,j}))$$

- The total workload of machines:

$$Cr2 = \sum_{1\leq k\leq m}(W_k)$$

- The workload of the most loaded machine:

$$Cr3 = \max_{1\leq k\leq m}(W_k)$$

These criteria are often conflicting. In fact, balance resource usage by minimizing the utilization of bottleneck equipement can be antagonistic with the minimization of the total time of production.

2.4 Lower Bounds

Lower bounds are usually used to measure the quality of solutions found. For our work, we use lower bounds proposed in (Dupas, 2004):

- $BCr1$: (lower bound for $Cr1$)

$$BCr1 = \max_i(\sum_j \min_k(p_{i,j,k}))$$

- $BCr2$: (lower bound for $Cr2$)

$$BCr2 = \sum_{i,j}\min_k(p_{i,j,k})$$

- $BCr3$: (lower bound for $Cr3$)

$$BCr3 = \lceil BCr2/m \rceil$$

3 PSO FOR FJSP

3.1 Particle Representation and Initial Swarm Generation

One of the key issues when designing the PSO algorithm lies on its solution representation which directly affects its feasibility and performance. In this paper, an operation-based representation is used. For the (m machines, n jobs, O operations) FJSP, each particle contains O number of dimensions corresponding to O operations and has a continuous set of values for its dimensions which represents particle's positions. The Smallest Position Value (Tasgetiren, 2004) (Tasgetiren, 2006), the SPV rule is used to find the permutation of operations and a randomly generated number is used to find the machine to which a task is assigned to during the course of PSO. Figure 3 illustrates the solution representation of a particle corresponding to FJSP described in table 1 and table 2. The smallest component of the particle's position is $-2,25$ which corresponds to the operation number 6 of job number 2. Thus, job 2 is scheduled first. The second smallest component of the particle's position is $-0,99$ which corresponds to the operation number 2 of job number 1. Therefore, job 1 is the second job in the ordering, etc.

Table 1: Processing time of operations of a (3 Jobs, 5 Machines) problem.

	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_3$	$\mathbf{M}_4$	$\mathbf{M}_5$
$O_{1,1}$	1	9	3	7	5
$O_{1,2}$	3	5	2	6	4
$O_{1,3}$	6	7	1	4	3
$O_{2,1}$	1	4	5	3	8
$O_{2,2}$	2	8	4	9	3
$O_{2,3}$	9	5	1	2	4
$O_{3,1}$	1	8	9	3	2
$O_{3,2}$	5	9	2	4	3

Table 2: The operating sequences of jobs of a (3 Jobs, 5 Machines) problem.

J_1	$O_{1,1}$	$O_{1,2}$	$O_{1,3}$
J_2	$O_{2,1}$	$O_{2,2}$	$O_{2,3}$
J_3	$O_{3,1}$	$O_{3,2}$	

The PSO randomly generates an initial swarm of S particles, where S is the swarm size. These particle vectors will be iteratively updated based on collective experiences in order to enhance their solution quality.

Jobs	J_1			J_2			J_3	
Operations	$O_{1,1}$	$O_{1,2}$	$O_{1,3}$	$O_{2,1}$	$O_{2,2}$	$O_{2,3}$	$O_{3,1}$	$O_{3,2}$
Particle position	1,8	−0,99	3,01	0,72	−0,45	−2,25	5,3	4,8
Sorted positions	6	2	5	4	1	3	8	7
Sequence of jobs	2	1	2	2	1	1	3	3
Sorted operations	$O_{2,1}$	$O_{1,1}$	$O_{2,2}$	$O_{2,3}$	$O_{1,2}$	$O_{1,3}$	$O_{3,1}$	$O_{3,2}$
Random numbers	2	3,7	5,3	1,4	5,1	5	4	4,6
	↓	↓	↓	↓	↓	↓	↓	↓
Machines	(M_2,t_1)	(M_3,t_2)	(M_5,t_3)	(M_1,t_4)	(M_5,t_5)	(M_5,t_6)	(M_4,t_7)	(M_4,t_8)

Figure 3: The mapping between particle and FJSP.

3.2 Our Approach

Our approach is a novel proposal to solve multiobjective optimization problems using PSO. It is inspired by The Vector Evaluated Particle Swarm Optimization (VEPSO)(Parsopoulos, 2002b) algorithm wich incorporates ideas from the Vector Evaluated Genetic Algorithm (VEGA) (Shaffer, 1985).

Our approach is based on the use of Weighted Average Ranking (WAR) (Collette, 2002) and a subdivision of decision variable space into $(k+1)$ subswarms (k: is the number of criteria). Each subswarm i (i between 1 and k) is exclusively evaluated with the objective function number i, but, information coming from other sub-swarm(s) especially from the sub-swarm number $(k+1)$ is used to influence its motion in the search space. The execution of the flight of each sub-swarm can be seen as an entire PSO process (with the difference that it will optimize only a part of the search space and not the entire search space). The sub-swarm $(k+1)$ looks for the solutions of compromise between the k studied criteria. It generates the leaders set among the particle swarm set by using the Weighted Average Ranking. Leaders of other sub-swarms can migrate to the sub-swarm $(k+1)$ until a number of iterations is reached in order to variate the selection pressure. The procedure of exchanging information among sub-swarms can lead to Pareto optimal points.

Stages of the algorithm described in figure 1 are repeated until a certain prefixed number of iterations is reached.

4 PERFORMANCE MEASURES

Different instances of the present problem have been chosen to test our approach, in order to ensure a certain diversity. These instances present a number of operations between 8 and 56 (the number of jobs is between 3 and 15) and a number of resources between 4 and 10 machines. The studied problem nature is varied enough according to the performance of resources, their flexibility and the number of the precedence constraints. So, cases of parallel machine problems, where all the machines have the same performance, have been also tested. We also studied total and partial flexibility cases when machines presented variable performances. As results to the simulations, some findings can be pulled:

- Most particular problems have been solved in an optimal manner (case of problems having parallel machines).
- The problems with parallel machines are easier to solve than the problems having machines with variable performance.
- The found solutions are generally of a good quality. Is is noted while comparing them with the existing approaches in the literature and also while comparing obtained values of the criteria with the computed lower bounds. As an illustration, we choose to present the following instance: we consider the problem described in table 3 (10 jobs, 30 operations, 10 machines). The computation of the different lower bounds gives the following values: $BCr1 = 7$, $BCr2 = 41$, $BCr3 = 4$. This example has been already processed in the literature by many methods: temporal decomposition (Chetouane, 1995), classic GAs (Mesghouni, 1999), approach by localization and approach by localization and controlled EAs and approach by hybridizating particle swarm optimization and simulated annealing (Xia, 2005). The schedule obtained in these cases is characterized by the following values presented in figure 4.

	Cr1	*Cr2*	*Cr3*
Temporal decomposition	16	59	16
Classic GA	7	53	7
Approach by localization	8	46	6
AL+CGA	7	45	5
PSO+SA	7	44	6
Our approach	7	44	6

Figure 4: Solutions in the literature of (10J, 10M).

Table 3: Matrix of processing times of FJSP (10J,10M).

		M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
J_1	$O_{1,1}$	1	4	6	9	3	5	2	8	9	5
	$O_{1,2}$	4	1	1	3	4	8	10	4	11	4
	$O_{1,3}$	3	2	5	1	5	6	9	5	10	3
J_2	$O_{2,1}$	2	10	4	5	9	8	4	15	8	4
	$O_{2,2}$	4	8	7	1	9	6	1	10	7	1
	$O_{2,3}$	6	11	2	7	5	3	5	14	9	2
J_3	$O_{3,1}$	8	5	8	9	4	3	5	3	8	1
	$O_{3,2}$	9	3	6	1	2	6	4	1	7	2
	$O_{3,3}$	7	1	8	5	4	9	1	2	3	4
J_4	$O_{4,1}$	5	10	6	4	9	5	1	7	1	6
	$O_{4,2}$	4	2	3	8	7	4	6	9	8	4
	$O_{4,3}$	7	3	12	1	6	5	8	3	5	2
J_5	$O_{5,1}$	7	10	4	5	6	3	5	15	2	6
	$O_{5,2}$	5	6	3	9	8	2	8	6	1	7
	$O_{5,3}$	6	1	4	1	10	4	3	11	13	9
J_6	$O_{6,1}$	8	9	10	8	4	2	7	8	3	10
	$O_{6,2}$	7	3	12	5	4	3	6	9	2	15
	$O_{6,3}$	4	7	3	6	3	4	1	5	1	11
J_7	$O_{7,1}$	1	7	8	3	4	9	4	13	10	7
	$O_{7,2}$	3	8	1	2	3	6	11	2	13	3
	$O_{7,3}$	5	4	2	1	2	1	8	14	5	7
J_8	$O_{8,1}$	5	7	11	3	2	9	8	5	12	8
	$O_{8,2}$	8	3	10	7	5	13	4	6	8	4
	$O_{8,3}$	6	2	13	5	4	3	5	7	9	5
J_9	$O_{9,1}$	3	9	1	3	8	1	6	7	5	4
	$O_{9,2}$	4	6	2	5	7	3	1	9	6	7
	$O_{9,3}$	8	5	4	8	6	1	2	3	10	12
J_{10}	$O_{10,1}$	4	3	1	6	7	1	2	6	20	6
	$O_{10,2}$	3	1	8	1	9	4	1	4	17	15
	$O_{10,3}$	9	2	4	2	3	5	2	4	10	23

5 CONCLUSIONS

This paper presents a novel approach using parti- cle swarm optimization to solve the multicriteria .exible job shop scheduling with total or partial .exibility. It is based on the vector evaluated particle swarm optimization and the weighted av- erage ranking.

Our work, resulted in to the development of a generic method to resolve multiobjective opti- mization. It provides relevant solutions for the individual optimization of criteria or for the com- promise between the dierent objectives. Future research will cover an investigation on the eects of diversity control in the search performances of multiobjective particle swarm optimization.

REFERENCES

Abraham A., Guo. H, Liu. H., Swarm Intelligence: Foundations, Perspectives and Applications. Studies in Computational Intelligence (SCI) 26;pp. 3-25; 2006.

Abraham A., Lui H. and Ghang T. G., Variable neighborhood Particle Swarm Optimization Algorithm. In GECCO'06 Seattle, Washington, USA, July 8-12, 2006.

Chetouane F., *Ordonnancement d'atelier tches gnralises, perturbations, ractivit*. Rapport DEA de l'institut national polytechnique de Grenoble, 1995.

Clerc. M., *L'optimisation par essaims particulaires, versions paramtriques et adaptatives*. Edition Herms, Lavoisier, 2005.

Collette Y., Siarry P., *Optimisation Multiobjectif*. Edition Eyrolles, Paris, 2002.

Coello C. A., Lechuga M. S., MOPSO:*A Proposal for Multiple Objective Particle Swarm Optimization*. IEEE Congress on Evolutionary Computation, Honolulu, Hawaii USA, 2002.

Coello C. A., Pulido G. T., Lechuga M. S., *Handling Multiple Objectives with Particle Swarm Optimization*. IEEE Transactions on Evolutionary Computation, IEEE, Piscataway, NJ, 8(3) 256-279, 2004.

Dupas R., *Amlioration de Performance des Systmes de Production: Apport des Algorithmes volutionnistes aux Problmes d'Ordonnancement Cycliques et flexibles*. Habilitation Diriger des Recherches, Universit d'Artois, 2004.

Hu X., Shi Y., Eberhart R. C., *Recent advances in Particle Swarm*. In Proceedings of Congress on Evolutionary Computation (CEC), Portland, Oregon, 90-97, 2004.

Hu X., Eberhart R. C., *Multiobjective Optimization using Dynamic Neighborhood Particle Swarm Optimization*. Proceeding of the 2002 Congress on Evolutionary Computation, Honolulu, Hawaii, May 12-17, 2002.

Hu X., Shi Y., Eberhart R.C., *Particle Swarm with extended Memory for Multiobjective Optimization*. Proc. of 2003 IEEE Swarm Intelligence Symposium, pp 193-197. Indanapolis, Indiana, USA, IEEE Service Center, April 2003.

Kacem I., Hammadi S., Borne P., *Approach by Localization and Multiobjective Evolutionary Optimization for flexible Job Shop Scheduling Problems*. IEEE Transactions on Systems, man and cybernetics-Part c/ Applications and reviews vol 32, N.1 February , 2002.

Kacem I., Hammadi S., Borne P., *Pareto-optimality Approach for Flexible job-shop Scheduling Problems: hybridization of evolutionary algorithms and fuzzy logic*. IEEE Transactions on Systems, man and cybernetics-Part c/ Applications and reviews vol 32, N.1 February , 2002.

Kennedy J., Eberhart R. C., *Particle Swarm Optimization*. IEEE International Conference on Neural Networks(Perth, Australia). IEEE Service Center, Piscataway, NJ, IV, pp. 1942-1948; 1995.

Liu H., Abraham A., Choi O., Moon S. H., *Variable Neighborhood Particle Swarm Optimization for Multi-objective Flexible Job-shop Scheduling Problems*. SEAL 2006, 197-204, 2006.

Liu H., Abraham A., Grosan C., Li N., *A novel Variable Neighborhood Particle Swarm Optimization for Multi-objective Flexible Job-shop Scheduling Problems*. ICDIM.07 Lyon, France, October 28-31, 2007.

Mendes R., *Population Topologies and their Influence in Particle Swarm Performance*. Thse, 2004.

Mesghouni K., *Application des Algorithmes volutionnistes dans les Problmes d'Optimisation en Ordonnancement de la Production*. Thse, Universit des Sciences et Technologies de Lille; 1999.

Parsopoulos, K. E., Verhatis, M. N., *Recent Approaches to Global Optimization Problems through Particle Swarm Optimization*. Natural Computing 1: 235-306, 2002.

Parsopoulos, K. E., Verhatis, M. N.(2002b), *Particle Swarm Optimization Method in Multiobjective Problems*. Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002), pp. 603-607, 2002.

Sakarovitch M., *Optimisation Combinatoire. Mthodes Mathmatiques et Algorithmiques*. Hermann, Editeurs des sciences et des arts, Paris, 1984.

Shaffer D., *Multiple Objective Optimization with Vector Evaluated Genetic Algorithm*. In genetic Algorithm and their Applications: Proceedings of the First International Conference on Genetic Algorithm, pages 93-100, 1985.

Shi Y., *Particle Swarm Optimization*. IEEE Neural Networks Society, February 2004. PSO Tutorial, http://www.swarmintelligence.org/tutorials.php

Tasgetiren M. F., Sevkli M., Liang Y. C., Gencyilmaz G., *Particle Swarm Optimization Algorithm for single Machine Total Tardiness Problem*. IEEE 2004.

Tasgetiren M. F., Sevkli M., Liang Y. C., Yenisey M. M., *Particle Swarm Optimization and Diferential Evolution Algorithms for Job Shop Scheduling Problem*. International Journal of Operational Research, vol.3, N.2, Oct 2006 pp.120-135.

Xia W., Wu Z., *An effective Hybrid Optimization Approach for Multi-objective flexible Job-shop Scheduling Problems*. Computers and Industrial Engineering, 48:409-425, 2005.

TWO-SIDED ASSEMBLY LINE
Estimation of Final Results

Waldemar Grzechca
Institute of Automatic Control, The Silesian University of Technology, ul.Akademicka 16, 44-100 Gliwice, Poland
waldemar.grzechca@polsl.pl

Keywords: Assembly line balancing, Two-sided structure, Line time, Line efficiency, Smoothness index.

Abstract: The paper considers simple assembly line balancing problem and two-sided assembly line structure. In the last four decades a large variety of heuristic and exact solutions procedures have been proposed to balance one-sided assembly line in the literature. Some heuristic were given to balance two-sided lines, too. Some measures of solution quality have appeared in line balancing literature: balance delay (BD), line efficiency (LE), line time (LT) and smoothness index (SI). These measures are very important for estimation the balance solution quality. Author of this paper modified and discussed the line time and smoothness for two-sided assembly line. Some problems, which appeared during evaluations, are mentioned.

1 INTRODUCTION

The manufacturing assembly line was first introduced by Henry Ford in the early 1900's. It was designed to be an efficient, highly productive way of manufacturing a particular product. The basic assembly line consists of a set of workstations arranged in a linear fashion, with each station connected by a material handling device. The basic movement of material through an assembly line begins with a part being fed into the first station at a predetermined feed rate. A station is considered any point on the assembly line in which a task is performed on the part. These tasks can be performed by machinery, robots, and/or human operators. Once the part enters a station, a task is then performed on the part, and the part is fed to the next operation. The time it takes to complete a task at each operation is known as the process time (Sury, 1971). The cycle time of an assembly line is predetermined by a desired production rate. This production rate is set so that the desired amount of end product is produced within a certain time period (Baybars, 1986). For instance, the production rate might be set at 480 parts per day. Assuming an eight-hour shift, this translates into a requirement of 60 parts per hour (1 part per minute) being produced by the assembly line. In order for the assembly line to maintain a certain production rate, the sum of the processing times at each station must not exceed the stations' cycle time (Fonseca et. al, 2005). If the sum of the processing times within a station is less than the cycle time, idle time is said to be present at that station (Erel, Erdal and Sarin, 1998). One of the main issues concerning the development of an assembly line is how to arrange the tasks to be performed. This arrangement may be somewhat subjective, but has to be dictated by implied rules set forth by the production sequence (Kao, 1976). For the manufacturing of any item, there are some sequences of tasks that must be followed. The assembly line balancing problem (ALBP) originated with the invention of the assembly line. Helgeson et. al (Helgeson and Birnie, 1961) were the first to propose the ALBP, and Salveson (Salveson, 1955) was the first to publish the problem in its mathematical form. However, during the first forty years of the assembly line's existence, only trial-and-error methods were used to balance the lines (Erel, Erdal and Sarin, 1998). Since then, there have been numerous methods developed to solve the different forms of the ALBP. Salveson (Salveson, 1955) provided the first mathematical attempt by solving the problem as a linear program. Gutjahr and Nemhauser (Gutjahr and Nemhauser, 1964) showed that the ALBP problem falls into the class of NP-hard combinatorial optimization problems. This means that an optimal solution is not guaranteed for problems of significant size. Therefore, heuristic methods have become the most popular techniques for solving the problem.

2 TWO-SIDED ASSEMBLY LINE

Two-sided assembly lines are typically found in producing large-sized products, such as trucks and buses. Assembling these products is in some respects different from assembling small products. Some assembly operations prefer to be performed at one of the two sides (Bartholdi, 1993).

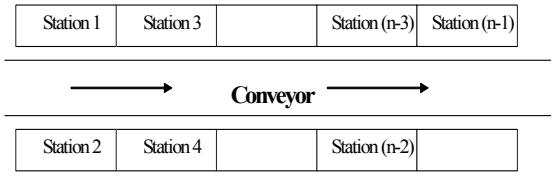


Figure 1: Two-sided assembly line.

Let us consider, for example, a truck assembly line. Installing a gas tank, air filter, and toolbox can be more easily achieved at the left-hand side of the line, whereas mounting a battery, air tank, and muffler prefers the right-hand side. Assembling an axle, propeller shaft, and radiator does not have any preference in their operation directions so that they can be done at any side of the line. The consideration of the preferred operation directions is important since it can greatly influence the productivity of the line, in particular when assigning tasks, laying out facilities, and placing tools and fixtures in a two-sided assembly line (Kim et. al, 2001). A two-sided assembly line in practice can provide several substantial advantages over a one-sided assembly line (Bartholdi, 1993). These include the following: (1) it can shorten the line length, which means that fewer workers are required, (2) it thus can reduce the amount of throughput time, (3) it can also benefit from lowered cost of tools and fixtures since they can be shared by both sides of a mated-station, and (4) it can reduce material handling, workers movement and set-up time, which otherwise may not be easily eliminated. These advantages give a good reason for utilizing two-sided lines for assembling large-sized products.

A line balancing problem is usually represented by a precedence diagram as illustrated in Figure 2. A circle indicates a task, and an arc linking two tasks represents the precedence relation between the tasks. Each task is associated with a label of (t_i, d), where t_i is the task processing time and d (=L, R or E) is the preferred operation direction. L and R, respectively, indicate that the task should be assigned to a left- and a right-side station. A task associated with E can be performed at either side of the line.

While balancing assembly lines, it is generally needed to take account of the features specific to the lines. In a one-sided assembly line, if precedence relations are considered appropriately, all the tasks assigned to a station can be carried out continuously without any interruption. However, in a two-sided assembly line, some tasks assigned to a station can be delayed by the tasks assigned to its companion (Bartholdi, 1993). In other words, idle time is sometimes unavoidable even between tasks assigned to the same station. Consider, for example, task j and its immediate predecessor i. Suppose that j is assigned to a station and i to its companion station. Task j cannot be started until task i is completed. Therefore, balancing such a two-sided assembly line, unlike a one-sided assembly line, needs to consider the sequence-dependent finish time of tasks.

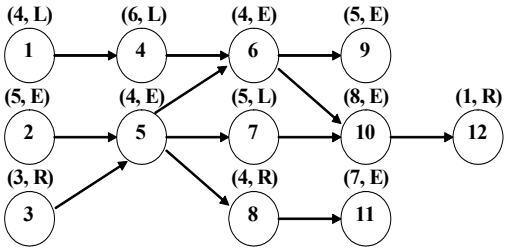


Figure 2: Precedence graph (cycle time =16).

This notion of sequence dependency further influences the treatment of cycle time constraint. Every task assigned to a station must be able to be completed within a predetermined cycle time. In a one-sided assembly line, this can readily be achieved by checking the total operation time of tasks assigned to a station. Therefore, a task not violating any precedence constraints can be simply added to the station if the resulting total amount of operation time does not exceed the cycle time. However, in a two-sided assembly line, due to the above sequence-dependent delay of tasks, the cycle time constraint should be more carefully examined. The amount of time required to perform tasks allocated to a station is determined by the task sequences in both sides of the mated-station as well as their operation time. It should be mentioned that two-sided assembly line is a special case of single assembly line. Therefore it is possible to use some procedures and measurements, which were for simple assembly line developed.

3 HEURISTIC APPROACH

3.1 Grouping Tasks

A task group consists of a considered task i and all of its predecessors. Such groups are generated for every un–assigned task. As mentioned earlier, balancing a two–sided assembly line needs to additionally consider operation directions and sequence dependency of tasks, while creating new groups (Kim et. al, 2005).

While forming initial groups IG(i), the operation direction is being checked all the time. It's disallowed for a group to contain tasks with preferred operation direction from opposite sides. But, if each task in initial group is E – task, the group can be allocated to any side. In order to determine the operation directions for such groups, the rules (direction rules DR) are applied:

DR 1. Set the operation direction to the side where tasks can be started earlier.

DR 2. The start time at both sides is the same, set the operation direction to the side where it's expected to carry out a less amount of tasks (total operation time of unassigned L or R tasks).

Generally, tasks resulting from "repeatability test" are treated as starting ones. But there is exception in form of first iteration, where procedure starts from searching tasks (initial tasks IT), which are the first ones in precedence relation. After the first step in the first iteration we get:

IG (1) = {1}, Time{IG (1)} = 2, Side{IG (1)} = 'L'

IG (2) = {2}, Time{IG (2)} = 5, Side{IG (2)} = 'E'

IG (3) = {3}, Time{IG (3)} = 3, Side{IG (3)} = 'R

where:
Time{IG(i)} – total processing time of i^{th} initial group,
Side{IG(i)} – preference side of i^{th} initial group.

To those who are considered to be the first, the next tasks will be added, (these ones which fulfil precedence constraints).
Whenever new tasks are inserted to the group i, the direction, cycle time and number of immediate predecessors are checked. If there are more predecessors than one, the creation of initial group j comes to the end.

First iteration – second step

IG (1) = {1, 4, 6}, Time{IG (1)} = 8, Side{IG (1)} = 'L'

IG (2) = {2, 5}, Time{IG (2)} = 9 , Side{IG (2)} = 'E'

IG (3) = {3, 5} , Time{IG (3)} = 7 , Side{IG (3)} = 'R'

When set of initial groups is created, the last elements from those groups are tested for repeatability. If last element in set of initial groups IG will occur more than once (groups pointed by arrows), the groups are intended to be joined – if total processing time (summary time of considered groups) is less or equal to cycle time. Otherwise, these elements are deleted.

In case of occurring only once, the last member is being checked if its predecessors are not contained in Final set FS. If not, it's removed as well. So far, FS is empty.
First iteration – third step

IG (1) = {1, 4}, Time{IG (1)} = 4, Side{IG (1)} = 'L'

IG (2) = {2, 3, 5}, Time{IG (2)} = 12, Side{IG (2)} = 'R'

Whenever two or more initial groups are joined together, or when initial group is connected with those one coming from Final set – the "double task" is added to initial tasks needed for the next iteration. In the end of each iteration, created initial groups are copied to FS.
First iteration – fourth step

FS = { (1, 4); (2, 3, 5) },

Side{FS (1)} = 'L', Side{FS (2)} = 'R'

Time {FS(2)} = 12, Time {FS(1)} = 14,

IT = {5}.

In the second iteration, second step, we may notice that predecessor of last task coming from IG(1) is included in Final Set, FS(2). The situation results in connecting both groups under holding additional conditions:

Side{IG(1)} = Side{FS(2)},

Time + time < cycle.

After all, there is no more IT tasks, hence, preliminary process of creating final set is terminated.
The presented method for finding task groups is to be summarized in simplified algorithm form. Let U denote to be the set of un – assigned tasks yet and

IG_i be a task group consisting of task i and all its predecessors (excluded from U set).

STEP 1. If U = empty, go to step 5, otherwise, assign starting task from U.
STEP 2. Identify IG_i . Check if it contains tasks with both left and right preference operation direction, then remove task i.
STEP 3. Assign operation direction Side{ IG_i } of group IG_i . If IG_i has R-task (L-task), set the operation direction to right (left). Otherwise, apply so called direction rules DR.
STEP 4. If the last task i in IG_i is completed within cycle time, the IG_i is added to Final set of candidates FS(i). Otherwise, exclude task i from IG_i and go to step 1.
STEP 5. For every task group in FS(i), remove it from FS if it is contained within another task group of FS.
The resulting task groups become candidates for the mated-station.
FS = {(1,4), (2,3,5,8)}.

3.2 Groups Assignment

The candidates are produced by procedures presented in the previous section, which claim to not violate precedence, operation direction restrictions, and what's more it exerts on groups to be completed within preliminary determined cycle time. Though, all of candidates may be assigned equally, the only one group may be chosen. Which group it will be – for this purpose the rules helpful in making decision, will be defined and explained below:

AR 1. Choose the task group FS(i) that may start at the earliest time.

AR 2. Choose the task group FS(i) that involves the minimum delay.

AR 3. Choose the task group FS(i) that has the maximum processing time.

In theory, for better understanding, we will consider a left and right side of mated – station, with some tasks already allocated to both sides. In order to achieve well balanced station, the AR 1 is applied, cause the unbalanced station is stated as the one which would probably involve more delay in future assignment. This is the reason, why minimization number of stations is not the only goal, there are also indirect ones, such as reduction of unavoidable delay. This rule gives higher priority to the station, where less tasks are allocated. If ties occurs, the AR 2 is executed, which chooses the group with the least amount of delay among the considered ones. This rule may also result in tie. The last one, points at relating work within individual station group by choosing group of task with highest processing time. For the third rule the tie situation is impossible to obtain, because of random selection of tasks. The implementation of above rules is strict and easy except the second one. Shortly speaking, second rule is based on the test, which checks each task consecutively, coming from candidates group FS(i) – in order to see if one of its predecessors have already been allocated to station. If it has, the difference between starting time of considered task and finished time of its predecessor allocated to companion station is calculated. The result should be positive, otherwise time delay occurs.

3.3 Final Procedures

Having rules for initial grouping and assigning tasks described in previous sections, we may proceed to formulate formal procedure of solving two – sided assembly line balancing problem (Kim et. al, 2005).
Let us denote companion stations as *j* and *j',*
D(i) – the amount of delay,
Time(i) – total processing time (Time{FS(i)}),
S(j) – start time at station j,
STEP 1. Set up j = 1, j' = j + 1, S(j) = S(j') = 0, U – the set of tasks to be assigned.
STEP 2. Start procedure of group creating (3.2), which identifies
FS = {FS(1), FS(2), …, FS(n)}. If FS = ∅, go to step 6.
STEP 3. For every FS(i), i = 1,2, … , n – compute D(i) and Time(i).
STEP 4. Identify one task group FS(i), using AR rules in Section 3.3
STEP 5. Assign FS(i) to a station j (j') according to its operation direction, and update S(j) = S(j) + Time(i) + D(i). U = U – {FS(i)}, and go to STEP 2.
STEP 6. If U ≠ ∅, set j = j' + 1, j' = j + 1, S(j) = S(j') = 0, and go to STEP 2, Otherwise, stop the procedure.

4 MEASURES OF FINAL RESULTS OF ASSEMBLY LINE BALANCING PROBLEM

Some measures of solution quality have appeared in line balancing problem. Below are presented three of them (Scholl, 1998).

Line efficiency (LE) shows the percentage utilization of the line. It is expressed as ratio of total station time to the cycle time multiplied by the number of workstations:

$$LE = \frac{\sum_{i=1}^{K} ST_i}{c \cdot K} \cdot 100\% \quad (1)$$

where:

K - total number of workstations,
c - cycle time.

Smoothness index (SI) describes relative smoothness for a given assembly line balance. Perfect balance is indicated by smoothness index 0. This index is calculated in the following manner:

$$SI = \sqrt{\sum_{i=1}^{K} (ST_{max} - ST_i)^2} \quad (2)$$

where:

ST_{max} = maximum station time (in most cases cycle time),
ST_i = station time of station i.

Time of the line (LT) describes the period of time which is need for the product to be completed on an assembly line:

$$LT = c \cdot (K - 1) + T_K \quad (3)$$

where:

c - cycle time,
K -total number of workstations.

5 NUMERICAL RESULTS

The results of proposed procedure for the example from Figure 2 are given in a Gantt chart – Figure 3. Before presenting performance measures for current example, it would be like to stress difference in estimation of line time form, resulting from restrictions of parallel stations. In two – sided line method within one mated-station, tasks are intended to perform its operations at the same time, as it is shown in example in Figure.3, where tasks 7, 11 respectively are processed simultaneously on single station 3 and 4, in contrary to one – sided heuristic methods. Hence, modification has to be introduced to that particular parameter which is the consequence of parallelism. Having two mated-stations from Figure 3, the line time LT is not 3*16 + 13, as it was in original expression. We must treat those stations as two double ones (mated-stations), rather than individual ones S_k. Accepting this line of reasoning, new formula is presented below:

$$LT = c \cdot (Km - 1) + Max\{t(S_K), t(S_{K-1})\} \quad (4)$$

where:

Km – number of mated-stations
K – number of assigned single stations
$t(S_K)$ – processing time of the last single station

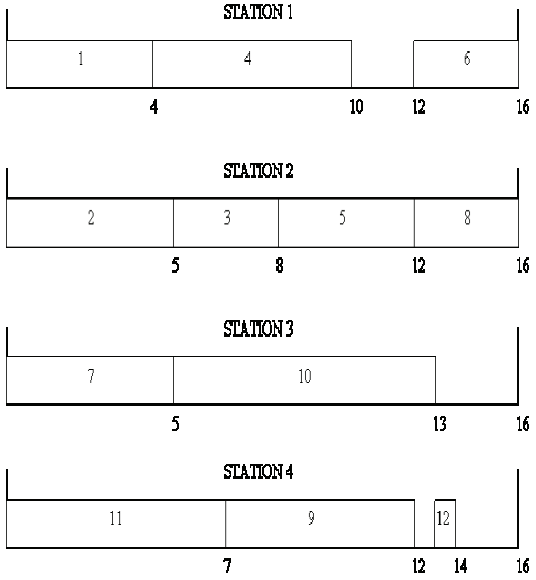


Figure 3: Results for the example problem.

As far as smoothness index and line efficiency are concerned, its estimation, on contrary to LT, is performed without any change to original version. These criterions simply refer to each individual station, despite of parallel character of the method.
But for more detailed information about the balance of right or left side of the assembly line additional measures will be proposed:
Smoothness index of the left side

$$SI_L = \sqrt{\sum_{i=1}^{K} (ST_{maxL} - ST_{iL})^2} \quad (5)$$

where:
SI_L- smoothness index of the left side of two-sided line
ST_{maxL}- maximum of duration time of left allocated stations
ST_{iL}- duration time of i-th left allocated station

Smoothness index of the right side

$$SI_R = \sqrt{\sum_{i=1}^{K} (ST_{maxR} - ST_{iR})^2} \qquad (6)$$

where:
SI_R- smoothness index of the right side of two-sided line
ST_{maxR}- maximum of duration time of right allocated stations
ST_{iR}- duration time of i-th right allocated station

Table 1: Numerical results.

Name	Value
LE	84,38%
LT	30
SI	4,69
SI_R	2
SI_L	3

The numerical results of different measures in Table 1 are given. The value of line efficiency is acceptable, smoothness indexes of the right and left side of the line show which part of the assembly line is better balanced. The smoothness index SI informs about balance of the whole line. It is possible to compare the two-sided line balance with single assembly line balance and to consider the influence of position restrictions (L,R or E).

Next it will be consider a small example presented in Figure 4.

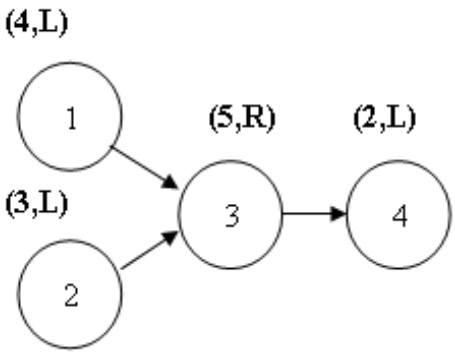


Figure 4: Precedence graph (4 tasks, c=10).

In this point, it's worth to mention about a special case, when mated-station includes instead of two stations, just one. Such a situation takes place, where one station is loaded to a certain point that not allows for assigning any more tasks for this part of the line. As the result, one station stays empty. Balance of this case is presented in Figure 5.

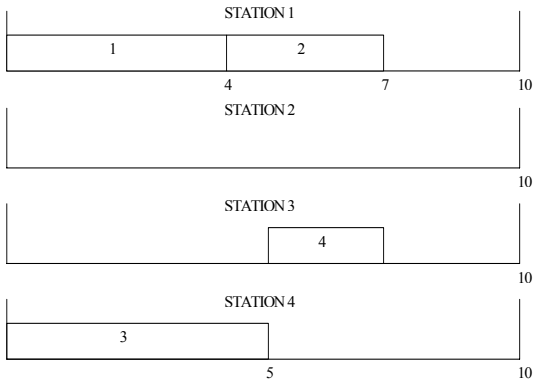


Figure 5: Balance of two-sided line (N=4, c=10).

In this case we got an assembly line which is a structure of incomplete two-sided assembly line. It is possible to estimate the balanced line in two ways: as a single line with parallel stations or incomplete two-sided line.

In the first case we obtain:

K = 3
LE = 46,67%
SI = 9,9

Considering this case as two-sided line we get:

K = 4
LE = 35%
SI_R = 11,18
SI_L = 8,54
SI = 14,07

As we can see there are some differences in final measurements of the balanced line. The reason is that using heuristic methods we design two-sided assembly line. These kinds of heuristics are very sensitive to cycle time value. Some final balances for different value of cycle time for an example from Figure 2 in Table 2 are shown.

Table 2: Final results of different measures (c = var).

c	K	LT	SI	LE
14	6	37	15,81	66,67%
15	6	39	17,66	62,22%
16	4	30	4,69	84,38%
17	6	43	22,05	54,90%
18	4	32	4,69	77,70%

6 CONCLUSIONS

Two-sided assembly lines become more popular in last time. Therefore it is obvious to consider this structure using different methods. In this paper a heuristic approach was discussed. Two-sided assembly line structure is very sensitive to changes of cycle time values. It is possible very often to get incomplete structure of the two-sided assembly line (some stations are missing) in final result. We can use different measures for comparing the solutions (line time, line efficiency, smoothness index). Author proposes additionally two measures: smoothness index of the left side (SI_L) and smoothness index of the right side (SI_R) of the two-sided assembly line structure. These measurements allow to get more knowledge about allocation of the tasks and about the balance on both sides.

This research was supported by grant of Ministry of Science and Higher Education 3T11Ao2229 in 2005-2008.

REFERENCES

Bartholdi J.J., 1993. Balancing two-sided assembly lines: a case study, *International Journal of Production Research,* 23, 403-421

Baybars, I., 1986. A survey of exact algorithms for simple assembly line balancing problem, *Management Science,* 32, 11-17

Eral, Erdal, Sarin S.C., 1998. A survey of the assembly line balancing procedures, *Production Planning and Control,* 9, 34-42

Forteca D.J., Guest C.L., Elam M., Karr C.L., 2005. A fuzzy logic approach to assembly line balancing, *Mathare & Soft Computing,* 57-74

Gutjahr, A.L., Neumhauser G.L., 1964, An algorithm for the balancing problem, *Management Science*, 11, 23-35

Halgeson W. B., Birnie D. P., 1961. Assembly line balancing using the ranked positional weighting technique, *Journal of Industrial Engineering,* 12*,* 18-27

Kao, E.P.C., 1976. A preference order dynamic program for stochastic assembly line balancing, *Management Science*, 22, 19-24

Lee, T.O., Kim Y., Kim Y.K., 2005. Two-sided assembly line balancing to maximize work relatedness and slackness, *Computers & Industrial Engineering,* 40, 273-292

Salveson, M.E., 1955. The assembly line balancing problem, *Journal of Industrial Engineering,* 62-69

Scholl, A., 1998. Balancing and sequencing of assembly line, *Physica- Verlag*

Sury, R.J., 1971. Aspects of assembly line balancing, *International Journal of Production Research*, 9, 8-14

AN EVOLUTIONARY ALGORITHM FOR UNICAST/ MULTICAST TRAFFIC ENGINEERING

Miguel Rocha, Pedro Sousa
Dep. Informatics/ CCTC, Universidade do Minho, Campus de Gualtar, Braga, Portugal
mrocha@di.uminho.pt, pns@di.uminho.pt

Paulo Cortez
Dep. Information Systems/ Algoritmi, Universidade do Minho, Campus de Azurem, Guimaraes, Portugal
pcortez@dsi.uminho.pt

Miguel Rio
Dep. Electric and Electronic Engineering, University College London, Torrington Place, London, U.K.
m.rio@ee.ucl.ac.uk

Keywords: Traffic engineering, Multicast content, Evolutionary Algorithms, OSPF.

Abstract: A number of Traffic Engineering (TE) approaches have been recently proposed to improve the performance of network routing protocols, both developed over MPLS and intra-domain protocols such as OSPF. In this work, a TE approach is proposed for routing optimization in scenarios where unicast and multicast demands are simultaneously present. Evolutionary Algorithms are used as the optimization engine with overall network congestion as the objective function. The optimization aim is to reach a set of (near-)optimal weights to configure the OSPF protocol, both in its standard version and also considering the possibility of using multi-topology variants. The results show that the proposed optimization approach is able to obtain networks with low congestion, even under scenarios with heavy unicast/multicast demands.

1 INTRODUCTION

A new plethora of network services is putting strong requirements on TCP/IP networks, for which these were not initially designed. Many of these services will need a multicast enabled network with demanding quality of service (QoS) constraints in the end-to-end data delivery. The advent of 3-play service providers, where the same entity is involved in the network and TV provision is just the first of these scenarios. Interactive TV, virtual reality, video-conferencing, video games or video surveillance are just some of the applications that would gain from QoS enabled multicast content delivery. In these scenarios, data needs to arrive to a set of users with minimal loss (therefore requiring minimization of congestion) and with acceptable end-to-end delays.

Multicast has been present for a while in TCP/IP networks, but its widespread use has never occurred in the Internet. It is, however, used in closed TCP/IP networks where its scalability problems are not a deterrent. In fact, many IPTV and video-on-demand services operate in closed networks using multicast to save bandwidth and enhance QoS levels.

In this context, Traffic Engineering (TE) techniques can be used to improve network performance by achieving near-optimal configurations for routing protocols. TE approaches can be classified into: Multi-Protocol Label Switching (MPLS) (Davie and Rekhter, 2000)(Awduche and Jabbari, 2002) based and pure IP-based intra-domain routing protocols. With MPLS, packets are encapsulated with labels at ingress points, that can be used to route these packets along an explicit label-switched path). Together with resource reservation mechanisms, these capabilities are able to support stringent end-to-end bandwidth guarantees for multicast content delivery. However, the use of MPLS presents significant drawbacks: firstly, it adds significant complexity to the model, since per-flow state has to be stored in every router of the path; secondly, MPLS failure recovery mechanisms are considerably more complex than the typical router convergence ones; finally, it represents a considerable network management overhead.

As regards intra-domain routing protocols, the most commonly used today is Open Shortest Path First(OSPF)(Thomas II, 1998). Here, the administrator assigns weights to each link in the network,

which are then used to compute the best path from each source to each destination using the Dijkstra algorithm (Dijkstra, 1959). The results are then used to compute the routing tables in each node.

A number of studies have proposed TE procedures which optimize the weights of intra-domain routing protocols to achieve near optimal routing, taking as input the expected traffic demands. This was the approach taken by Fortz and Thorup (2000) where this task was viewed as an optimization problem by defining a cost function that measured network congestion. The authors proved that this task is a NP-hard problem and proposed some local search heuristics that compared well with the MPLS model. Another approach was the use of Evolutionary Algorithms (EAs) to improve these results (Ericsson et al., 2002). Additional research has been carried out with the objective of pursuing multiconstrained QoS optimization, where both traffic demands and delay requirements are considered in the optimization of routing configurations for unicast traffic (Rocha et al., 2006).

In this paper, EAs are employed to reach OSPF weights that optimize network congestion, taking into account both unicast and multicast demands of a given domain. This work is based on the reasoning that in the optimization process both the unicast and multicast demands should be considered simultaneously, in contrast with previous work where optimization is performed in two distinct phases, the first for unicast traffic and the second devoted to multicast optimization (Wang and Pavlou, 2007).

2 PROBLEM FORMULATION

2.1 Unicast Traffic

In this section, a model for a network only with unicast traffic demands will de described. This is based on the framework proposed in (Fortz and Thorup, 2000). The general routing problem (Ahuja et al., 1993) that underpins this work represents routers and links by a set of nodes (N) and arcs (A) in a directed graph $G = (N,A)$. In this model, c_a represents the capacity of each link $a \in A$. A demand matrix D is available, where each element d_{st} represents the traffic demand between nodes s and t. For each arc a, the variable $f_a^{(st)}$ represents how much of the traffic demand between s and t travels over arc a. The total unicast load on each arc a (l_a) can be defined as: $l_a = \sum_{(s,t)\in N\times N} f_a^{st}$ while the link utilization rate u_a is given by: $u_a = \frac{l_a}{c_a}$. It is then possible to define a congestion measure for each link: $\Phi_a = p(u_a)$. using a penalty function p that has small values near 0, but as the values approach the unity it becomes more expensive and exponentially penalizes values above 1 (Fortz and Thorup, 2000).

In OSPF, all arcs have an integer weight, used by each node to calculate the shortest paths to all other nodes in the network, using the Dijkstra algorithm (Dijkstra, 1959). The traffic from a given source to a destination travels along the shortest path. If there are two or more paths with equal length, traffic is evenly divided among the arcs in these paths (load balancing) (Moy, 1998). Let us assume a given a weight assignment, and the corresponding values of u_a. In this case, the total routing cost is expressed by $\Phi = \sum_{a\in A}\Phi_a$ for the loads and penalties (Φ_a) calculated based on the given OSPF weights. In this way, the *OSPF weight setting problem* is equivalent to finding the optimal weight value for each link, in order to minimize Φ. The congestion measure can be normalized (Φ^*) over distinct scenarios to values in the range [1,5000]. It is important to note that in the case when all arcs are exactly full ($l_a = c_a$), the value of Φ^* is $10\frac{2}{3}$, a value that will be considered a threshold that bounds the acceptable working region of the network.

2.2 Multicast Demands

A model that considers only multicast traffic in the network will be described, that is based on a the work by Wang and Pavlou (2007) . If there are unicast and multicast demands, this model can be used to perform a two-step optimization process (explained in the next section). Consider, as before, a network topology $G = (N,A)$, with arc capacities (c_a). The multicast demands are given for a set of G groups, where for each group $g \in G$ the following parameters are defined: a root node r_g, a bandwidth demand M_g and a a set of receivers (V_g). The multicast optimization problem is typically defined as the computation of a bandwidth constrained Steiner tree, with the objective of minimizing overall bandwidth consumption, using integer programming. The target is to instantiate a number of binary decision variables: y_a^g, are equal to 1 if link a is included in the multicast tree for group g; and $x_a^{g,k}$ are equal to 1 if link a is included in the multicast tree for group g, in the branch from the root node to receiver k. The objective function is to minimize the overall bandwidth consumption ($L1$):

$$L1 = \sum_{g\in G}\sum_{a\in A} M_g \times y_a^g \qquad (1)$$

The deployment of the obtained Steiner trees can be enforced by using an explicit routing overlay, through MPLS on a per-group basis. An alterna-

tive with some advantages, previously discussed, is to consider that the routing will be achieved by using an intra-domain protocol such as OSPF. In this case, the tree for a given group will be built from the shortest paths between the root node and each receiver. Therefore, the values assigned to y_a^g variables will be computed as follows: y_a^g is equal to 1 if link a is in the shostest path from the root node g to at least one of the receivers in V_g, and is equal to 0 otherwise.

In previous work (Wang and Pavlou, 2007), EAs have been proposed to optimize OSPF weights for multicast traffic. The objective function used in this case is based on the overall network load (L1) but also on the excessive bandwidth allocated to overloaded links (L2), that can is given by:

$$L2 = \sum_{a \in A} [w_a \sum_{g \in G} (M_g \times y_a^g) - c_a] \quad (2)$$

$$w_a = \begin{cases} 0, & if \quad \sum_{g \in G} M_g \times y_a^g \leq c_a \\ 1, & otherwise \end{cases} \quad (3)$$

The EA's fitness is, therefore, given by:

$$f(L1, L2) = \frac{\mu}{\alpha \times L1 + \beta \times L2} \quad (4)$$

where μ, α and β are constants, whose values are set to 10^7, 1 and 10 respectively.

2.3 Unified Model with Unicast and Multicast Demands

In this work, a unified approach will be proposed that is able to reach OSPF weights that optimize the network congestion measure, simultaneously considering unicast and multicast demands. In this case, the multicast load for a given link a can be computed as: $ml_a = \sum_{g \in G} M_g \times y_a^g$. The values of y_a^g will be calculated from the OSPF weights as explained in the previous section. So, the total load on a given arc a is given by: $l_a = ml_a + ul_a$, where ul_a is the unicast load in arc a (given by l_a in the previous section). It should noted that l_a here takes the meaning of the total load in the network, while in Section 2.1 l_a is only used for unicast loads since in that case those were the only loads considered. After calculating the overall values of l_a for all links, the process proceeds as described in Section 2.1, in order to reach the normalized congestion measure Φ^*.

Another interesting measure of the network performance in this scenario is the excessive bandwidth in overloaded links (BOL). This is a generalization of L2 but now applied to the global loads and not only to the multicast traffic. This is defined as:

$$BOL = \sum_{a \in A} z_a (l_a - c_a) \quad (5)$$

$$z_a = \begin{cases} 0, & if \quad l_a \leq c_a \\ 1, & otherwise \end{cases} \quad (6)$$

3 OPTIMIZATION ALGORITHMS

3.1 Evolutionary Algorithms

Evolutionary Algorithms (EAs) (Michalewicz, 1996) are a popular family of optimization methods, inspired in the biological evolution. These methods work by evolving a population, i.e. a set of individuals, each encoding solutions to a target problem in an artificial chromosome. Each individual is evaluated through a fitness function, that assigns it a numerical value, corresponding to the quality of the encoded solution. EAs are stochastic methods due to their selection process. In fact, individuals selected to create new solutions are taken from the population using probabilities. Highly fit individuals have a higher probability of being selected, but the less fit still have their chance.

In the proposed EA, each individual encodes a solution in a direct way, i.e. as a vector of integer values, where each value corresponds to the weight of an arc in the network (the values range from 1 to w_{max}). Therefore, the size of the individual equals the number of links in the network. If multiple topologies are used, i.e. different sets of weights for unicast and multicast, the size of the individual is twice the number of links and the two sets of weights are encoded linearly, i.e. the first L genes encode the weights for unicast traffic, while the latter L links encode the weights for multicast (L is the number of links).

The weight values for individuals in the initial population are randomly generated, taken from a uniform distribution. In order to create new solutions, several reproduction operators were used, more specifically two mutation and one crossover operator:

- *Random Mutation*, replaces a given weight value by a random value, within the allowed range;
- *Incremental/decremental Mutation*, replaces a given weight value w by $w+1$ or by $w-1$ (with equal probabilities);
- *Uniform crossover*, a standard crossover operator (Michalewicz, 1996).

The operators are all used to create new solutions with equal probabilities. The selection procedure is done by converting the fitness value into a linear ranking in the population, and then applying a roulette

wheel scheme. In each generation, 50% of the individuals are selected from the previous generation, and 50% are bred by the application of the genetic operators over selected parents. A population size of 100 individuals was considered.

3.2 Optimization Approaches

Three distinct optimization approaches are compared, with the aim to optimize OSPF weights in networks where both unicast and multicast demands are available. All these methods use EAs as the optimization engine. The first method is a 2-step optimization process (2S), based on the proposal from Section 2.2 (Wang and Pavlou, 2007), that can be described as:

1. the OSPF weights are optimized (using EAs) to minimize congestion penalties (Φ^*) only taking into account the unicast demands;
2. the bandwidths used for each link in unicast traffic are deduced from the link capacities;
3. a second optimization process is conducted, where a different set of weights is calculated from multicast traffic only, by running a new EA with f(L1,L2) (Equation 4) as the fitness function.

This method assumes that a protocol that allows multiple sets of weights, each for a distinct type of traffic, is deployed. This is the case, for instance, of the multi-topology protocol MT-OSPF (Psenak et al., 2006). The remaining alternatives are based on the model proposed in Section 2.3. Using this model, two different optimization approaches may be followed:

- Single topology (ST), i.e. a single set of OSPF weights is used for both types of traffic demands;
- Multiple topologies (MT), i.e. two sets of OSPF weights are used, one for unicast traffic and the other for multicast demands. In this case, as before, a multi-topology protocol has to be used.

4 EXPERIMENTS AND RESULTS

4.1 Experimental Setup

To evaluate the proposed algorithms, a number of experiments were conducted. The experimental platform used in this work is presented in Figure 1. All algorithms and the OSPF routing simulator were implemented using the Java language. A set of 3 network topologies was created using the Brite topology generator (Medina et al., 2001), varying the number of nodes ($N = 30, 50, 80$) and the average degree of each node was kept in ($m = 4$). This resulted in networks ranging with 110, 190 and 310 links, respectively.

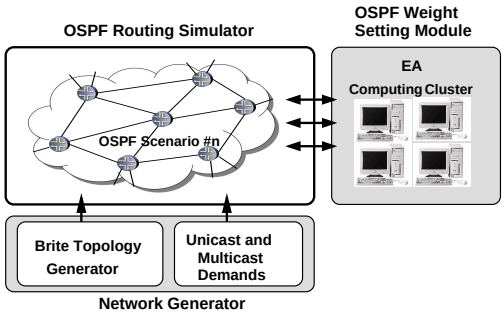


Figure 1: Experimental platform for EA's performance evaluation.

The link bandwidth (capacity) was generated by a uniform distribution between 1 and 10 Gbits/s. The networks were generated using the Barabasi-Albert model, using a heavy-tail distribution and an incremental grow type (parameters *HS* and *LS* were set to 1000 and 100). Next, the unicast demand matrices (D) were generated (two distinct matrices for each network). A parameter (D_p) was considered, representing the expected mean of congestion in each link (values for D_p were 0.2 and 0.3).

The generation of the multicast traffic demands was based on the following: Firstly, for each network the number of groups G was set equal to the number of nodes. The root node for each group was randomly chosen from the set of nodes (with equal probabilities). For each group, the number of receivers was generated from the range $[2, n/2]$, where n is the number of nodes. The set of receivers V_g was created with the given cardinality, by randomly selecting a set of nodes different from the root. Finally, the demand M_g was generated taking a parameter (R) into account. R is defined as the ratio between the total multicast demands and the total unicast demands. Given R and given the unicast demands, a target is calculated for the total multicast demands. The group demands are generated by dividing the target value by the different groups in an uneven way, so that groups with different demands are created resulting in a more plausible scenario. By using R, a better understanding of the results is possible since an approximate idea of the trade-offs between unicast and multicast is known. The values of R used were 1 and 0.5.

The termination criteria for all optimization approaches consisted in a maximum number of solutions evaluated. This value ranged from 100000 to 300000, increasing linearly with the number of links. For all cases, w_{max} was set to 20 and 20 runs were executed and the results presented are the means.

Table 1: Results for the network with 30 nodes.

Demands	Metric	ST	MT	2S
D = 0.2	Φ^*	1.38	1.31	1.83
R = 0.5	BOL	0	0	42
	L1 ($\times 10^5$)	1.10	1.09	0.96
D = 0.2	Φ^*	3.27	3.00	8.66
R = 1.0	BOL	160	156	1138
	L1 ($\times 10^5$)	2.11	2.11	1.97
D = 0.3	Φ^*	3.32	2.83	7.78
R = 0.5	BOL	257	128	875
	L1 ($\times 10^5$)	1.50	1.54	1.39
D = 0.3	Φ^*	66.2	38.0	93.5
R = 1.0	BOL	14239	7040	16820
	L1 ($\times 10^5$)	3.05	3.04	2.84

Table 2: Results for the network with 50 nodes.

Demands	Metric	ST	MT	2S
D = 0.2	Φ^*	1.38	1.34	1.90
R = 0.5	BOL	0	0	56
	L1 ($\times 10^5$)	2.10	2.11	1.91
D = 0.2	Φ^*	2.29	1.85	5.69
R = 1.0	BOL	99	9	1166
	L1 ($\times 10^5$)	4.50	4.54	4.18
D = 0.3	Φ^*	2.44	2.04	3.15
R = 0.5	BOL	88	30	241
	L1 ($\times 10^5$)	2.58	2.63	2.38
D = 0.3	Φ^*	21.9	7.47	26.8
R = 1.0	BOL	11820	2862	10880
	L1 ($\times 10^5$)	5.58	5.72	5.39

4.2 Results

In Tables 1, 2 and 3 the results for the optimization approaches are shown. In the first column, the demand generation parameters (*D*,*R*) are shown. The second column shows the metrics, while columns 3, 4 and 5 show the results of each optimization approach, according to the metrics. Four scenarios are given for each network: the first rows show the example with less demands, while the last rows show the worst case scenario. The middle rows show two intermediate scenarios, where in one case the unicast demands are low, but the multicast demands are high and in the next the reverse takes place.

A different perspective is shown in Figures 2, 3 and 4, where the congestion measure (Φ^*) for the three networks is plotted. In each plot, the four scenarios in terms of demands are shown. The values are shown in a logarithmic scale, given the exponential nature of the penalty function.

Table 3: Results for the network with 80 nodes.

Demands	Metric	ST	MT	2S
D = 0.2	Φ^*	1.44	1.38	1.74
R = 0.5	BOL	0	0	7
	L1 ($\times 10^5$)	2.54	2.63	2.33
D = 0.2	Φ^*	2.09	1.90	3.02
R = 1.0	BOL	160	77	582
	L1 ($\times 10^5$)	5.01	5.12	4.64
D = 0.3	Φ^*	2.63	2.50	3.97
R = 0.5	BOL	227	247	821
	L1 ($\times 10^5$)	3.48	3.59	3.18
D = 0.3	Φ^*	17.7	10.4	32.6
R = 1.0	BOL	11426	6856	15242
	L1 ($\times 10^5$)	7.25	7.49	6.90

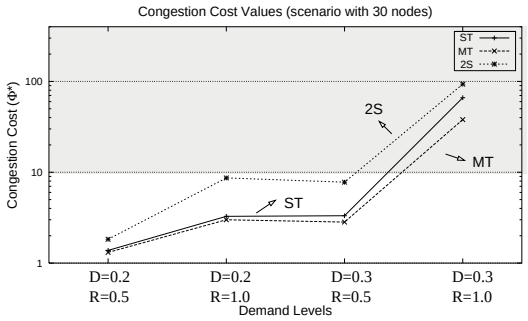


Figure 2: A plot of the results for congestion measure (network with 30 nodes).

4.3 Discussion

The first conclusion to draw from these results is that 2S leads to sub-optimal results, in terms of overloaded links and network congestion (visible both in the BOL and Φ^*). Both the MT and ST show better results, being able to keep the network in an acceptable behaviour in most scenarios. The scenario shown in the last rows ($D = 0.2$ and $R = 1.0$) is an extreme case, where the demands are quite high. Although this would not be acceptable in a real world network, it is useful to have an idea of how the distinct optimization methods scale. When comparing ST and MT, the results are quite near in the low demand scenarios. In these cases, the gain obtained by using MT is not impressive. The gap increases with the values of *D* and *R*, i.e. as the problem gets harder. In practical terms, this would mean that if the network has lots of resources in terms of bandwidth and low demands, it is probably not worth to pay the cost of deploying a multi-topology protocol. On the other hand, using this kind of protocol allows the network to support higher demands with the same bandwidth resources.

Regarding the L1 values, the best alternative is

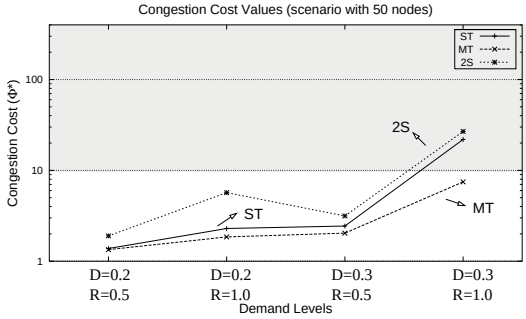


Figure 3: A plot of the results for congestion measure (network with 50 nodes.

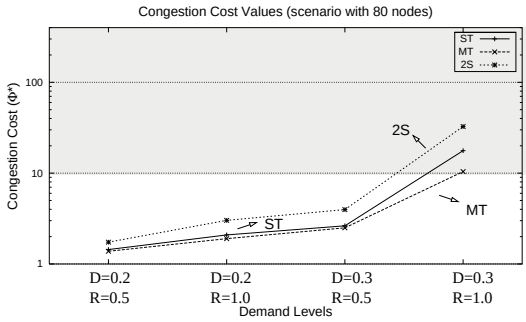


Figure 4: A plot of the results for congestion measure (network with 80 nodes.

2S and thus this method allocates better the multicast traffic. This is not surprising, since L1 is part of the objective function in this case. However, this solution does not result in a good performance when the network congestion is taken as a whole. So, this solution would lead to high levels of loss, that would be unacceptable in most cases.

5 CONCLUSIONS

The optimization of OSPF weights brings important tools for traffic engineering, without demanding modifications on the basic network model. This work presented EAs for routing optimization in networks with unicast and multicast demands. Resorting to a set of network configurations and unicast/ multicast demands, it was shown that the proposed EAs were able to provide OSPF weights that can lead to good network behaviour. The proposed approach was favourably compared to a 2-step optimization procedure, proposed in previous work, that leads to sub-optimal results in terms of network congestion and overloaded links. The advantages of using a multi-topology protocol in these scenarios were also studied and it was concluded that these are most advantageous when the network bandwidth resources are limited.

The main contribution of this work is the capability of optimizing the OSPF weights considering all factors involved (i.e. all types of traffic). Using the proposed methods, the network administrator can decide if a multi-topology protocol is needed or simply use a standard implementation of OSPF. In the future, we will proceed with the integration of further QoS constraints in the model, with a priority on the introduction of end-to-end delays.

ACKNOWLEDGEMENTS

This work was supported by the Portuguese Foundation for Science and Technology under project POSC/EIA/59899/2004, partially funded by FEDER.

REFERENCES

Ahuja, R., Magnati, T., and Orlin, J. (1993). *Network Flows*. Prentice Hall.

Awduche, D. and Jabbari, B. (2002). Internet traffic engineering using multi-protocol label switching (MPLS). *Computer Networks*, 40:111–129.

Davie, B. and Rekhter, Y. (2000). *MPLS: Multiprotocol Label Switching Technology and Applications*. Morgan Kaufmann, USA.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(269-271).

Ericsson, M., Resende, M., and Pardalos, P. (2002). A Genetic Algorithm for the Weight Setting Problem in OSPF Routing. *J. of Combinatorial Optimization*, 6:299–333.

Fortz, B. and Thorup, M. (2000). Internet Traffic Engineering by Optimizing OSPF Weights. In *Proceedings of IEEE INFOCOM*, pages 519–528.

Medina, A., Lakhina, A., Matta, I., and Byers, J. (2001). BRITE: Universal Topology Generation from a User's Perspective. Technical Report 2001-003.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, USA, third edition.

Moy, J. (1998). *OSPF, Anatomy of an Internet Routing Protocol*. Addison Wesley.

Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and Pillay-Esnault, P. (2006). Multi-topology (mt) routing in ospf (internet draft).

Rocha, M., Sousa, P., Rio, M., and Cortez, P. (2006). Qos constrained internet routing with evolutionary algorithms. In *Proc. IEEE Conference Evolutionary Computation*, pages 9270–9277. IEEE Press.

Thomas II, T. (1998). *OSPF Network Design Solutions*. Cisco Press.

Wang, N. and Pavlou, G. (2007). Traffic Engineered Multicast Content Delivery Without MPLS Overlay. *IEEE Transactions on Multimedia*, 9(3).

THE PARALLELIZATION OF MONTE-CARLO PLANNING
Parallelization of MC-Planning

S. Gelly, J. B. Hoock, A. Rimmel, O. Teytaud
TAO (Inria), Lri, UMR Cnrs 8623, Univ. Paris-Sud, France
name@lri.fr

Y. Kalemkarian
High Performance Calculus, Bull, Grenoble, France
yann.kalemkarian@bull.net

Keywords: Bandit-based Monte-Carlo planning, parallelization, multi-core machines, message-passing.

Abstract: Since their impressive successes in various areas of large-scale parallelization, recent techniques like UCT and other Monte-Carlo planning variants (Kocsis and Szepesvari, 2006a) have been extensively studied (Coquelin and Munos, 2007; Wang and Gelly, 2007). We here propose and compare various forms of parallelization of bandit-based tree-search, in particular for our computer-go algorithm XYZ.

1 INTRODUCTION

Dynamic programming (Bellman, 1957; Bertsekas, 1995) provides a robust and stable tool for dynamic optimization. However, its application to large scale problems is far from simple and involves approximations, as in approximate value functions in approximate dynamic programming (Powell, 2007). Some alternate approaches, like using simulations for focusing on more important areas of the state space as in RTDP (Barto et al., 1993), have provided some important yet unstable tools for large scale approximate dynamic programming. Monte-Carlo planning, applied to Go since (Bruegmann, 1993), and in particular Bandit-based Monte-Carlo planning (Kocsis and Szepesvari, 2006a), provides an alternate solution, increasing the importance of simulations. Bandit-based Monte-Carlo planning features the construction of a tree approximating incrementally the distribution of possible futures. Tools from the bandit literature (see (Lai and Robbins, 1985; Auer et al., 2001); adversarial case in (Kocsis and Szepesvari, 2005); huge sets of arms in (Banks and Sundaram, 1992; Agrawal, 1995; Dani and Hayes, 2006; Berry et al., 1997)) are used in order to bias the random development of the tree in the direction of the most important directions; the important paper (Kocsis and Szepesvari, 2006a) applies recursively bandits for the biased random development of a tree. The efficiency of recursive bandits is also shown by some challenges won by such methods (Hussain et al., 2006). Its application to computer-go has provided particularly impressive results (Coulom, 2006; Wang and Gelly, 2007). The decision taking, at each step, is far more expensive, but the overall cost is much lower than the computational cost of one dynamic programming loop when the branching factor (or the state space) is large. The analysis of Monte-Carlo planning if far from being closed, as it is not clear that upper-confidence bounds are the best tool for that, in particular when heuristics like AMAF(Bruegmann, 1993) (also termed RAVE) are included. However, we here consider a fixed empirically tuned algorithm, and we focus on its parallelization, for various forms of parallelization (multi-core machines, standard clusters). The application to computer-go is one of the most illustrative applications of bandit-based Monte-Carlo planning methods. In particular, Go is still far from being solved, as moderate human players are still much better than the best computer-go programs. Monte-Carlo planning is a revolution in computer-go, in some cases combined with tactical knowledge (Cazenave and Helmstetter, 2005; Coulom, 2007) but also in some cases almost from scratch without any specific knowledge (Wang and Gelly, 2007), except some likelihood of random playouts. The parallelization of Monte-Carlo planning is classical in the multi-core case; we here provide an analysis of the speed-up of this multi-core approach (shared memory). We then move to the message-passing architecture, for

a cluster-parallelization. We briefly introduce bandits and bandit-based Monte-Carlo planning below, for the sake of clarity of notations. We then present the multi-core parallelization in section 2, and two different approaches for the cluster parallelization in section 3, 3.1. A bandit problem is typically defined as follows: (1) A finite set $A = \{1,\dots,N\}$ of arms is given. (2) Each arm $a \in A$ is equipped with an unknown probability of reward p_a. (3) At each time step $t \in \{1,2,\dots\}$, the algorithm chooses $a_t \in A$ depending on $(a_1,\dots,a_{t-1})$ and $(r_1,\dots,r_{t-1})$. (4) The bandit gives a reward r_t, with $r_t = 1$ with probability p_{a_t} (independently), and $r_t = 0$ otherwise. A bandit algorithm is an algorithm designed for minimizing the so-called regret:

$$Regret = T \inf_{a \in A} p_a - \sum_{i=1}^{T} r_t.$$

Important results, depending on assumptions on the set of actions and their distribution of rewards, include (Lai and Robbins, 1985; Auer et al., 2001; Banks and Sundaram, 1992; Agrawal, 1995; Dani and Hayes, 2006; Berry et al., 1997; Hussain et al., 2006; Kocsis and Szepesvari, 2005; Coquelin and Munos, 2007). A bandit is said to have side information if some information other than the reward is available to the algorithm. We present various bandit algorithms in section 1: these bandit algorithms usually work as follows for each time step:

- compute a score, for each possible arm, which consists in (i) an exploitation term, larger for arms for which the average rewards in the past is good; (ii) an exploration term, larger for arms which have not been tried often.
- choose the arm with maximum score.

Typical improvements are: (i) give heuristically a score to arms which have not been visited yet (see e.g. (Wang and Gelly, 2007; Coulom, 2007)); (ii) guess some side information (see the AMAF heuristic e.g. (Gelly and Silver, 2007)); (iii) restrict the sampling to the $k(t)$ first nodes for $k(t)$ some non-decreasing mapping (see progressive widening in (Coulom, 2007)). In the case of Go and more generally bandit-based Monte-Carlo planning, we use one bandit at each node of the tree, as explained in section 1. The difficult elements are the anytime nature of the problem, its non stationarity (Hussain et al., 2006; Kocsis and Szepesvari, 2006b), its large number of arms (Banks and Sundaram, 1992). The side information (usually termed AMAF) is detailed below.

This section provides an overview of bandit-based Monte-Carlo planning algorithms. It is presented in the case of a game; the experiments are performed in the case of Go. We point out that UCT can be applied to trees far from minimax problems: max/max problems, or max/Expectation, etc. Algorithm 1 provides an overview (see also flowchart in Fig. 1), and Algorithm 3 provides a more detailed presentation. The main point in Algorithm 1 is that the tree is unbalanced, with a strong bias in favor of important parts of the tree. The function used for taking decisions out of the tree (i.e. the so-called Monte-Carlo part) is defined in (Wang and Gelly, 2007) The function used for simulating in the tree is presented in Algorithm

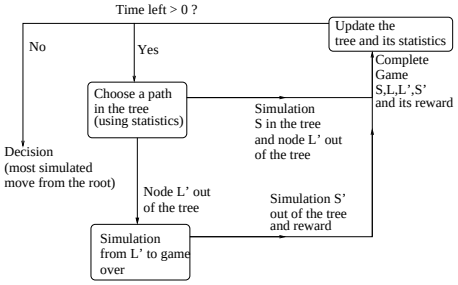


Figure 1: Flow chart of UCT. The bandit algorithm is applied in the simulation part in the tree.

Algorithm1: Monte-Carlo planning algorithm. The final line chooses the decision with the conservative rule that the most simulated decision should be chosen. Other solutions, such as taking the decision with the higher ratio "number of winning simulations divided by the number of simulations", are too much dangerous because a decision might be very poor due to a small number of simulations. Some other robust and elegant techniques include the use of lower confidence bounds. Here, rewards are binary (win or loss), but arbitrary distributions of rewards can be used.

Initialize T to the root, representing the current state. T is a tree of positions, with each node equipped with statistics.
while Time left > 0 **do**
  Simulate one game until a leaf (a position) L of T (thanks to bandit algorithms). Then, choose one son (a successor) L' of L.
  Simulate one game from position L' until the game is over.
  Add L' as a son of L in T.
  Update statistics in all the tree. In UCT, each node knows how many winning simulations (from this node) have been performed and how many simulations (from this node) have been performed. In other forms of tree-search, other informations are necessary (heuristic information in AMAF, total number of nodes in the tree in BAST(Coquelin and Munos, 2007)).
end while
Pick up the decision which has been simulated most often from the root.

2. This function is a main part of the code: it decides in which direction the tree should be extended. There are various formulas, all of them being based on the idea of a compromise between exploitation (further simulating the seemingly good moves) and exploration (further simulating the moves which have not been explored a lot). The empirically best tool in

Algorithm 2: Taking a decision in the tree. The total number of simulations at situation s is $Simulations(s) = \sum_d Simulations(s,d)$. We present here various formulas for computing the score (see (Lai and Robbins, 1985; Auer et al., 2001; Gelly and Silver, 2007) for UCB1, UCB-Tuned and AMAF respectively); other very important variants (for infinite domains, larger number of arms, of specific assumptions) can be found in (Banks and Sundaram, 1992; Agrawal, 1995; Dani and Hayes, 2006; Coquelin and Munos, 2007).

Function $decision = Bandit(situation\ s\ in\ the\ tree)$.
for d in the set of possible decisions **do**
  Let $\hat{p}(d) = Wins(s,d)/Simulations(s,d)$.
  SWITCH **(bandit formula)**:
  - **UCB1**: compute $score(d) = \hat{p}(d) + \sqrt{2\log(Simulations(s))/Simulations(s,d)}$.
  - **UCB-Tuned.1**: compute $score(d) = \hat{p}(d) + \sqrt{\hat{V}\log(Simulations(s))/Simulations(s,d)}$ with $\hat{V} = \max(0.001, \hat{p}(d)(1-\hat{p}(d)))$.
  - **UCB-Tuned.2**: compute $score(d) = \hat{p}(d) + \sqrt{\hat{V}\log(Simulations(s))/Simulations(s,d)} + \log(Simulations(s))/Simulations(s,d)$ with $\hat{V} = \max(0.001, \hat{p}(d)(1-\hat{p}(d)))$.
  - **AMAF-guided exploration**: Compute $score(d) = \alpha(d)\hat{p}(d) + (1-\alpha(d))\hat{\hat{p}}(d)$ with:
    - $\hat{\hat{p}}(d)$ the AMAF-estimate of the asymptotic value of $score(d)$.
    - $\alpha(d)$ a coefficient depending on $Simulations(s,d)$ and the AMAF-confidence in $\hat{\hat{p}}(d)$ (see (Gelly and Silver, 2007)).
  END SWITCH
end for
Return $\arg\max_d\ score(d)$.

Algorithm 2 is the AMAF-guided exploration. In that case, in the classical formalism of bandits, each time one arm is played, we get: (i) a positive or null reward for the chosen arm (0 or 1); (ii) a list of arms, consisting in half of the arms roughly, positively correlated with the list of arms for which the rewards would have been positive if these arms have been played. The bandit algorithm for this bandit problem has been empirically derived in (Gelly and Silver, 2007). To the best of our knowledge, there's no mathematically grounded bandit algorithm in that case. An important improvement (termed progressive widening(Coulom, 2007)) of Algorithm 2 consists in considering only the $K(n)$ "best" moves (the best according to some heuristic), at the n^{th} simulation in a given node, with $K(n)$ is an non-decreasing mapping from $\mathbb{N}$ to $\mathbb{N}$.

Algorithm 3: More detailed Monte-Carlo planning algorithm.

Initialize T to the root, representing the current state. T is a tree of positions.
while Time left > 0 **do**
  Simulate one game until a leaf (a position) L of T (thanks to bandit algorithms applied until a leaf is met, see Algorithm 2).
  Choose one son (a successor) L' of L, possibly with some offline learning (Gelly and Silver, 2007).
  Simulate one game from position L' until the game is over.
  Add L' as a son of L in T.
  Update statistics in all the tree. In UCT, each node knows how many winning simulations (from this node) have been performed and how many simulations (from this node) have been performed. In other forms of tree-search, other informations are necessary (heuristic information in AMAF, total number of nodes in the tree in BAST(Coquelin and Munos, 2007)).
end while
Pick up the decision which has been simulated most often from the root.

2 MULTI-CORE PARALLELIZATION

The multi-core parallelization is intuitively the most natural one: the memory is shared. We just have to distribute the loop on various threads (each thread performs simulations independently of other threads, with just mutexes protecting the updates in memory), leading to algorithm 4. Consider N the number of

Algorithm 4: Multi-core Monte-Carlo planning algorithm.

Initialize T to the root, representing the current state. T is a tree of positions.
For each thread simultaneously:
while Time left > 0 **do**
  Simulate one game until a node (=position) L of T (thanks to bandit algorithms); put this simulation S in memory. Choose a successor L' of L.
  Simulate one game from position L' until the game is over; put this simulation S' in memory.
  Add L' as a son of L in T.
  Update statistics in all the tree thanks to S and S'.
end while
Pick up the decision which has been simulated most often.

threads. The number of simulations per second is typically almost multiplied by N. However, this algorithm is not equivalent to the sequential one: possibly,

Table 1: Success rate against mogoRelease3, for various computation times. Under the two-assumptions (1) almost N times more simulations per second with N cores (2) no impact of the "delay" point pointed out in the text, the speed-up would be linear and all the raws would be equal. We see a decay of performance for 4 threads.

Nb threads × comp. time	10 sec.procs	20 secs.procs	40 secs.procs
1 thread	51.1 ± 1.8	62.0 ± 2.2	74.4 ± 2.4
2 threads		62.9 ± 1.8	
4 threads			66.4 ± 2.1

$N-1$ simulations are running when one more simulation is launched, and the updates of T corresponding to these $N-1$ simulations are not taken into account. There is a $N-1$ delay, and the analysis of the delay is not straightforward - we will quantify this effect experimentally. We get the following empirical results: Table 1 confirms the roughly 63% success rate known when doubling the computational power. As a conclusion, in 9x9 Go, the speed-up of the multi-core algorithm 4 is linear for two nodes, slightly below the linear speed-up for 4 nodes (by interpolation, we can estimate the speed-up as 3 for 4-cores).

3 CLUSTER PARALLELIZATION

First, let's consider the generalization of the multi-core approach to a cluster, i.e. a version with massive communication in order to keep roughly the same state of the memory on all nodes. As the memory is not shared here, we have to broadcast on the network many update-informations; each simulation on one node leads to one broadcast. Possibly, we can group communications in order to reduce latencies. This leads to the algorithm 5. $T = 0$ is perhaps possible, for high performance clusters or processors inside the same machine. Let's consider the idealized case of a cluster with negligible communication cost and infinite number of nodes. Let's assume that a proportion α of time is spent in the updates. Also, let's assume that the delay of updates does not reduce the overall efficiency of the algorithm. What is the speed-up in that case ? Consider M the number of simulations per second on one node in the (mono-node) case. With N nodes, at each time steps, we get NM simulations. The number of updates is therefore NM per second of simulation. If the time of one update is T, for each group of M simulations, (i) each node performs M simulations (costs $1-\alpha$ second); (ii) each node sends M update-information (costs 0 second); (iii) each node receives $(N-1)M$ update-informations (costs 0 second); (iv) each node updates its tree with these $(N-1)M$ update informations (costs $\alpha(N-1)$ second). If we divide by the number N of nodes and let $N \to \infty$, we get (i) a cost $(1-\alpha)/N \to 0$; (ii) a cost 0 for sending update-informations; (iii) a cost 0 for receiving update-informations; (iv) a cost $\alpha(N-1)/N \to \alpha$ for updates. This implies that the main cost is the update-cost, and that asymptotically, the speed-up is $1/\alpha$. In the case of MoGo, this leads to $\alpha \simeq 0.05$ and therefore roughly 20 as maximal speed-up for the case of a tree simultaneously updated on all nodes. As communications are far from negligible, as preliminary experiments were disappointing and as we expect better than the 20 speed-up, we will not keep this algorithm in the sequel.

Algorithm 5: Cluster algorithm for Monte-Carlo planning. As there are many paths leading to the same node, we must use a hash table, so that with some key (describing a goban) we can find if a node is in the tree and what are its statistics in constant time.

for Each node **do**
  Initialize T to the root, representing the current state of the root. T is a tree of positions, with statistics attached to each node.
end for
for For each computer simultaneously: **do**
  for For each thread simultaneously: **do**
    while Time left > 0 **do**
      Simulate one game until a node (=position) L of T (thanks to bandit algorithms); put this simulation S in memory. Choose a successor L' of L.
      Simulate one game from position L' until the game is over; put this simulation S' in memory. S,S' is a complete game starting at the root.
      Add L' as a son of L in T, and update all the statistics in T with S,S'.
      Add (L,L',S,S') to a stack of to-be-sent simulations.
      if $time - t0 \geq T$ and thread=first thread **then**
        Set $t0 = time$.
        Send all the (L,L',S,S') in the stack to all other nodes.
        Reset the stack.
        Receive many (L,L',S,S') from all other nodes.
        for Each (L,L',S,S') received **do**
          Add L' as a son of L in T (if not present).
          Update all the statistics in T with S,S'.
        end for
      end if
    end while
  end for
end for
Pick up the decision which has been simulated most often.

3.1 An Alternate Solution with Less Communications

Section 3 showed that whenever communications are perfect, the speed-up[1] is limited to some constant $1/\alpha$, roughly 20 in MoGo. We propose the following algorithm (Algorithm 6), with the following advantages: (1) much less communications (can run on usual ethernet); (2) tolerant to inhomogeneous nodes (as other algorithms above also); (3) our implementation is not yet fault-tolerant, but it could be done; (4) self-assessment possible (strong variance → more time). The algorithm is detailed in Algorithm 6.

Algorithm 6: Algorithm for Monte-Carlo planning on a cluster.

```
Initialize T to the root, representing the current state. T
is a tree of positions. T is the same on all computers.
for Each computer simultaneously: do
  for Each thread simultaneously: do
    while Time left > 0 do
      Simulate one game until a node (=position) L of
      T (thanks to bandit algorithms).
      Choose one son L' of L.
      Simulate one game from position L' until the
      game is over.
      Add L' as a son of L in T_0.
      Thread 0 only: if time - t0 ≥ T_0, set t0 = time,
      and average statistics in all the tree for nodes of
      depth ≤ K with at least N_min simulations.
    end while
  end for
end for
Decision step:Pick up the decision which has been sim-
ulated most often, on the whole set of nodes.
```

An important advantage of this technique is that averaging vectors of statistics is possible quickly on a large number of nodes: the computational cost of this averaging over N nodes is $O(\log(N))$. The case $T_0 = \infty$ in Algorithm 6 (no communication before the final step of the decision making) is just a form of averaging. For computer-go, we get 59 %± 3% of success rate with an averaging over 43 machines versus a single machine, whereas a speed-up 2 leads to 63%. This means that averaging provides a speed-up less than 2 with 43 machines; this is not a good parallelization. We then experiment T_0 finite and $K = 0$ (i.e. only the root of the tree is shared between nodes):

[1]This is not a "real" speed-up, as the parallel algorithm, even in the multi-core case, is not equivalent to the sequential one - the difference is the "delay" detailed in section 2. We here consider that the speed-up is k if the parallel algorithm is as efficient as the sequential one with k times more time.

Number N of nodes	T_0 (time between updates)	Success rate	Estimated Speed-up divided by N
3	1.00s	67.2 ± 3.5	0.95 (9x9)
3	0.11s	67.3 ± 2.1	0.94 (9x9
4	0.33s	69.6 ± 1.5	0.81 (9x9)
9	0.33s	79.0 ± 1.0	(9x9)
9	0.11s	83.8 ± 5.4	(19x19)

We have no estimate for the 9-machines speed-up, because comparing computational budget B with 9 machines and 9B with 1 machine implies the use of games with average time per move 5.B, which requires a lot of computational power. However, the 83.8% is in the extrapolation of linear speed-up. The results were not significantly different with higher numbers of levels. We guess that for larger numbers of nodes the results will be different but we have not yet any empirical evidence of this.

4 CONCLUSIONS

Computer-Go is both a main target for computer-games, as the main unsolved game, and a challenging planification problem as the most promising approaches have a moderate expert knowledge and are therefore quite general. In particular, successful applications of the UCT approach have been reported very far from computer-Go (Kocsis and Szepesvari, 2006a). A main advantage of the approach is that it is fully scalable. Whereas many expert-based tools have roughly the same efficiency when doubling the computational power, bandit-based Monte-Carlo planning with time 2B has success rate roughly 63% against a bandit-based Monte-Carlo planning algorithm with time B. This leads to the hope of designing a parallel platform, for an algorithm that would be efficient in various tasks of planifications. The main results in this paper are the followings:

- Doubling the computational power (doubling the time per move) leads to a 63% success rate against the non-doubled version.
- The straightforward parallelization on a cluster (imitating the multi-core case by updating continuously the trees in each node so that the memory is roughly the same in all nodes) does not work in practice and has strong theoretical limitations, even if all computational costs are neglected.
- A simple algorithm, based on averages which are easily computable with classical message passing libraries, a few times per second, can lead to a great successes; in 19x19, we have reached, with 9 nodes, 84 % success rate against one equivalent node. This

success rate is far above simple voting schemas, suggesting that communications between independent randomized agents are important and that communicating only at the very end is not enough.
- Our two parallelizations (multi-core and cluster) are orthogonal, in the sense that: (i) the multi-core parallelization is based on a faster breadth-first exploration (the different cores are analyzing the same tree and go through almost the same path in the tree; in spite of many trials, we have no improvement by introducing deterministic or random diversification in the different threads. (ii) the cluster parallelization is based on sharing statistics guiding the first levels only of the tree, leading to a natural form of load balancing. The deep exploration of nodes is completely orthogonal. Moreover, the results are cumulative; we see the same speed-up for the cluster parallelization with multi-threaded versions of the code or mono-thread versions.
- In 9x9 Go, we have roughly linear speed-up until 4 cores or 9 nodes. The speed-up is not negligible beyond this limit, but not linear. In 19x19 Go, the speed-up remains linear until at least 4 cores and 9 machines.
Extending these results to higher numbers of machines is the natural further work. Increasing the number of cores is difficult, as getting an access to a 16-cores machine is not easy. Monte-Carlo planning is a strongly innovative tool with more and more applications, in particular in cases in which variants of backwards dynamic programming do not work. Extrapolating the results to the human scale of performance is difficult. People usually consider that doubling the computational power is roughly equivalent to adding almost one stone to the level. This is confirmed by our experiments. Then, from the 2nd or 3rd Kyu of the sequential MoGo in 19x19, we need 10 or 12 stones for the best human level. Then, we need a speed-up of a few thousands. This is far from impossible, if the speed-up remains close to linear with more nodes.

REFERENCES

Agrawal, R. (1995). The continuum-armed bandit problem. *SIAM J. Control Optim.*, 33(6):1926–1951.

Auer, P., Cesa-Bianchi, N., and Gentile, C. (2001). Adaptive and self-confident on-line learning algorithms. *Machine Learning Journal*.

Banks, J. S. and Sundaram, R. K. (1992). Denumerable-armed bandits. *Econometrica*, 60(5):1071–96. Available at http://ideas.repec.org/a/ecm/emetrp/v60y1992i5p1071-96.html.

Barto, A., Bradtke, S., and Singh, S. (1993). Learning to act using real-time dynamic programming. Technical Report UM-CS-1993-002.

Bellman, R. (1957). *Dynamic Programming*. Princeton Univ. Press.

Berry, D. A., Chen, R. W., Zame, A., Heath, D. C., and Shepp, L. A. (1997). Bandit problems with infinitely many arms. *Ann. Statist.*, 25(5):2103–2116.

Bertsekas, D. (1995). *Dynamic Programming and Optimal Control, vols I and II*. Athena Scientific.

Bruegmann, B. (1993). Monte carlo go. *Unpublished*.

Cazenave, T. and Helmstetter, B. (2005). Combining tactical search and monte-carlo in the game of go. *IEEE CIG 2005*, pages 171–175.

Coquelin, P.-A. and Munos, R. (2007). Bandit algorithms for tree search. In *Proceedings of UAI'07*.

Coulom, R. (2006). Efficient selectivity and backup operators in monte-carlo tree search. *In P. Ciancarini and H. J. van den Herik, editors, Proceedings of the 5th International Conference on Computers and Games, Turin, Italy*.

Coulom, R. (2007). Computing elo ratings of move patterns in the game of go. In van den Herik, H. J., Uiterwijk, J. W. H. M., Winands, M., and Schadd, M., editors, *Computer Games Workshop, Amsterdam*.

Dani, V. and Hayes, T. P. (2006). Robbing the bandit: less regret in online geometric optimization against an adaptive adversary. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 937–943, New York, NY, USA. ACM Press.

Gelly, S. and Silver, D. (2007). Combining online and offline knowledge in uct. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 273–280, New York, NY, USA. ACM Press.

Hussain, Z., Auer, P., Cesa-Bianchi, N., Newnham, L., and Shawe-Taylor, J. (2006). Exploration vs. exploitation challenge. *Pascal Network of Excellence*.

Kocsis, L. and Szepesvari, C. (2005). Reduced-variance payoff estimation in adversarial bandit problems. In *Proceedings of the ECML-2005 Workshop on Reinforcement Learning in Non-Stationary Environments*.

Kocsis, L. and Szepesvari, C. (2006a). Bandit-based monte-carlo planning. *ECML'06*.

Kocsis, L. and Szepesvari, C. (2006b). Discounted-ucb. In *2nd Pascal-Challenge Workshop*.

Lai, T. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22.

Powell, W.-B. (2007). *Approximate Dynamic Programming*. Wiley.

Wang, Y. and Gelly, S. (2007). Modifications of UCT and sequence-like simulations for Monte-Carlo Go. In *IEEE Symposium on Computational Intelligence and Games, Honolulu, Hawaii*, pages 175–182.

THE BEES ALGORITHM AND MECHANICAL DESIGN OPTIMISATION

D. T. Pham, M. Castellani, M. Sholedolu
Manufacturing Engineering Centre, Cardiff University, Cardiff, U.K.
PhamDT@cf.ac.uk, CastellaniM@cf.ac.uk, SholedoluM3@cf.ac.uk

A. Ghanbarzadeh
Mechanical Engineering Department, Engineering Faculty, Shahid Chamran University, Ahvaz, Iran
Ghanbarz@yahoo.com

Keywords: Bees Algorithm, Optimisation, Mechanical Design.

Abstract: The Bees Algorithm is a search procedure inspired by the way honey-bees forage for food. A standard mechanical design problem, the design of a welded beam structure, was used to benchmark the Bees Algorithm against other optimisation techniques. The paper presents the results obtained showing the robust performance of the Bees Algorithm.

1 INTRODUCTION

Researchers have used the design of welded beam structures (Rekliatis et al., 1983) as a benchmark problem to test their optimisation algorithms. The welded beam design problem involves a nonlinear objective function and eight constraints. A number of optimisation techniques have been applied to this problem. Some of them, such as geometric programming (Ragsdell and Phillips, 1976), require extensive problem formulation; some (see, for example, (Leite and Topping, 1998)) use specific domain knowledge which may not be available for other problems, and others (see, for example, (Ragsdell and Phillips, 1976)) are computationally expensive or give poor results.

The Bees Algorithm has been applied to different optimisation problems (Pham et al., 2005, Pham et al., 2006b, Pham et al., 2006a). The design problems discussed in this paper are constrained optimisation problems to be solved using this new algorithm.

The remainder of the paper is structured as follows. Section 2 explains the main features of the foraging process and the steps of the Bees Algorithm. Section 3 describes the welded beam design problem. Section 4 presents the results obtained using the Bees Algorithm and other optimisation procedures. Section 5 concludes the paper.

2 THE BEES ALGORITHM

2.1 The Foraging Process in Nature

During the harvesting season, a colony of bees keeps a percentage of its population as scouts (Von Frisch, 1976) and uses them to explore the field surrounding the hive for promising flower patches. The foraging process begins with the scout bees being sent to the field where they move randomly from one patch to another.

When they return to the hive, those scout bees that found a patch of a sufficient quality (measured as the level of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the "waggle dance" (Seeley, 1996). This dance is the means to communicate to other bees three pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive, and its quality rating (or fitness) (Von Frisch, 1976, Camazine et al., 2003). This information helps the bees watching the dance to find the flower patches without using guides or maps. After the waggle dance, the dancer (i.e. the scout bee) goes back to

the flower patch with follower bees recruited from the hive. The number of follower bees will depend on the overall quality of the patch. Flower patches with large amounts of nectar or pollen that can be collected with less effort are regarded as more promising and attract more bees (Seeley, 1996, Bonabeau et al., 1999). In this way, the colony can gather food quickly and efficiently.

2.2 The Bees Algorithm

This section summarises the main steps of the Bees Algorithm. For more details, the reader is referred to (Pham et al., 2006b, Pham et al., 2006a, Pham et al., 2005). Figure 1 shows the pseudo code for the Bees Algorithm. The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of sites selected for neighbourhood searching (out of n visited sites) (m), number of top-rated (elite) sites among m selected sites (e), number of bees recruited for the best e sites (nep), number of bees recruited for the other (m-e) selected sites (nsp), the initial size of each patch (ngh) (a patch is a region in the search space that includes the visited site and its neighbourhood), and the stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitnesses of the sites visited by the scout bees are evaluated in step 2.

1. Initialise population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) //Forming new population.
4. Select sites for neighbourhood search.
5. Determine the patch size.
6. Recruit bees for selected sites (more bees for best *e* sites) and evaluate fitnesses.
7. Select the fittest bee from each patch.
8. Abandon sites without new information.
9. Assign remaining bees to search randomly and evaluate their fitnesses.
10. End While.

Figure 1: Pseudo code of the Bees Algorithm.

In step 4, the *m* sites with the highest fitnesses are designated as "selected sites" and chosen for neighbourhood search. In step 5, the size of neighbourhood around the selected sites is determined. In step 6, the algorithm conducts searches around the selected sites, assigning more bees to search in the vicinity of the best *e* sites. Selection of the best sites can be made directly according to the fitnesses associated with them. Alternatively, the fitness values are used to determine the probability of the sites being selected. Searches in the neighbourhood of the best *e* sites – those which represent the most promising solutions - are made more detailed. As already mentioned, this is done by recruiting more bees for the best *e* sites than for the other selected sites. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm.

In step 7, for each patch, only the bee that has found the site with the highest fitness (the "fittest" bee in the patch) will be selected to form part of the next bee population. In nature, there is no such a restriction. This restriction is introduced here to reduce the number of points to be explored. In step 8, sites which have not shown improvements in fitness values over a number of recruitment cycles are abandoned. This is because it is assumed that such sites correspond to stationary points. The locations of the sites are recorded. In step 9, the remaining bees in the population are assigned randomly around the search space to scout for new potential solutions.

At the end of each iteration, the colony will have two parts to its new population: representatives from the selected patches, and scout bees assigned to conduct random searches. These steps are repeated until a stopping criterion is met.

As described above, the Bees Algorithm is suitable for unconstrained optimisation problems. If a problem involves constraints, a simple technique can be adopted to enable the optimisation to be applied. The technique involves subtracting a large number from the fitness of a particular solution that has violated a constraint in order drastically to reduce the chance of that solution being found acceptable. This was the technique adopted in this work. As both design problems were minimisation problems, a fixed penalty was added to the cost of any constraint-violating potential solution.

3 WELDED BEAM DESIGN PROBLEM

A uniform beam of rectangular cross section needs to be welded to a base to be able to carry a load of $6000\ lbf$. The configuration is shown in Figure 2. The beam is made of steel 1010.

The length *L* is specified as 14 *in*. The objective of the design is to minimise the cost of fabrication while finding a feasible combination of weld thickness h, weld length l, beam thickness t and beam width b. The objective function can be formulated as (Rekliatis et al., 1983) :

$$\text{Min } f = (1+c_1)h^2 l + c_2 tb(L+l) \quad (1)$$

where

f = Cost function including setup cost, welding labour cost and material cost;

c_1 = Unit volume of weld material cost $=0.10471\ \$/in.^3$;

c_2 = Unit volume of bar stock cost $=0.04811\ \$/in.^3$;

L = Fixed distance from load to support = $14\ in$;

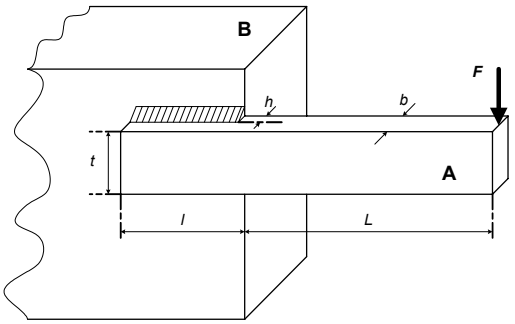


Figure 2: A Welded Beam.

Not all combinations of *h*, *l*, *t* and *b* which can support *F* are acceptable. There are limitations which should be considered regarding the mechanical properties of the weld and bar, for example, shear and normal stresses, physical constraints (no length less than zero) and maximum deflection. The constraints are as follows (Rekliatis et al., 1983):

$$g_1 = \tau_d - \tau \geq 0 \quad (2)$$
$$g_2 = \sigma_d - \sigma \geq 0 \quad (3)$$
$$g_3 = b - h \geq 0 \quad (4)$$
$$g_4 = l \geq 0 \quad (5)$$
$$g_5 = t \geq 0 \quad (6)$$
$$g_6 = P_c - F \geq 0 \quad (7)$$
$$g_7 = h - 0.125 \geq 0 \quad (8)$$
$$g_8 = 0.25 - \delta \geq 0 \quad (9)$$

where

τ_d = Allowable shear stress of weld = $13600\ Psi$;

τ = Maximum shear stress in weld;

σ_d = Allowable normal stress for beam material = $30000\ Psi$;

σ = Maximum normal stress in beam;

P_c = Bar buckling load;

F = Load = $6000\ lbf$;

δ = Beam end deflection.

The first constraint (g_1) ensures that the maximum developed shear stress is less than the allowable shear stress of the weld material. The second constraint (g_2) checks that the maximum developed normal stress is lower than the allowed normal stress in the beam. The third constraint (g_3) ensures that the beam thickness exceeds that of the weld. The fourth and fifth constraints (g_4 and g_5) are practical checks to prevent negative lengths or thicknesses. The sixth constraint (g_6) makes sure that the load on the beam is not greater than the allowable buckling load. The seventh constraint (g_7) checks that the weld thickness is above a given minimum, and the last constraint (g_8) is to ensure that the end deflection of the beam is less than a predefined amount.

Normal and shear stresses and buckling force can be formulated as (Shigley, 1973, Rekliatis et al., 1983):

$$\sigma = \frac{2.1952}{t^3 b} \quad (10)$$

$$\tau = \sqrt{(\tau')^2 + (\tau'')^2 + (l\tau'\tau'') \Big/ \sqrt{0.25(l^2 + (h+t)^2)}} \quad (11)$$

where

$$\tau' = \frac{6000}{\sqrt{2}hl} \quad \text{(Primary stress)}$$

$$\tau'' = \frac{6000(14+0.5l)\sqrt{0.25(l^2+(h+t)^2)}}{2\left\{0.707hl\left(l^2/12 + 0.25(h+t)^2\right)\right\}}$$

(Secondary stress)

$$P_c = 64746.022(1-0.0282346t)tb^3 \quad (12)$$

4 RESULTS AND DISCUSSION

The empirically chosen parameters for the Bees Algorithm are given in Table 1 with the stopping criterion of 750 generations. The search space was defined by the following intervals (Deb, 1991):

$$0.125 \le h \le 5 \quad (13)$$

$$0.1 \le l \le 10 \quad (14)$$

$$0.1 \le t \le 10 \quad (15)$$

$$0.1 \le b \le 5 \quad (16)$$

With the above search space definition, constraints g_4, g_5 and g_7 are already satisfied and do not need to be checked in the code.

Table 1: Parameters of the Bees Algorithm for the welded beam design problem.

Bees Algorithm parameters	Symbol	Value
Population	n	80
Number of selected sites	m	5
Number of top-rated sites out of m selected sites	e	2
Initial patch size	ngh	0.1
Number of bees recruited for best e sites	nep	50
Number of bees recruited for the other (m-e) selected sites	nsp	10

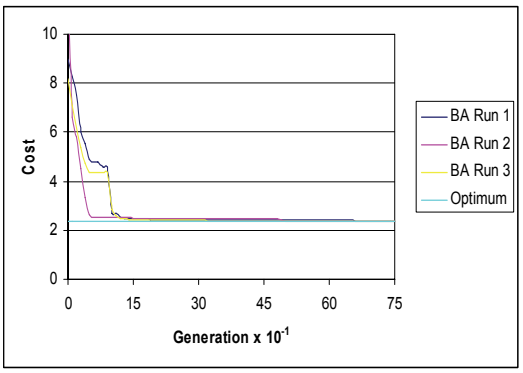


Figure 3: Evolution of the lowest cost in each iteration.

Figure 3 shows how the lowest value of the objective function changes with the number of iterations (generations) for three independent runs of the algorithm. It can be seen that the objective function decreases rapidly in the early iterations and then gradually converges to the optimum value.

A variety of optimisation methods have been applied to this problem by other researchers (Ragsdell and Phillips, 1976, Deb, 1991, Leite and Topping, 1998). The results they obtained along with those of the Bees Algorithm are given in Table 2. APPROX is a method of successive linear approximation (Siddall, 1972). DAVID is a gradient method with a penalty (Siddall, 1972). Geometric Programming (GP) is a method capable of solving linear and nonlinear optimisation problems that are formulated analytically (Ragsdell and Phillips, 1976). SIMPLEX is the Simplex algorithm for solving linear programming problems (Siddall, 1972).

As shown in Table 2, the Bees Algorithm produces better results than almost all the examined algorithms including the Genetic Algorithm (GA) (Deb, 1991), an improved version of the GA (Leite and Topping, 1998), SIMPLEX (Ragsdell and Phillips, 1976) and the random search procedure RANDOM (Ragsdell and Phillips, 1976). Only APPROX and DAVID (Ragsdell and Phillips, 1976) produce results that match those of the Bees Algorithm. However, as these two algorithms require information specifically derived from the problem (Leite and Topping, 1998), their application is limited. The result for GP is close to those of the Bees Algorithm but GP needs a very complex formulation (Ragsdell and Phillips, 1976).

5 CONCLUSIONS

A constrained optimisation problem was solved using the Bees Algorithm. The algorithm converged to the optimum without becoming trapped at local optima. The algorithm generally outperformed other optimisation techniques in terms of the accuracy of the results obtained. A drawback of the algorithm is the number of parameters that must be chosen. However, it is possible to set the values of those parameters after only a few trials.

Indeed, the Bees Algorithm can solve a problem without any special domain information, apart from that needed to evaluate fitnesses. In this respect, the Bees Algorithm shares the same advantage as global search algorithms such as the Genetic Algorithm (GA). Further work should be addressed at reducing the number of parameters and incorporating better learning mechanisms to make the algorithm even simpler and more efficient.

Table 2: Results for the welded beam design problem obtained using the Bees Algorithm and other optimisation methods.

Methods	Design variables				Cost
	h	*l*	*t*	*b*	
APPROX (Ragsdell and Phillips, 1976)	0.2444	6.2189	8.2915	0.2444	2.38
DAVID (Ragsdell and Phillips, 1976)	0.2434	6.2552	8.2915	0.2444	2.38
GP (Ragsdell and Phillips, 1976)	0.2455	6.1960	8.2730	0.2455	2.39
GA (Deb, 1991) **Three independent runs**	0.2489	6.1730	8.1789	0.2533	2.43
	0.2679	5.8123	7.8358	0.2724	2.49
	0.2918	5.2141	7.8446	0.2918	2.59
IMPROVED GA (Leite and Topping, 1998) Three independent runs	0.2489	6.1097	8.2484	0.2485	2.40
	0.2441	6.2936	8.2290	0.2485	2.41
	0.2537	6.0322	8.1517	0.2533	2.41
SIMPLEX (Ragsdell and Phillips, 1976)	0.2792	5.6256	7.7512	0.2796	2.53
RANDOM (Ragsdell and Phillips, 1976)	0.4575	4.7313	5.0853	0.6600	4.12
BEES ALGORITHM Three independent runs	0.24429	6.2126	8.3009	0.24432	2.3817
	0.24428	6.2110	8.3026	0.24429	2.3816
	0.24432	6.2152	8.2966	0.24435	2.3815

ACKNOWLEDGEMENTS

The research described in this paper was performed as part of the Objective 1 SUPERMAN project, the EPSRC Innovative Manufacturing Research Centre Project and the EC FP6 Innovative Production Machines and Systems (I*PROMS) Network of Excellence.

REFERENCES

Bonabeau, E., Dorigo, M. & Theraulaz, G. (1999) *Swarm Intelligence: from Natural to Artificial Systems,* New York, Oxford University Press.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraula, G. & Bonabeau, E. (2003) *Self-Organization in Biological Systems,* Princeton, Princeton University Press.

Deb, K. (1991) Optimal Design of a Welded Beam via Genetic Algorithm. *AIAA Journal,* 29**,** 2013-2015.

Leite, J. P. B. & Topping, B. H. V. (1998) Improved Genetic Operators for Structural Engineering Optimization. *Advances in Engineering Software,* 29**,** 529-562.

Pham, D. T., Ghanbarzadeh, A., Koc, E. & Otri, S. (2006a) Application of the Bees Algorithm to the Training of Radial Basis Function Networks for Control Chart Pattern Recognition. IN TETI, R. (Ed.) *5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (CIRP ICME '06).* Ischia, Italy.

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S. & Zaidi, M. (2005) Technical Report MEC 0501- The Bees Algorithm. Cardiff, Manufacturing Engineering Centre, Cardiff University.

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S. & Zaidi, M. (2006b) The Bees Algorithm, A Novel Tool for Complex Optimisation Problems. *2nd Int Virtual Conf on Intelligent Production Machines and Systems (IPROMS 2006).* Oxford, Elsevier.

Ragsdell, K. M. & Phillips, D. T. (1976) Optimal Design of a Class of Welded Structures Using Geometric Programming. *ASME Journal of Engineering for Industry,* 98**,** 1021-1025.

Rekliatis, G. V., Ravindrab, A. & Ragsdell, K. M. (1983) *Engineering Optimisation Methods and Applications,* New York, Wiley.

Seeley, T. D. (1996) *The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies,* Cambridge, Massachusetts, Harvard University Press.

Shigley, J. E. (1973) *Mechanical Engineering Design,* Ney York, McGraw-Hill.

Siddall, J. N. (1972) *Analytical Decision-making in Engineering Design,* New Jersey, Prentice-Hall.

Von Frisch, K. (1976) *Bees: Their Vision, Chemical Senses and Language,* Ithaca, N.Y., Cornell University Press.

POSTERS

EVALUATION OF NEURAL PDF CONTROL STRATEGY APPLIED TO A NONLINEAR MODEL OF A PUMPED-STORAGE HYDROELECTRIC POWER STATION

G. A. Munoz-Hernandez, C. A. Gracios-Marin, A. Diaz-Sanchez
Instituto Tecnologico de Puebla, Puebla, México
gmunoz@ieee.org, cgracios@hotmail.com, adiazsan@inaoep.mx
Phone (52) 222-229-88-24 Fax (52) 222-222-21-14

S. P. Mansoor, D. I. Jones
University of Wales, Bangor, School of Informatics, Dean Street, Bango, LL57 1UT, U.K.
s.mansoor@bangor.ac.uk, dewi@informatics.bangor.ac.uk
Phone (44) 1248-38-27-16 Fax (44) 1248-36-14-29

Keywords: Model and Simulation, Power systems, Control applications, Neural control.

Abstract: In this paper, a neural Pseudoderivative control (PDF) is applied to a nonlinear mathematical model of the Dinorwig pumped - storage hydroelectric power station. The response of the system with this auto-tuning controller is compared with that of a classic controller, currently implemented on the system. The results show how the application of PDF control to a hydroelectric pumped-storage station improves the dynamic response of the power plant, even when multivariable effects are taken into account.

1 INTRODUCTION

Dinorwig is a large pumped storage hydroelectric scheme located in North Wales that is operated by the First Hydro Company. The station has six 300 MW rated turbines, driving synchronous generators which feed power into the national grid. Dinorwig provides rapid response frequency control when peak demands occur. This hydroelectric station has a single tunnel, drawing water from an upper reservoir into a manifold, which splits the main flow into six penstocks. Each penstock feeds a turbine to generate power using a guide vane to regulate the flow. The electrical power generated is controlled by individual feedback loops on each unit. The reference input to the power loop is the grid frequency deviation from its 50 Hz set point, thus forming an outer frequency control loop. Mansoor et al, have derived a multivariable nonlinear simulation model of this plant, which has provided an improved understanding of its characteristics (Mansoor, Jones, Bradley, & Aris, Stability of a pumped storage hydropower station connected to a power system, 1999) (Mansoor, Jones, Bradley, Aris, & Jones, 2000). Its main features are non-minimum-phase dynamics, poorly damped poles (associated with water-hammer in the supply tunnel and electrical synchronization) and a nonlinear relationship between flow and power. It is also known (Kundur, 1994) (Working group on prime mover energy supply, 1992) that there is a significant hydraulic coupling between the turbines because of the common supply. This makes the plant a good candidate for the application of auto-tuning control.

The paper begins with a brief discussion of the nonlinear mathematical model of the power plant. Then a few concepts of neural network theory are reviewed, followed by a description of the application of neural Pseudoderivative control (PDF) to the model of Dinorwig (Kang, Lee, Kim, Kwon, & Choi, 1991). Finally, results are presented which show the improved response provided by neural PDF.

2 HYDROELECTRIC PLANT MODEL

The hydroelectric plant model can be divided into three subsystems: guide vane, nonlinear hydraulics and turbine/generator (figure 1). Mansoor et al developed a multivariable non-linear model that includes a rate limit and saturation in the guide vane dynamics, as shown in figure 2 (Mansoor, Jones, Bradley, Aris, & Jones, 2000).

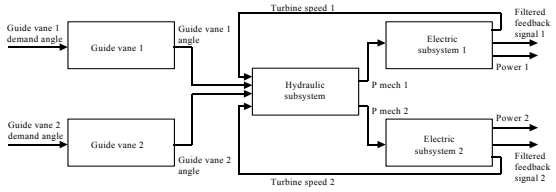


Figure 1: MIMO model of the hydroelectric plant with two penstocks.

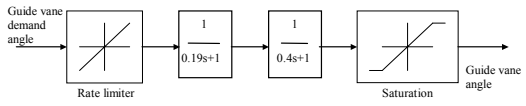


Figure 2: Guide vane subsystem.

In this study a nonlinear model that takes into account the effects of the water column, including water compressibility and pipe wall elasticity, was employed (Working group on prime mover energy supply, 1992). Figure 3 shows the nonlinear elastic model of a single penstock. The coupling effect between the units is included in the model (main tunnel block).

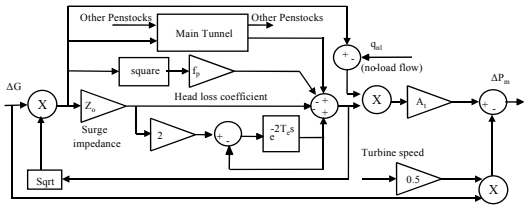


Figure 3: Hydraulic subsystem.

The turbine gain value of A_t depends directly on the turbine MW rating and inversely on the Generator MVA rating. f_p is the head loss coefficient for the penstock. Z_0 is the surge impedance of the conduit. T_e is the wave travel time; it is defined as the time taken for the pressure wave to travel the length of the penstock (l) to the open surface. v is the velocity of sound in water.

$$T_e = \frac{l}{v} \quad (1)$$

$$Z_0 = \frac{T_W}{T_e} \quad (2)$$

T_w is the *water starting time* of the main tunnel and the penstocks. Kundur defines the *water starting time* as the time required for a head to accelerate the water in the penstock from standstill to a specific velocity (Kundur, 1994). Its value depends directly on the constructional dimensions of main tunnel and penstocks.

In this model G is the per unit gate opening, P *mech* is the mechanical power produced by a single turbine. The value of T_e depends on the length of the penstock and inversely on the wave velocity (equation 1). Z_o depends directly on the flow rate, inversely on the head of water and on the acceleration due to gravity (equation 2). The value of A_t depends directly on the turbine MW rating and inversely on the Generator MVA rating (Mansoor, 2000). The models are expressed in the per-unit system, normalized to 300 MW and 50 Hz. The electrical subsystem is based on the 'swing' equations (Kundur, 1994) and includes the effect of synchronizing torque. For noise reduction a first order filter is included in the feedback loop (fig. 4).

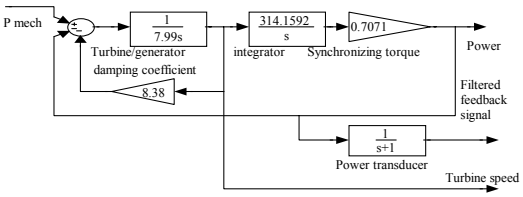


Figure 4: Electrical subsystem.

3 NEURAL NETWORKS

3.1 Basic Theory

Since the early 1980s, there has been a dramatic increase in research on the computational properties of highly interconnected networks of simple processing units called artificial neural networks. These networks are loosely patterned after the structure of biological nervous systems. However, the use of these artificial neural networks (NN) to improve the behavior of several real systems in engineering applications has recently been increased. One of the engineering disciplines that have been enriched with the properties of the NN is the adaptive control theory, because they offer the

possibility to adjust the parameters of the regulator in order to reduce the difference between the set-point and the output of the process.
There are several types of NN can be found in literature (Narendra & Mukhopadhyay, 1996) but in adaptive control, back propagation is used most frequently, because its calculation speed is fast and easy to implement. A back - propagation artificial neural network is a linear combination of nodes interconnected to form several layers of nodes that may or may not have interactions between them, figure 5.

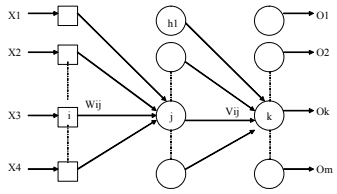


Figure 5: Generic structure for three layer neural network.

The number of layers used in the network plays an important factor during the design stage. Two layers NN have its own limitation but it has a good performance (Minsky & Papert, 1988). Multilayer NN have a wide spectrum of applications and they can deal with problems that are "impossible" to NN with two layers. As was discussed by Rumelhart et al (Rumelhart, McClelland, & group, 1986), the addition of internal layers will allow the back propagation algorithm to develop an internal representation of system dynamics; that feature could be crucial to find a solution. Linear models as the ARX result on internal models of two layers NN with back-propagation.

3.2 Neural PDF

One of the main reasons for using NN in control system is the ability to adjust any non-linear system. A *prior* knowledge about the structure of the system being controlled is very important to tune and improve the performance of PDF controller.

There are several approaches to define a fast and efficient control strategy to calculate and adjust the parameters of discrete PID control systems (Narendra & Mukhopadhyay, 1996) (Garcez & Garcez, 1995). For this work a similar strategy was used to tune a discrete PDF.

Narendra and Mukhopadhyay (Narendra & Mukhopadhyay, 1996) provided a good alternative to make identification on-line of the coefficients using a model on the system. In this situation, the non-linear part of the model is approximated to a linear system. The coefficients of the process are fed back to re-calculate the K's parameters of the PID applied.

There have been several works where the NN have been applied to hydroelectric systems. Garcez applied a PI neural to a linear simulator of a 20 MW hydroelectric power plant (Garcez & Garcez, 1995). Djukanovic, validated an adaptive-network based on fuzzy inference system to control a low head hydropower system (Djukanovic, Calovic, Vesovic, & Sobajic, 1997). Yin-Song, presented a self-learning control system using a PID Fuzzy NN, which was applied it to hydraulic turbine governor system (Yin-Song, Guo-Cai, & Ong-Xiang, 2000). Recently, Shu-Qing, compared a PID controller with a hybridized controller based on genetic algorithms and fuzzy NN for governors of a hydroelectric power plant model (Shu-Qing, Zhao-Hui, Zhi-Huai, & Zi-Peng, 2005).

In this paper a back-propagation strategy has been used to adjust the parameters of a discrete PDF regulator. This technique was introduced by Aguado (Aguado Behar, 2000). Figure 6 shows the scheme of Neural-PDF. The regulation can be calculated by:

$$v_j(t+1) = v_j(t) + \eta sign(\frac{\partial e_y}{\partial e_u})\delta^1 h_j \qquad (3)$$

$$w_{ji}(t+1) = w_{ji}(t) + \eta sign(\frac{\partial e_y}{\partial e_u})\delta^2{}_j x_i \qquad (4)$$

$$\frac{\partial E(t)}{\partial v_j} = -\delta^1 h_j \frac{\partial e_y}{\partial e_u} \qquad (5)$$

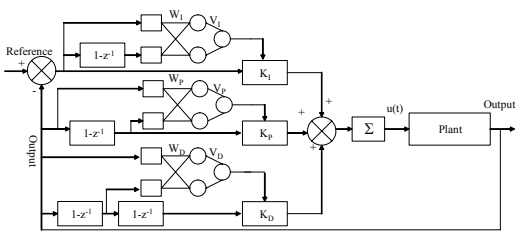


Figure 6: Neural PDF.

4 SIMULINK MODEL AND PROGRAM

A Simulink model was developed to facilitate studies of the power plant under different governors. Libraries of special functions (blocks) and the power plant models were constructed by connecting these functions to the standard Simulink functions. Using a dialog box, the parameters of a specific block can be adjusted, for example, the operating point of linear models may be changed. These models can represent the power plant as SISO or MIMO system and linear or nonlinear behaviour may be selected. Figure 7 shows a schematic of the Simulink power plant model.

The full hydroelectric station model is constructed combining the four sub-systems: Guide vane dynamics, hydraulic subsystem, turbine/generator and sensor filters. Each block is part of the Simulink library developed for this study; they can be selected to represent a diversity of modes of operation. For example there are three models available to simulate the hydraulic subsystem: Linear, nonlinear nonelastic and nonlinear elastic. The guide vane dynamics can be selected with or without rate limitation and saturation. The sensor filters block is a fixed block. The grid model can be adjusted to represent different conditions of the national grid. Through the governor block classic and advanced controls can be selected.

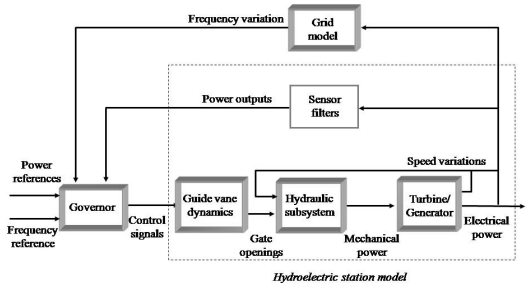


Figure 7: Simulink power plant model.

Simulink S-functions for the neural PDF algorithms were developed. These functions are connected to Simulink plant models. The neural PDF block accepts η (learning parameters) and sample time. The input signals to the PDF block consist of the reference and the output signals of the plant and its output is the plant control signal. The versatility of Simulink is very important to change easily the plant model or even modify the algorithm and quickly see the new results. The neural algorithm calculates the optimal values of the control parameters. The current optimal criterion programmed is quadratic error, where the error is the output deviation from the set-point; however the criterion of optimization can be changed. The algorithm takes some time to find the "best" range of parameter values (training time) when these ranges have been reached the parameters stay constant until the set-point or the plant model change.

5 RESULT OF SIMULATION

The role of a hydroelectric station in frequency control mode is to provide timely and accurate supply of its target power contribution to the power system. The actual form of the power demand is related to Grid frequency variation but, for testing, it can be specified in terms of step, ramp and random input signals. Jones *et al* have proposed a step and ramp response for single unit operation (Jones, Mansoor, Aris, Jones, Bradley, & King, 2004). This step response specification for single unit operation is expressed in Figure 8 and Table 1 (these are not valid for commercial purposes). The most important criterion is usually Test P1 for the *primary response,* which requires that the station, under defined conditions, achieves at least 90% of the demanded step power change within 10s of initiation. Table 1 also shows that the over-shoot P_2 must not exceed 5% and the initial negative excursion P_6 (undershoot), associated with the non-minimum phase response, must not exceed 2%.

Table 1: Specification of step response for advanced control design at Dinorwig.

Test	Specification for single unit operation.	Single unit response with current governor.
P1	$P_1 \geq 90\%$ at $t_{p1} = 10s$	81% at 10s, 90% at 13.7s
P2	$P_2 \leq 5\%$ and $t_{p2} \leq 20s$	No overshoot
P3	$t_{p3} = 25s$ for $P_3 \leq 1\%$	25.9s
P4	$t_{p4} = 60s$ for $P_4 \leq 0.5\%$	29.2s
P5	$t_{p5} = 8s$	12.1s
P6	$P_6 = 2\%$	1.75%
P7	$t_{p7} = 1.5s$	0.88s

The neural PDF controller was connected to the nonlinear model of the hydroelectric power plant. The model is expressed in the per-unit system,

normalized to 300 MW and 50 Hz, and assumes a Grid system with infinite busbars. A PI controller with parameters fixed at K=0.1 and T_i=0.12 (as currently implemented in practice) is used as a basis of comparison. Figure 9 shows small step responses (0.04 p.u.) of hydroelectric plant under PI and neural PDF controllers for one unit operational. Figure 10 shows small step responses (0.04 p.u.) of the power station when six units are connected. In both cases, the hydroelectric plant shows a better performance under neural PDF controller; the response under the neural PDF controller is 10% and 30% faster in one unit operational and six units operational, respectively. The undershoot is also reduced in both cases when a PDF controller is driven the process.

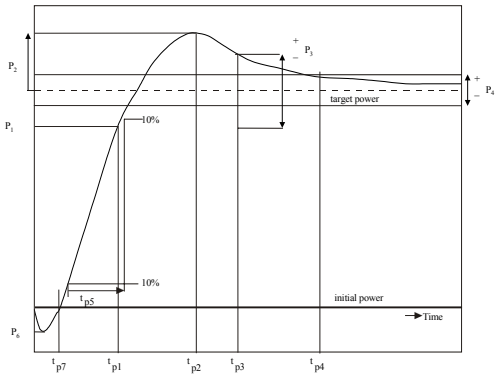


Figure 8: Specifications for a response to a step change in demanded power.

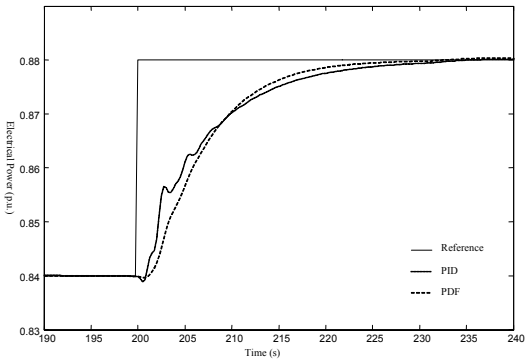


Figure 9: Step response of hydro plant under neural PDF and PI controllers with one unit operational.

The ramp response specification for single unit operation is expressed in Figure 11 and Table 2. Again, the most important criterion is usually Test Q1 for the *primary response (t_{q1}),* which requires that the station, under defined conditions, achieves at least 90% of the demanded power change, ramp amplitude (A_r), within 15s of initiation. Table 2 also shows that the maximum rate Q_2 must not be less than 90% of the ramp rate and the steady-state accuracy Q_3 must not be longer than 30s. Test Q_4 shows the effective under-delivery of power over the period of the ramp (Jones, Mansoor, Aris, Jones, Bradley, & King, 2004). The ramp response of the nonlinear elastic model of Dinorwig is shown in Figure 11.

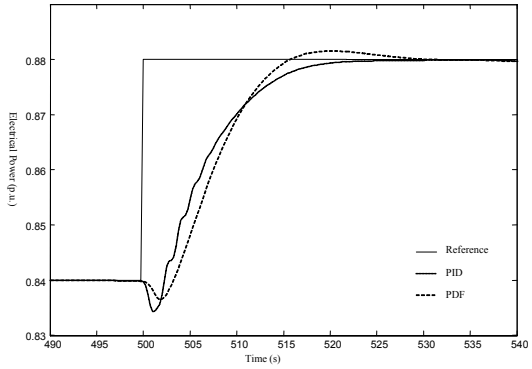


Figure 10: Step response of hydro plant under neural PDF and PI controllers with six units operational.

Table 2: Specification of ramp response for advanced control design at Dinorwig.

Test	Specification for a single unit operation	Single unit response with current PI control
Q_1	$Q_1 \geq 90\%$ at t_{q1}=15s	14.7
Q_2	Q_2=90% of 6 MWs^{-1}	1.8 MWs^{-1}
Q_3	t_{q3}=30s for $Q_3 \leq 1\%$	27
Q_4	None specified	E(RMS)=3.09 MW for t_{q4}=50s

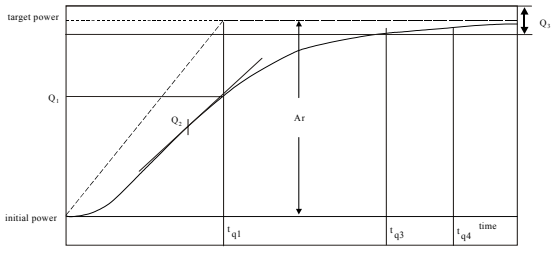


Figure 11: Specification for a ramp input power target.

Figure 12 shows large ramp responses (0.3 p.u.) of hydroelectric plant under PI and neural PDF

controllers for one unit operational. Figure 13 shows large ramp responses (0.3 p.u.) of the power station when six units are connected. In both cases, the hydroelectric plant shows a better performance under neural PDF controller; the response under the neural PDF controller is 15% and 13% faster in one unit operational and six units operational, respectively. When a PDF controller is driven the plant, the under-shoot is also reduced for both cases.

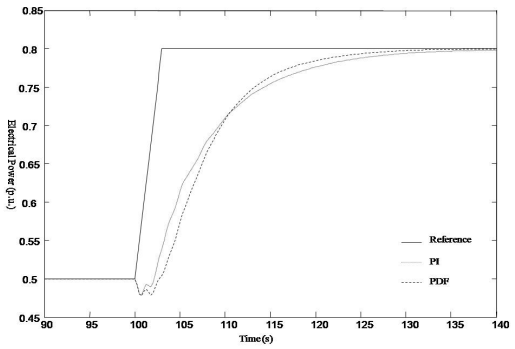


Figure 12: Large ramp response of hydro plant under neural PDF and PI controllers with one unit operational.

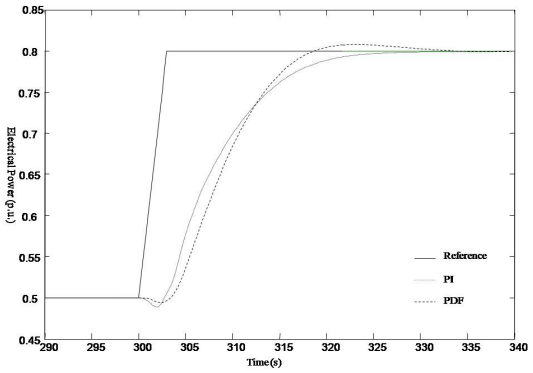


Figure 13: Large ramp response of hydro plant under neural PDF and PI controllers with six units operational.

To evaluate the cross coupling interaction a 0.04 step was applied simultaneously at t=500 to units 2-6 and the perturbation of unit 1 were observed. Figure 14 shows that although the neural PDF response has a higher overshoot, the PI response has a longer settling time and a higher undershoots.

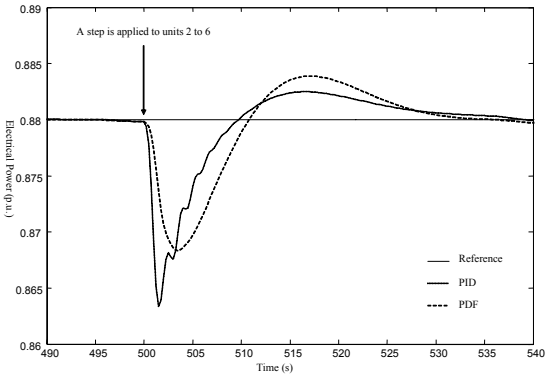


Figure 14: Cross coupling of hydro plant under PI and neural PDF controllers.

6 CONCLUSIONS

The results have shown how the neural PDF can be applied to a hydroelectric pumped-storage station to improve its dynamic response. In particular, this paper has shown that the step response of the system with neural PDF is improved. Multivariable effects have been taken into account to represent closely the real plant. The close relation between penstocks has been included into the nonlinear model. These are promising results for the use of neural PDF in this application and encourage us to address the issue of robustness of the response in future work.

ACKNOWLEDGEMENTS

The authors wish to thank First Hydro Company for their assistance. G. A. Munoz-Hernandez wishes to thank "*CONACyT*" and the "*Instituto Tecnológico de Puebla*" who have supported him in his postdoctoral work.

REFERENCES

Aguado Behar, A. (2000). *Topics on identification and adaptive control (in Spanish).* La Habana, Cuba: ICIMAF.

Djukanovic, M. B., Calovic, M. S., Vesovic, B. V., & Sobajic, D. J. (1997). Neuro-fuzzy controller of low head hydropower plants using adaptive-network based fuzzy inference system. *IEEE Transactions on Energy Conversion , 12*, 375-381.

Garcez, J. N., & Garcez, A. R. (1995). A connectionist approach to hydroturbine speed control parameters

tuning using artificial neural network. In IEEE (Ed.), *38th IEEE Midwest Symposium on Circuits and Systems*, *2*, pp. 641-644.

Jones, D. I., Mansoor, S. P., Aris, F. C., Jones, G. R., Bradley, D. A., & King, D. J. (2004). A standard method for specifying the response of hydroelectric plant in frequency-control mode. *Electric Power Systems Research , 68*, 19-32.

Kang, J. K., Lee, J. T., Kim, Y. M., Kwon, B. H., & Choi, K. S. (1991). Speed controller design for induction motor drivers using a PDF control and load disturbance observer. *IECON* (pp. 799-803). Kobe, Japan: IEEE.

Kundur, P. (1994). *Power System Stability and Control New York.* New York, USA: Mc Graw Hil.

Mansoor, S. P. (2000). *PhD. Thesis. Behaviour and Operation of Pumped Storage Hydro Plants.* Bangor, U.K.: University of Wales, Bangor.

Mansoor, S. P., Jones, D. I., Bradley, D. A., & Aris, F. C. (1999). Stability of a pumped storage hydropower station connected to a power system. *IEEE Power Eng. Soc. Winter Meeting* (pp. 646-650). New York, USA: IEEE.

Mansoor, S. P., Jones, D. I., Bradley, D. A., Aris, F. C., & Jones, G. R. (2000). Reproducing oscillatory behaviour of a hydroelectric power station by computer simulation. *Control Engineering practice , 8*, 1261-1272.

Minsky, M. L., & Papert, S. A. (1988). *Perceptrons: Introduction to Computational Geometry.* Cambridge, USA: MIT Press.

Narendra, K. S., & Mukhopadhyay, S. (1996). Adaptive control using neural networks and approximate models. *American Control Conference* (pp. 355-359). Seattle, USA: IEEE.

Rumelhart, D. E., McClelland, J. L., & group, T. P. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, USA: MIT Press.

Shu-Qing, W., Zhao-Hui, L., Zhi-Huai, X., & Zi-Peng, Z. (2005). Application of GA-FNN hybrid control system for hydroelectric generating units. *International Conference on Machine Learning and Cybernetics. 2*, págs. 840-845. IEEE.

Werbos, P. J. (1974). *PhD. Thesis: Beyond regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* Cambridge, USA: Harvard University, Cambridge, MA.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *IRE WESCON Convention Record*, *4*, págs. 96-104.

Working group on prime mover energy supply, I. (1992). Hydraulic turbine and turbine control model for system dynamic studies. *IEEE Transactions on Power Systems , 7*, 167-179.

Yin-Song, W., Guo-Cai, S., & Ong-Xiang. (2000). The PID-type fuzzy neural network control and it's application in the hydraulic turbine generators. *Power Engineering Society meeting. 1*, págs. 338-343. IEEE.

GPC AND NEURAL GENERALIZED PREDICTIVE CONTROL

S. Chidrawar[1] Nikhil Bidwai[2], L. Waghmare[2] and B. M. Patre[2]
[1]*MGM's College of Engineering, Nanded (MS) 431 602, India*
[2]*SGGS Institute of Engineering and Technology, Nanded (MS) 431 606, India*
sadhana_kc@rediffmail.com

Keyword: Neural network, Model predictive control, GPC, NGPC.

Abstract: An efficient implementation of Model Predictive Control (MPC) using a multilayer feed forward network as the plants linear model is presented. This paper presents a comparison between the Generalized Predictive Control and Neural Generalized Predictive Control with Newton-Raphson as minimization algorithm. Three different linear models are taken and their performances are tested. Simulation result shows the effect of neural network on Generalized Predictive Control for linear systems. The performance comparison of these system configurations has been given in terms of ISE and IAE.

1 INTRODUCTION

Model predictive control (MPC) has found a wide range of applications in the process, chemical, food processing, automotive, aerospace, metallurgy, and pulp and paper industries. (Qin and Badgwell, 2003; Yu, Yu and Gomm, 2006; Lawrynczuk, 2007). In recent years, the requirements for the quality of automatic control in the process industries increased significantly due to the increased complexity of the plants and sharper specifications of product quality. As a result, computer models that are computationally expensive became applicable even to rather complex problems. Intelligent and model based control techniques were developed to obtain tighter control for such applications. Neural network techniques has been found to be particularly useful for modeling and controlling highly uncertain nonlinear and complex systems. (Noorgard, Ravn, Poulsen and Hansen, 2000). The Model Predictive Control (MPC) techniques found to be very effective in control systems. MPC was introduced successfully in several industrial plants. Some of the most popular MPC algorithms that found a wide acceptance in industry are Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC), Predictive Functional Control (PFC), Extended Prediction Self Adaptive Control (EPSAC), Extended Horizon Adaptive Control (EHAC) and Generalized Predictive Control (GPC). (Morari and Lee, 1999, Rossiter, 2003). In this work, comparison between GPC and Neural GPC has been carried out for linear systems. The results show the efficacy of NGPC for such plants.

2 GENERALIZED PREDICTIVE CONTROL

The GPC method was proposed by Clarke et. al. (Clarke, Mohatadi and Tuffs, 1987) and has become one of the most popular MPC methods both in industry and academia.

The basic idea of GPC is to calculate a sequence of future control signals in such a way that it minimizes a multistage cost function defined over a prediction horizon. The index to be optimized is the expectation of a quadratic function measuring the distance between the predicted system output and some reference sequence over the horizon plus a quadratic function measuring the control effort. The GPC scheme consists of the plant to be controlled, a reference model that specifies the desired performance of the plant, a linear model of the plant, and the Cost Function Minimization (CFM) algorithm that determines the input needed to produce the plant's desired performance. The GPC system starts with the input signal, *r(t),* which is presented to the reference model. This model produces a tracking reference signal, *w (t)* that is used as an input to the CFM block. The CFM algorithm produces an output, which is used as an input to the plant. Between samples, the CFM

algorithm uses this model to calculate the next control input, *u(t+1)*, from predictions of the response from the plant's model. Once the cost function is minimized, this input is passed to the plant.

3 NEURAL GENERALIZED PREDICTIVE CONTROL

The ability of the GPC to make accurate predictions can be enhanced if a neural network is used to learn the dynamics of the plant instead of standard nonlinear modeling techniques.(Noorgard, Ravn, Poulsen and Hansen, 2000). The selection of the minimization algorithm affects the computational efficiency of the algorithm. In this work Newton-Raphson method is used as the optimization algorithm. The main cost of the Newton-Raphson algorithm is in the calculation of the Hessian, but even with this overhead the low iteration numbers make Newton-Raphson a faster algorithm for real-time control. (Soloway, 1996). The Neural Generalized Predictive Control (NGPC) system can be seen in Fig. 1. It consists of four components, the plant to be controlled, a reference model that specifies the desired performance of the plant, a neural network that models the plant, and the Cost Function Minimization (CFM) algorithm that determines the input needed to produce the plant's desired performance. The NGPC algorithm consists of the CFM block and the neural net block.

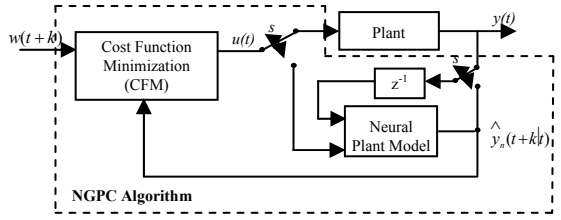


Figure 1: Block Diagram of NGPC System.

The NGPC system starts with the input signal, *r(t),* which is presented to the reference model. This model produces a tracking reference signal, *w(t+k),* that is used as an input to the CFM block. The CFM algorithm produces an output that is either used as an input to the plant or the plant's model. The double pole double throw switch, *S*, is set to the plant when the CFM algorithm has solved for the best input, *u(t)*, that will minimize a specified cost function. Between samples, the switch is set to the plant's model where the CFM algorithm uses this model to calculate the next control input, *u(t+1)*, from predictions of the response from the plant's model. Once the cost function is minimized, this input is passed to the plant. The computational performance of a GPC implementation is largely based on the minimization algorithm chosen for the CFM block. Models using neural networks have been shown to have the capability to capture nonlinear dynamics**.** Improved predictions affect rise time, over-shoot, and the energy content of the control signal.

3.1 Formulation of NGPC

3.1.1 Cost Function

As mentioned earlier, the NGPC algorithm (Soloway, 1996) is based on minimizing a cost function over a finite prediction horizon. The cost function of interest to this application is

$$J(N_1,N_2,N_u)=\sum_{j=N_1}^{N_2}\delta(j)[\hat{y}(t+j\,|\,t)-w(t+j)]^2+\sum_{j=1}^{N_u}\lambda(j)[\Delta u(t+j-1)]^2 \quad (1)$$

N_1 = Minimum Costing Prediction Horizon

N_2 = Maximum Costing Prediction Horizon

N_u = Length of Control Horizon

$\hat{y}(t+k|t)$ = Predicted Output from Neural;

Network, $u(t+k|t)$ = Manipulated Input

$w(t+k)$ = Reference Trajectory

δ and λ = Weighing Factor

When this cost function is minimized, a control input that meets the constraints is generated that allows the plant to track the reference trajectory within some tolerance. There are four tuning parameters in the cost function, N_1, N_2, N_u, and λ. The predictions of the plant will run from N_1 to N_2 future time steps. The bound on the control horizon is N_u. The only constraint on the values of N_u and N_1 is that these bounds must be less than or equal to N_2. The second summation contains a weighting factor, λ that is introduced to control the balance between the first two summations. The weighting factor acts as a damper on the predicted *u(n+1)*.

3.1.2 Cost Function Minimization Algorithm

The objective of the CFM algorithm is to minimize *J* in Equation (1) with respect to $[u(n+1), u(n+2), \ldots, u(n+N_u)]^T$, denoted as *U*. This is accomplished by setting the Jacobian of Equation (1) to zero and solving for *U*. With Newton-Rhapson used as the CFM algorithm, *J* is minimized iteratively to

determine the best *U*. An iterative process yields intermediate values for *J* denoted *J(k)*. For each iteration of *J(k)* an intermediate control input vector is also generated and is denoted as:

$$U(k) \stackrel{\Delta}{=} \begin{bmatrix} u(t+1) \\ u(t+2) \\ . \\ . \\ . \\ u(t+N_u) \end{bmatrix} \quad k=1,....N_u \tag{2}$$

Using this Newton-Raphson update rule, $U(k+1)$ is

$$U(k+1) = U(k) - \left(\frac{\partial^2 J}{\partial U^2}(k) \right)^{-1} \frac{\partial J}{\partial U}(k)'$$

$$\text{where } f(x) = \frac{\partial J}{\partial U} \tag{3}$$

where the Jacobian is denoted as

$$\frac{\partial J}{\partial U}(k) \equiv \begin{bmatrix} \frac{\partial J}{\partial u(t+1)} \\ . \\ . \\ . \\ \frac{\partial J}{\partial u(t+N_u)} \end{bmatrix} \tag{4}$$

and the Hessian as

$$\frac{\partial^2 J}{\partial U^2}(k) \equiv \begin{bmatrix} \frac{\partial^2 J}{\partial u(t+1)^2} & . & . & \frac{\partial^2 J}{\partial u(t+1)\partial u(t+N_u)} \\ . & . & . & . \\ . & . & . & . \\ \frac{\partial^2 J}{\partial u(t+N_u)\partial u_k(t+1)} & . & . & \frac{\partial^2 J}{\partial u(t+N_u)^2} \end{bmatrix} \tag{5}$$

The each element of the Jacobian is calculated by partially differentiating (4) with respect to vector *U*.

3.1.3 Neural Network Architecture

In NGPC the model of the plant is a neural network. This neural model is constructed and trained using MATLAB Neural Network System Identification Toolbox commands and Control System Design Toolkit (Noorgard, Ravn, Poulsen and Hansen, 2000).

The output of trained neural network is used as the predicted output of the plant. This predicted output is used in the Cost Function Minimization Algorithm. If $y_n(t)$ is the neural network's output then it is nothing but plant's predicted output $\hat{y}_n(t+k|t)$.

The initial training of the neural network is typically done off-line before control is attempted

3.1.4 Prediction using Neural Network

The NGPC algorithm uses the output of the plant's model to predict the plant's dynamics to an arbitrary input from the current time, *t,* to some future time, *t+k.*

4 SIMULATION RESULTS

The objective of this study is to show how GPC and NGPC implementation can cope with linear systems. GPC is applied to the systems with changes in system order. The Neural based GPC is implemented using MATLAB Neural Network Based System Design Toolbox (Noorgard, Ravn, Poulsen and Hansen, 2000).

4.1 GPC and NGPC for Linear Systems

The GPC and NGPC algorithm was applied to the different linear models with varying system for simulation purpose. For all the systems Prediction Horizon N_1 =1, N_2 =7 and Control Horizon (N_u) is 2 have been considered. The weighing factor λ for control signal is kept to 0.3 and δ for reference trajectory is set to 0. The same controller setting is used for all the systems simulation. The following simulation results are obtained showing the plant output when GPC and NGPC are applied. Also the required control action for different systems is shown.

System I. The GPC and NGPC algorithms are applied to a second order system (6). Fig. 2 shows the plant output with GPC and NGPC. Fig. 3 shows the control efforts taken by both controllers.

$$G(s) = \frac{1}{1+10s+40s^2} \tag{6}$$

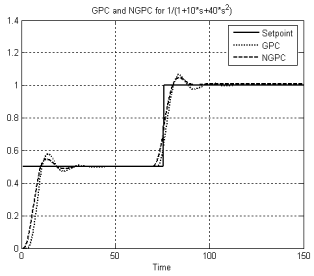


Figure 2: System I Output using GPC and NGPC.

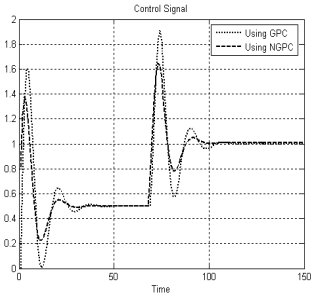


Figure 3: Control Signal for System I.

System II. A simple first order system (7) is controlled. Fig. 4 and Fig. 5 show the system output and control signal.

$$G(s) = \frac{1}{1+10s} \tag{7}$$

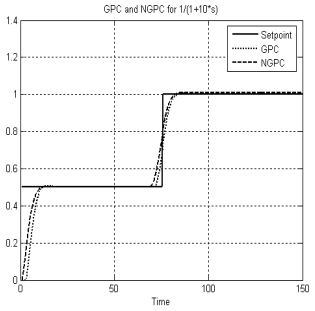


Figure 4: System II Output using GPC and NGPC.

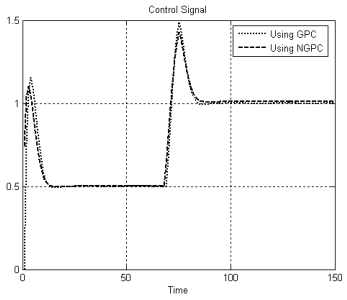


Figure 5: Control Signal for System II.

System III. A second order system (8) is controlled using GPC and NGPC. Fig.6 and Fig.7 Show the predicted output and control signal.

$$G(s) = \frac{1}{10s(1+2.5s)} \tag{8}$$

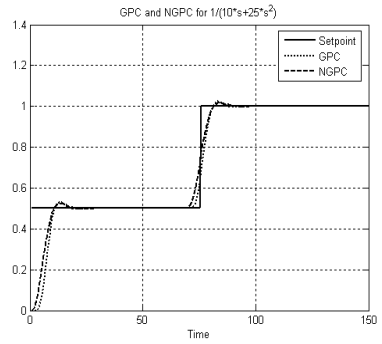


Figure 6: System III Output using GPC and NGPC.

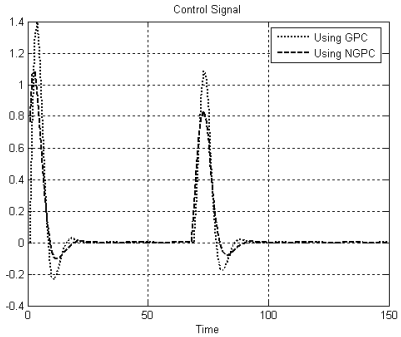


Figure 7: Control Signal for System III.

Initially systems were trained using Levenberg-Marquardt learning algorithm. Fig. 8 shows input data applied to the neural network for offline training purpose and corresponding neural network output.

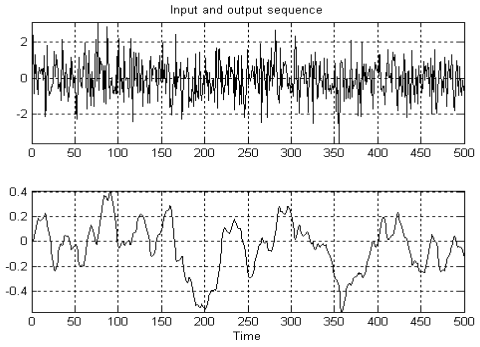


Figure 8: Input and output data for NN.

Performance evaluation of both the controller is carried out using ISE and IAE criteria. Table 1 gives ISE and IAE values for both GPC and NGPC implementation for all the linear systems considered. It was observed that for each system ISE and IAE using NGPC is smaller or equal to GPC.

Table 1: ISE and IAE Performance Comparison of GPC and NGPC for Linear System.

Systems	Setpoint	GPC		NGPC	
		ISE	IAE	ISE	IAE
I	0.5	1.827	4.4107	1.6055	3.6351
	1	0.2567	1.4492	0.1186	1.4312
II	0.5	1.1803	3.217	0.7896	2.6894
	1	0.1311	0.767	0.063	1.017
III	0.5	1.4639	3.7625	1.1021	3.3424
	1	0.1759	0.9065	0.0957	0.7062

5 CONCLUSIONS

In this paper a comparison between GPC and NGPC is carried out for linear systems. The performance of NGPC is better that GPC in terms of ISE and IAE Performance index.

REFERENCES

Qin, S. J., Badgwell, T., 2003. A survey of industrial model predictive control technology. *Control Engineering Practice.* Vol. 11. no. 7. pp. 733-764.

Yu, D. L., Yu, D. W., Gomm, J. B., 2006. Neural model adaptation and predictive control of a chemical rig. *IEEE Transaction on Control Systems Technology.* Vol. 14. no. 5. pp. 828-840.

Lawrynczuk, M., 2007. An efficient nonlinear predictive control algorithm with neural models based on multipoint on-line linearization. EUROCON 2007. *The International Conference on "Computer asa Tool. Warsaw. September 9-12. pp. 777-784*

Noorgard, M., Ravn, O., Poulsen, N. K., Hansen, L. K. 2000. *Neural networks for modeling and control of dynamical systems.* Springer, London.

Morari, M., Lee, J. 1999. Model predictive control: past, present and future. *Computers and Chemical Engineering.* Vol. 23. no. 4/5. pp. 667- 682

Rossister, J. A. 2003. *Model based predictive control.* CRC Press. Boca Raton.

Clarke, D. W., Mohtadi, C., Tuffs, P. S. 1987. Generalized predictive control- Part I, The basic algorithm. *Automatica.* Vol. 23. pp. 137-148.

Soloway, D. 1996. Neural generalized predictive control. *Proceedings of the 1996 IEEE International Symposium on Intelligent Control*. Dearborn, September, 15-18. pp. 227-282.

MOTOR PARAMETERS INFLUENCE ON STABILITY OF DRIVE FOR INDUSTRIAL ROBOT

Sorin Enache, Monica Adela Enache, Mircea Dobriceanu
Mircea Adrian Drighiciu and Anca Petrisor
Electromechanical Faculty, University of Craiova, 107 Decebal Street, 200440 Craiova, Romania
{senache, menache, mdobriceanu, adrighiciu, apetrisor}@em.ucv.ro

Keywords: Electrical drive, industrial robot, stability, asynchronous motor, parameters.

Abstract: This paper analyzes a driving system for an industrial robot from the stability point of view. For doing this, an original analysis method has been conceived. The method has as starting point the two axes mathematical model with equations written in per unit values. A Matlab program has been conceived with their help; this program has led to results and conclusions detailed in this paper. Finally a series of experimental results confirming the conclusions deduced with the new method are presented.

1 INTRODUCTION

The induction motors have been recognized for a long time as being the most reliable electrical machines, allowing an easy maintenance and utilization in dangerous medium, being at the same time cheap, easily to be built and having a high power/weight ratio. Yet, the speed adjustment of the squirrel cage induction motors is made with the help of some relatively complicated static equipments, of the voltage and frequency static converters.

The high cost of these ones is in a permanent decrease owing to the achievements from the field of power electronic components. Concomitantly with the progresses of the power electronics, we watch the introduction of the microelectronics in the command and adjustment part of the power converters.

Owing to the flexibility provided by the microcomputers programming, it is possible to achieve solutions of high complexity and reliability. Owing to these progresses there have been implemented some new multi-variable adjustment techniques, field orientation, field accelerating etc.

All these aspects made the induction motor to conquer new positions in variable speed drives.

This way, the power electronics has practically reduced the problem of the use extension of the induction motor supplied from a voltage and frequency static converter, to a problem of investment cost and economic efficiency.

The ever greater utilization of the induction motor as an execution element in the automatic systems has imposed an ever larger approach, in the speciality papers, of the problems concerning the dynamic regime of it.

This paper joins to this context, analyzing a few aspects regarding the dynamic processes when operating with variable frequency. So, first of all it is necessary to use an adequate mathematical model for performing a study in this field. The two axes theory models are discussed in the analyzed case.

Starting from the conclusion that the stability is a qualitative feature of the systems associated to their dynamic behaviour, it also results that it is necessary to analyze the stability of the converter-motor-robot assembly and the motor parameters influence on it.

2 MATHEMATICAL MODEL

The equations system that is used has the following form (Enache, 2005):

$$\omega_s^* = s_{ks}\left(\underline{\Psi}_s^* - k\underline{\Psi}_r^{/*}\right) + \frac{d\underline{\Psi}_s^*}{dt^*} + j\omega_s^*\underline{\Psi}_s^*$$

$$0 = s_{kr}\left(\underline{\Psi}_r^{/*} - k\underline{\Psi}_s^*\right) + \frac{d\underline{\Psi}_r^{/*}}{dt^*} + j\left(\omega_s^* - \omega^*\right)\underline{\Psi}_r^{/*} \quad (1)$$

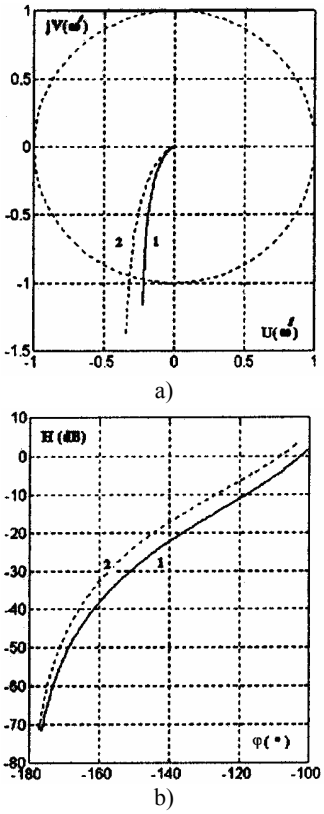
$$h\cdot\frac{d\omega^*}{dt^*} = -\frac{k}{x_{rt}^{/*}}\mathrm{Im}\left[\left(\underline{\Psi}_s^*\right)^*\underline{\Psi}_r^{/*}\right] - m_r^*$$

where m_r is the resistant torque corresponding to the industrial robot.

3 SIMULATIONS: QUANTITATIVE RESULTS

A Matlab program for the stability analysis has been conceived.

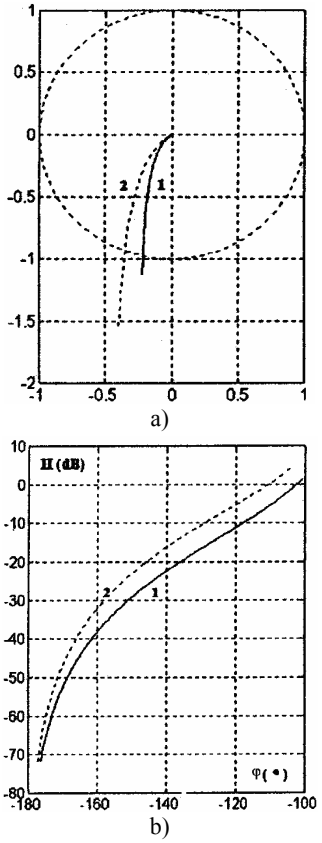
The representations from Figs. 1, 2 and 3 have been obtained by running this program, for the concrete case of a motor rated at 1,1 kW.



a)

b)

Figure 1: Transfer locus (a) and amplitude-phase characteristics (b) obtained in the case of the inductances modification: $L_S = 0{,}529$ H (1) and $L_S = 0{,}549$ H (2).

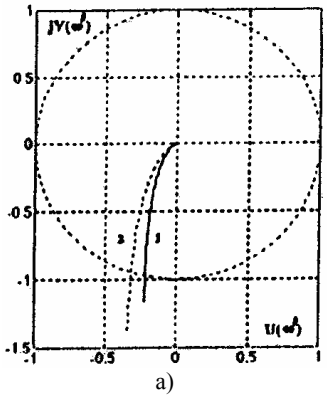


a)

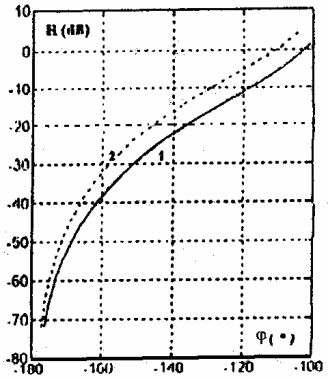
b)

Figure 2: Transfer locus (a) and amplitude-phase characteristics (b) obtained in the case of the inductances modification: $L_r^{/} = 0{,}528$ H (1) and $L_r^{/} = 0{,}548$ H (2).

There must be also mentioned the importance of the introduced method resulting from the possibility to emphasize the machine parameters influence and especially the inertia moment influence, on stability when operating at variable frequency, fact that provides originality to this method.



a)



b)

Figure 3: Transfer locus (a) and amplitude-phase characteristics (b) obtained in the case of the inductances modification: $L_{sh} = 0{,}498$ H (1) and $L_{sh} = 0{,}558$ H (2).

In order to catch quantitatively these interdependences the following table has been filled.

Table 1: Absolute values and phase margins.

Param.	Absolute value	Per unit param.	Per unit value	Phase margin [^{0}]
L_s	0,549	x_s^*	2,2735	69,13
$L_r^/$	0,548	$x_r^{/*}$	2,2694	75,31
L_{sh}	0,558	x_{lm}^*	2,3104	71,32

These results help us to emphasize a few important conclusions regarding the resistances influence on the studied system stability:

- the increase of the inductance L_s leads to the stability decrease;
- at the same time with the rotor inductance increase the system stability decreases;
- the increase of the main inductance has a stabilizing effect.

4 EXPERIMENTAL CIRCUIT

In order to confirm the previous conclusions, a series of experimental tests have been performed; a few of them are detailed further on.

Thus, the experimental circuit has the structure depicted in the following figure.

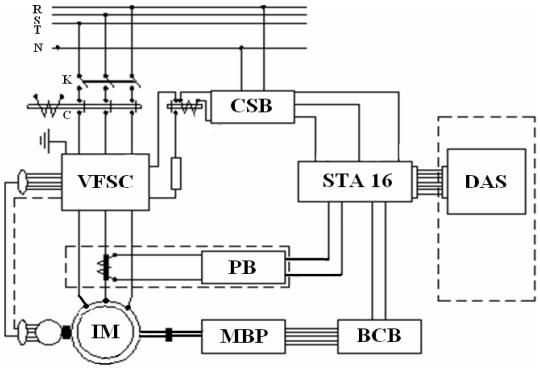


Figure 4: Scheme of the experimental circuit.

The notations have the following meaning:

IM – induction motor;
VFSC – voltage and frequency static converter;
DAS – data acquisition board;
CSB – command and synchronization block;
PB – protection block;
MPB – magnetic powder break;
BCB – brake command block;
STA 16 – connection block.

A picture of this circuit is depicted for conformity.

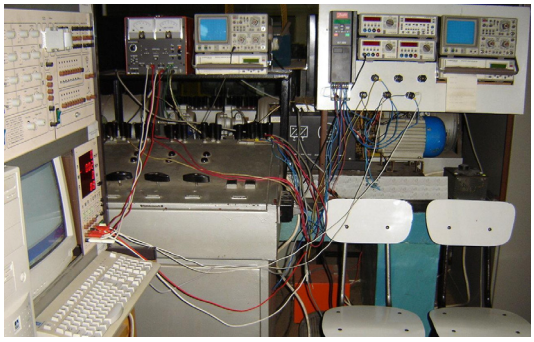


Figure 5: Picture of experimental circuit.

In order to obtain the determinations in dynamic regime the experimental circuit depicted before has been carried out, having a data acquisition board DAS as a central element (Enache, 2007). This high speed analogical and digital interface has been assembled inside a computer. Both the acquisition and the adequate data processing are controlled with the help of a program conceived in Matlab.

5 EXPERIMENTAL RESULTS

A series of graphic results have been obtained with the help of the acquisition program; the following figures are depicted further on.

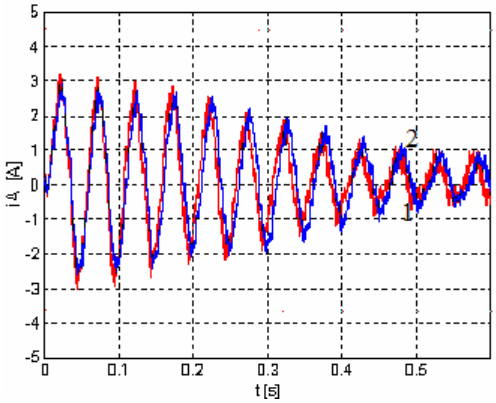


Figure 6: Graphic dependences corresponding to the cases L_s=0,529 H (1) and L_s=0,549 H (2).

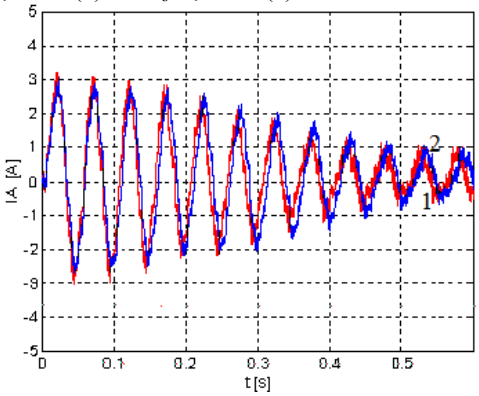


Figure 7: Graphic dependences corresponding to the cases $L_r^{/}$ =0,528 H (1) and $L_r^{/}$ =0,548 H (2).

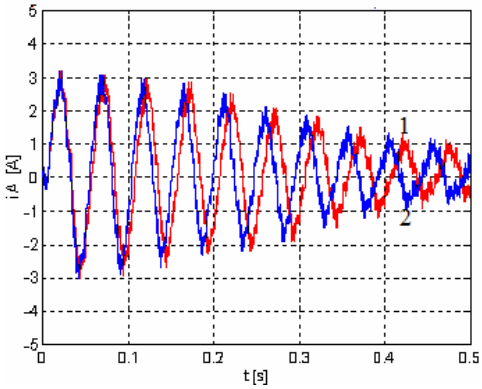


Figure 8: Graphic dependences corresponding to the cases L_{sh}=0,498 H (1) and L_{sh}=0,438 H (2).

These graphics lead to the following conclusions:

- when the value of the stator inductance increases the transient process duration increases (the stability decrease);
- the increase of the rotor inductance also involves the increase of the transient process duration (the stability decreases);
- the decrease of the main inductance value determines a faster stabilization of the process (stability increase).

These conclusions confirm the theoretical analysis performed before.

REFERENCES

Delemontey, B., Jacquot, B., Iung, C., De Fornel, B., Bavard, J., 1995. Stability Analysis and Stabilisation of an Induction Motor Drive with Input Filter. In: *Proccedings of 6th European Conference on Power Electronics and Applications*, Sevilla, Spain, pp. 121-126.

Enache, S., Vlad, I., Enache, M.A., 2005. Considerations Regarding Influences of Induction Motor Resistances upon Stability in case of Operation at Variable Frequency. In: *Proccedings of SIELMEN'05*, Chisinau, Moldova, Vol.2, ISBN GEN 973-716-208-0, 2005, pp 660-663.

Enache, S., Prejbeanu, R., Campeanu, A., Vlad, I., 2007. Aspects Regarding Simulation of the Saturated Induction Motors Control by the Voltage Inverter Commanded in Current. In: *Proceedings of IEEE Region 8, EUROCON2007*, Warsaw, Poland, 9-12 september 2007, IEEE Catalog Number: 07EX1617C, ISBN:1-4244-0813-X, Library of Congress: 2006937182, p. 1826.

Enache, S., Campeanu, A., Vlad, I., Enache, M.A., 2007. A New Method for Induction Motor Stability Analysis when Supplying at Variable Frecquency,. In: *Proceedings of ISTASC'07*, Athens p.126.

Lin, J.L., Shiau, L.G., 2000. On Stability and Performance of Induction Motor Speed Drives with Sliping Mode Current Control. *Asian Journal of Control*, Vol. 2, No. 2, pp 122-131.

Lipo, T., 1993 Stability Analysis of a Rectifier-inverter Induction Motor Drive. *IEEE Transaction on Industry Applications*, 88, 1, 1993, pp 57-68.

Subrahmanyam, V., Surendram, K., 1996. Stability Analysis of Variable Frequency Induction Motors Using D - Decomposition Method. *Electric Machines and Power Systems*, 11, pp. 105-112.

Suwankawin, S., Sangwongwanich, S., 2002. A speed-sensorless IM drive with decoupling control and stability analysis of speed estimation. IEEE *Transactions on Industrial Electronics*, vol. 49, no. 2, pp 444-455.

Vas, P., 1990. *Vector Control of A.C. Machines*, Clarendon Press.

DAS 1601 - Operating Manual, Keithley Metrabyte, 2002.

MATHEMATICAL MODELLING OF THERMAL AREA IN CUTTING TOOL

Daschievici Luiza, Ghelase Daniela and Goanta Adrian
Dunarea de Jos University Galati, Faculty of Engineering from Braila, Calea Calarasilor, nr. 29, Braila, Romania
Luiza.Tomulescu@ugal.ro, Daniela.Ghelase@ugal.ro, Adrian.Goanta@ugal.ro

Keywords: Cuttings, friction, heating, wearing.

Abstract: Since experimental researches regarding cutting process have stated a proportionality dependence of wear medium intensity on cutting area temperature and because this fact was avoid or ignored by thorough studies and researches, we considered to be helpful developing a physical-mathematical model able to correlate the two phenomena: wear and temperature in the cutting area. The complete and correct research on thermal phenomena in the cutting area is possible only by taking into consideration the feed-back relation between the physical and phenomenological elements of the studied tribosystem and also, by taking into account the splinter movement, resulting in a continuous supplying with cold layers of the splinter area and in heat evacuating by warm splinter movement.

1 RESEARCH ON METAL CUTTING, SPECIFIC PROCESSES AND PHENOMENA

Metal cutting, due to the action of tool blade pushed on the processed material, leads to a complex status of forces and deformations in the cutting area.

This simple diagram is the base for any detailed study on cutting process. It is a basic physical model, completing other specific models of elasticity and plasticity, thermodynamics, tribology, thus forming a complex model, more or less detailed according to requirements or claims. For almost 100 years of researching on cutting process, many theoretical and experimental data have been gathered, but they are far beyond from being a unitary whole.

The present paper develops on this quite applicable, enough explored, but still insufficiently understood field of application.

Based on the knowledge, the definition of the splinter appearance process has been stated, containing also the essence of interdisciplinary aspects.

The essence of these aspects is already shown in the specialty literature, in detail, and this paper deals only on those aspects considered able to be improved.

The analysis of some definite aspects solving manner presented in the specialty literature led to the conclusion that errors have been done, errors able to stray from the real status.

The main phenomena and processes accompanying the cuting process are friction, tribosystem heating and tool wear. The tribosystem, consisting of tool, processed part and splinter, can be considered as a system with direct contact, where no lubricating film is present. Based on his own researches and taking into account some results of his predecessors, Coulomb stated the three dry friction rules, which are often partially confirmed. It can be reminded that friction depends on the relative speed, this fact being experimentally ascertained at the beginning of the XIX century, as shown in figure 1 (Tomulescu, 2000).

The experimental researches dawn up so far led to the conclusion that dry friction force depends on many factors such as: the normal pressure force, the relative sliding speed, the type of the contact and the value of the contact area, the quality and the roughness of the surface, the nature of materials in contact, the character of the friction areas: rigid or elastic, tenacious or fragile. Even human and materials resources were highly implied in research, the complex friction phenomenon did not allow a universal valid theory elaboration, for at least from the quality point of view.

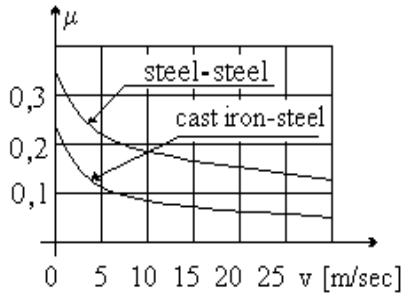


Figure 1: The variation of friction coefficient on speed.

Taking into account the prevalent role of friction on tribosystem parts heating, thus influencing the cutting tool wear, one considered necessary to do a research on he friction coefficient for a specific cutting case and highlight the fact that the friction coefficient is a non-coulombian one (Tomulescu, 1999; Tomulescu, 2003).

As for the dry or lubricated cutting friction, its framing in the dry friction category is based on the finding that the cooling fluids do not enter into the friction area.

The researches shown that the mechanical work necessary for volume plastic deformation in the cut metal layer is partially transformed into caloric energy that heats the cutting area; this process depends on the thermal conductivity of the cut metal and on the working conditions of cutting process.

The mechanical work consumed in the cutting process can be expressed by the next equation (Dumitras, 1983):

$$L=L_1+L_2+L_3+L_4+L_5+L_6 \quad (1)$$

where:

L_1, L_2 and L_3 are parts of the mechanical work transforming into heat during cutting and dispersed in the environment through the splinter, the tool and the processed part;

L_4, L_5 and L_6 are parts of mechanical work, consumed in the cutting process, with little values comparative with L_1, L_2 , L_3,.

The mechanical work consumed in the cutting process, entirely transformed in heat, L_1, L_2 and L_3 components, generates the heating sources Q_1, Q_2 and Q_3, fig. 2, ordered by their intensity, as follows: $Q_1>Q_2>Q_3$.

In fact, depending on the cutting process, different heat exchanges may develop between the processed part, the splinter and the tool, depending on the cutting process nature, so that the temperatures in the splinter, the tool and the processed part being not abided by the distribution of sources intensities.

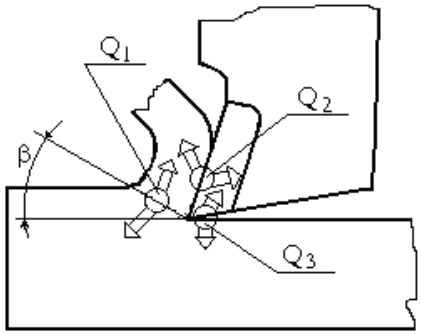


Figure 2: Mechanical work heat dispersing.

The factors influencing the heat generated by the cutting process are:

- the physical and mechanical characteristics of cut metal
- the cutting tool material
- the geometrical parameters of the tool
- the parameters of the cutting conditions

Since the blade temperature is the main wear-influencing factor, the researches went towards finding the empiric relations between blade temperature and the main cutting tool elements, less interest being paid to the aspect of heat quantities evaluation.

The blade wear behaviour depends in a large measure on its temperature. The experimental researches performed so far show that there is some proportionality between wear intensity and blade tool temperature, the variation rules demonstrating obvious parallelisms. That is why, in order to accurately evaluate wear intensity (and wear evolution) depending on cutting parameters, it is necessary to know the cutting temperature, including the tool blade temperature, depending on the cutting parameters.

Based on the researches performed, a series of dependence relations between the tribosystem elements temperatures have been defined.

The mathematic model shown by the specialty literature suits to an uniform stationary thermal status, respective to the same temperature in the entire splinter and tool blade and time constant, quite far from reality even for a qualitative and phenomenological analysis (Dumitras, 1983).

The hypothesis taken into consideration, according to which heating in the cutting area is uniform and stationary can be appreciated as a particular case that cannot be real.

That is why a specific model is required for the heating sources, able to lead to a correct

determination of real thermal status, non-stationary and non-uniformly distributed, that, in time, leads to a stationary and non-uniformly distributed thermal status, depending on the influencing factors, the most important of them being as follows:

- the cutting process parameters;
- the physical and mechanical properties of the tool blade and splinter;
- the environmental heat exchange;
- the feed-back interdependence between different influencing factors;
- the cutting process dynamic phenomena.

Cutting wear comes up because of interfering factors effects and it is important to be known, especially regarding the cutting tool.

Tool wear is progressive and it manifests under many aspects (temperature increasing, processed area deterioration, cutting forces increasing), finally leading to their stop functioning.

The researches highlighted more blade wear types, as shown in figure 3 that represents these wear influences and makes possible an appreciation of their weight as part of total wear. The diagram shows that the abrasive wear has the highest influence; it is determined by the friction conditions of areas in contact: tool - processed part-splinter.

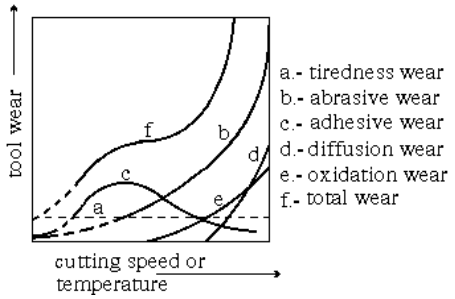


Figure 3: The influences of partial wear on total wear.

The splinter temperature increases due to the energy exclusively obtained from friction: the friction between the splinter and the tool and the friction between inter- and intracrystals that comes up during splinter formation and separation. The higher the temperature the more plastic is the splinter, some of its areas pass to the liquid phase, the intensity of the above mentioned frictions decreases, smaller amounts of energy are freed, splinter temperature decreases, the splinter is more solid, more intense frictions develop tending to increase the temperature, and so on. Therefore, a combination of effects with opposite tendencies takes place leading to a splinter temperature, which is not equal to the melting temperature of the processed part, but an equilibrium temperature beyond the melting one. Speed increasing, especially for high cutting speed, leads to a feedback chain, according to figure 4.

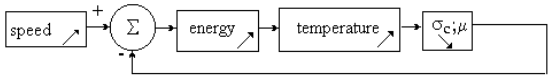


Figure 4: The feedback influence of temperature increasing on the mechanical characteristics of the processed material.

According to this chain, cutting speed increasing leads to cutting area temperatures increasing, the effect is a deformation resistance and mechanical work decreasing, thus implying wear reducing and a higher tool durability.

The analysis of cutting conditions influence on the tool temperature and tool wear leads to the conclusion that the variation of wear medium intensity is very similar to the variation of the temperature on the tool.

The main objective of the present paper is to combine in a single model the dynamic, thermal and tribologic phenomena, in order to evaluate in advance the wear intensity and to find out the way to influence it.

2 THE RESEARCH METHODOLOGY AND THE UTILIZED MANNERS

Among the considered objectives, there can be mentioned:

- the research of dynamic phenomena;
- the research of thermal and wear phenomena.

It was necessary to develop a physical model of the phenomenon in order to perform the researches; in fact, the phenomenon is a conventional image of the real status, representing the basics of mathematical modelling. The model has mathematical equations, functionally describing the physical model, and through it, the real phenomenon. For mathematical model solving, difficult to be analytically solved, there have been used numerical methods to obtain solutions for the differential equations (Tomulescu, 2000).

The stages covered for phenomenon modelling are, as follows:

- cutting area forces modelling, based on Merchant model for free orthogonal cutting, where is taken into consideration the fact that the splinter is

balanced by two categories of external and internal forces (Tomulescu, 2000);

- Heating Sources Modelling, figure 2;

It is thought that the heating source consists both by the non-conservative mechanical work wasted by plastic deformation in the cutting plane area and by the non-conservative mechanical work from the friction on the escaping and on the laying tool area.

- Heat Dispersing Modelling. The heat disperses into a non-homogeneous environment consisting of splinter, blade and tool body, each having different caloric coefficients, both as value and temperature dependence.

Solving the problem of heat dispersing under transitory conditions and in a heterogeneous environment leads to temperature knowing for every moment and in each point of the considered environment.

The theoretical study program, including the above-mentioned models, materialized in a very complex physical and mathematical computer model, enables the researching of the influence of different factors, such as:

- the parameters of the cutting conditions (speed, advance, depth);
- tool blade material;
- the cutting manner (continuous or interrupted).

As previously shown, the friction in the cutting area takes place under very particular conditions, such as high pressures, relative high-speed values, no lubrication. The bibliographical research shows that coulombian friction is an exaggerated approximation of dry friction.

As long as a realistic mathematical model is desired, the friction model for the cutting area should have a friction coefficient depending on speed for the couple splinter-blade.

By mathematical modelling, the differential equation for heat conducting is:

$$\rho \cdot c \cdot \frac{\partial \theta}{\partial t} = \lambda \cdot \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} + \frac{\partial^2 \theta}{\partial z^2} \right) \tag{2}$$

that for an anisotropic and non-homogeneous material generally turns to:

$$\frac{\partial}{\partial t}(\rho \cdot c \cdot \theta) = \frac{\partial}{\partial x}\left(\lambda_x \cdot \frac{\partial \theta}{\partial x} \right) + \frac{\partial}{\partial y}\left(\lambda_y \cdot \frac{\partial \theta}{\partial y} \right) + \frac{\partial}{\partial z}\left(\lambda_z \cdot \frac{\partial \theta}{\partial z} \right) \tag{3}$$

where:

ρ - material density (kg/m^3);

c - material specific heating (J/kgK);

λ_x, λ_y, λ_z - material thermal conductivity (W/mK).

The integration of the differential equation (3) is analytically difficult to solve, and the specialty literature does not offer exact solutions for each practical case. To obtain an analytical result, the following solution is used:

$$\theta(t,x,y,z) = T(t) \cdot F(x,y,z) \tag{4}$$

replacing it in heat equation (3) leads to:

$$\rho \cdot c \cdot \frac{\partial T}{\partial t} \cdot F(x,y,z) = T(t) \cdot \left[\lambda_x \cdot \frac{\partial^2 F(x,y,z)}{\partial x^2} \right] + T(t) \cdot \left[\lambda_y \cdot \frac{\partial^2 F(x,y,z)}{\partial y^2} + \lambda_z \cdot \frac{\partial^2 F(x,y,z)}{\partial z^2} \right] \tag{5}$$

Heat exchanges inside and on the areas of the tribosystem the mathematic model, knowing the border conditions, extremely difficult to be analytically described, materializes elements and that is why numerical integration is preferred, the most suited being the finite differences method.

The friction coefficient used by the computing program was experimentally determined by using energetic methods.

The results obtained for the friction coefficient clearly led to the conclusion that, in this case, the friction is non-coulombian (Tomulescu, 2003). Its dependence on speed is shown in figure 5.

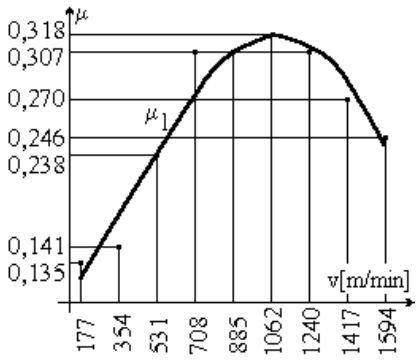
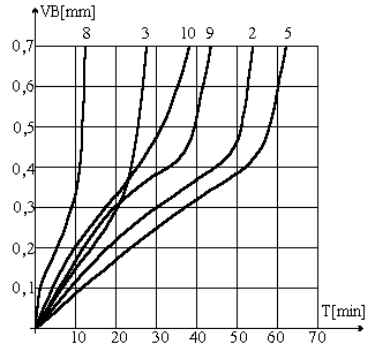


Figure 5: The variation of friction coefficient on relative speed.

The present paper also uses the results obtained by a classical research regarding tool wear, results taken out from a research project performed for manufacturing assimilation of metallic carbide cutting plates. These results were synthesized in wearing diagrams, VB = f(T), as shown in figure 6, for some specific processing cases. These wear curves, continuous in time, enabled the study of

wear evolution correlated to the suggested mathematical modelling.



Test	v[m/min]	s[mm/rot]	t[mm]
2	350	0,16	0,5
3	275	0,315	0,5
5	275	0,16	1
8	350	0,315	1
9,10	310	0,224	0,7

Figure 6: VB wear of TNGG 22.04.12/P10 plates for steel lathing 8550/97HB.

Taking into account the fact that the speed mostly determines process thermal status, with implications on cutting tool wear, it was considered necessary to be analyzed for the heating process, especially on the laying area, in order to diminish the implications and wear reducing.

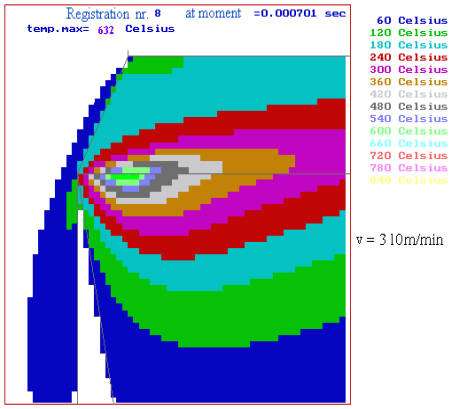


Figure 7: Thermal status in splinter-tool tribosystem.

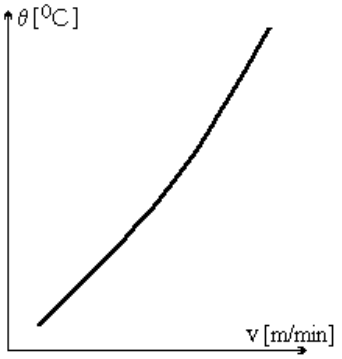


Figure 8: The variation of temperature on cutting speed.

By solving the mathematical model using the specialized developed program, thermal areas are obtained, as shown in figure 7; when analyzing for different cutting processes, with the required parameters *v*, *s* and *t*, appreciations on wear and durability of cutting tools can be stated. The obtained data can be used to trace a curve for temperature dependence on speed, as shown in figure 8 and the mathematic relation is:

$$\theta_{max} = 1001 - 2{,}38 \cdot v + 4{,}71 \cdot 10^{-3} \cdot v \qquad (6)$$

Notice the similitude between θ°[C] = f(v) curve and I_{med}=f(v) curve, figure 9; a direct relation I_{med}=f(θ°C) for case (P10) can be stated, as shown in figure 10 and mathematic relation proposed is:

$$I_{med} = -0{,}0634 + 7{,}2438 \cdot 10^{-5} \cdot \sqrt{21231 \cdot \theta - 1487056} + \\ + 9{,}7663 \cdot 10^{-5} \cdot \theta \qquad (7)$$

Diagram analysis concludes that there is a good proportionality of wear medium intensity with the maximum temperature of cutting process, an experimentally stated fact, also presented by the specialty literature.

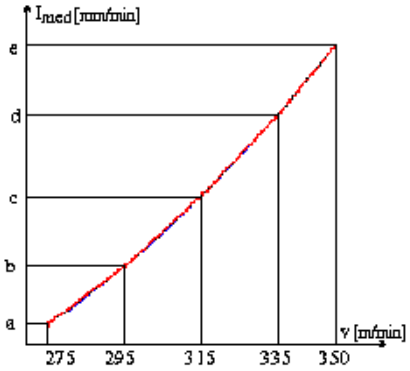


Figure 9: The variation of medium intensity on maximum temperature.

Experimental and theoretical researches regarding tool blade temperature and wear medium intensity settled up similitude between the evolutions of the two phenomena, leading to the conclusion that thermal phenomena evolution modelling enables the evolution of wear medium intensity, by applying a constant of proportionality, experimentally known, which remains the same for a couple splinter-blade.

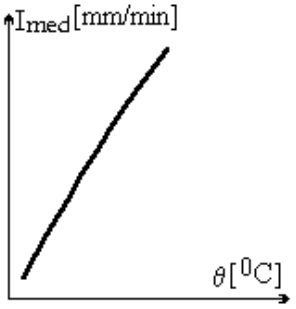


Figure 10: The variation of medium intensity on maximum temperature.

3 CONCLUSIONS

Based on the performed studies, a series of conclusions have been stated, among which are, as follows:

- the most important thermal sources, such as the source created by plastic deformations in the cutting plane and the source created by the friction between the splinter and the tool blade escaping area, have intensities and distributions depending on the values of the cutting conditions parameters and on the splinter-blade couple; they heat the cutting area to temperatures non-homogeneous distributed, and the temperatures influences those materials constants related to heating sources intensity;
- thermal status in the cutting area is characterized by a maximum in the splinter pressure center on the escaping area, as long as, in the wear area (the laying area) the temperature is much lower;
- it has been stated a dependence relation between a parameter characteristic tool blade wear, such as wear medium intensity, and cutting area temperature; the relation is, with pretty small deviations, a directly proportional dependence one; thus, by measuring the tribosystem temperature, the wear medium intensity can be evaluated, at least for the cutting process.

REFERENCES

Dumitras, C., Militaru, L., 1983. *Aschierea metalelor si fiabilitatea sculelor aschietoare.* Ed. Tehnica, Bucuresti.

L. Tomulescu, C. Falticeanu, Gh. Oproescu, 1999. *"Dynamic and thermal aspects of tribologic processes between tool blade and splinter during splitting"* - Balkantrib'99, The 3rd International Conference of Tribology, Sinaia,, ISSN 1221 - 9371, pg.233- 240.

Tomulescu, L., Ghelase, D., 2003. Experimental results obtained by friction process researching in processed part – tool tribosystem in cutting area. *International Science and Engineering Conference Machine - Building and Technosphere of the XXI Century* - Sevastopol, Ukraine, ISBN 966-7907-12-0, pg. 267 – 271

Tomulescu, L., 2000. *Tribologia taisului de scula aschietoare.* CEPROHART, Braila.

STOCHASTIC CONTROL STRATEGIES AND ADAPTIVE CRITIC METHODS

Randa Herzallah
Faculty of Engineering Technology, Al-Balqa' Applied University, Jordan
r_herzal@hotmail.com

David Lowe
NCRG, Aston University, U.K.
d.lowe@aston.ac.uk

Keywords: Adaptive critic methods, functional uncertainty, stochastic control.

Abstract: Adaptive critic methods have common roots as generalizations of dynamic programming for neural reinforcement learning approaches. Since they approximate the dynamic programming solutions, they are potentially suitable for learning in noisy, nonlinear and nonstationary environments. In this study, a novel probabilistic dual heuristic programming (DHP) based adaptive critic controller is proposed. Distinct to current approaches, the proposed probabilistic (DHP) adaptive critic method takes uncertainties of forward model and inverse controller into consideration. Therefore, it is suitable for deterministic and stochastic control problems characterized by functional uncertainty. Theoretical development of the proposed method is validated by analytically evaluating the correct value of the cost function which satisfies the Bellman equation in a linear quadratic control problem. The target value of the critic network is then calculated and shown to be equal to the analytically derived correct value.

1 INTRODUCTION

In recent research of stochastic control systems, much attention has been paid to the problem of characterizing and incorporating functional uncertainty of dynamical control systems. This is because there is an increasing demand for high reliability of complex control systems which are accompanied by high level of inherent uncertainty in modeling and estimation and are characterized by intrinsic nonlinear dynamics involving unknown functionals and latent processes. Several methods have been developed, and examples include feedback linearization techniques (Botto et al., 2000; Hovakimyan et al., 2001), backstepping techniques (Sastry and Isidori, 1989; Zhang et al., 2000; Lewis et al., 2000), neural network based methods (Wang and Huang, 2005; Ge and Wang, 2004; Ge et al., 2001; Murray-Smith and Sbarbaro, 2002; Fabri and Kadirkamanathan, 1998), stochastic adaptive control methods (Karny, 1996; Wang and Zhang, 2001; Wang, 2002; Herzallah and Lowe, 2007; Herzallah and Lowe,), and adaptive critic based methods (Herzallah, 2007).

In the feedback linearization, backstepping and neural network based methods, only parameters or forward model uncertainty have been considered. The inverse controller has been assumed to be deterministic or dependent on the forward model. Stochastic adaptive control methods on the other hand have considered modeling the distribution of the inverse controller. However uncertainty in the stochastic adaptive control methods proposed in (Karny, 1996; Wang and Zhang, 2001; Wang, 2002), has been treated as a nuisance or perturbation therefore; did not affect the derivation of the optimal control law. In other words, uncertainty has been assumed to be input–independent and consequently they did not contribute to the derivation of the optimal control law. The stochastic adaptive control methods developed in (Herzallah and Lowe, 2007; Herzallah and Lowe,) on the other hand have considered input–dependent uncertainty and the methods are proven to significantly improve the performance of the controller.

Selected adaptive Critic (AC) methods, known as action–independent adaptive critic methods, have been shown to implement useful approximations of Dynamic Programming, a method for designing

optimal control policies in the context of nonlinear plants (Werbos, 1992). However in their conventional form, the action–independent adaptive critic methods do not take into consideration model uncertainty. In most recent development to these methods, a novel dual heuristic programming (DHP) adaptive–critic–based cautious controller is proposed (Herzallah, 2007). The proposed controller avoids the preidentification training phase of the forward model and inverse controller by taking into consideration model uncertainty when calculating the control law. Only forward model uncertainty has been considered in (Herzallah, 2007). The inverse controller is assumed to be accurate and no knowledge of uncertainty needed to be characterized. However, similar to the forward model, the parameters of the inverse controller of the nonlinear dynamical systems are usually optimized using nonlinear optimization methods. This inevitably leads to uncertain model of the inverse controller. Consequently, uncertainty of the inverse controller should be estimated and considered in the derivation of the optimal control law.

As a result, the dual heuristic programming (DHP) adaptive–critic–based cautious control method (Herzallah, 2007) is still in need of further development. This forms the main purpose of this paper, where functional uncertainty of both the forward model and the inverse controller is characterized and used in deriving the optimal control law. Hence the novelty of this work stems from considering functional uncertainty in the inverse controller as well as the forward model. Furthermore, a new method for estimating functional uncertainty of the models will be introduced in this work. In contrast to the method proposed in (Herzallah, 2007) this method allows for considering multiplicative noise on both the state and the control law. Also it guarantees the positivity of the covariance matrix of the errors. This well lead to a novel theoretical development for the stochastic adaptive control. Moreover, the Riccati solution for a quadratic linear infinite horizon control problem will also be derived and compared to the solution of the developed probabilistic (DHP) adaptive critic method. The method developed in this paper, enhances the performance of the system by utilizing more fully the probabilistic information provided by the forward model and the inverse controller. No pre–identification will be needed for neither the forward model, the critic or the inverse controller. All networks in the new developed framework will be adapted at each instant of time.

2 PRELIMINARIES

This preparatory section recalls basic elements of modeling conditional distributions of system outputs and inverse controller and the aim of fully probabilistic control.

2.1 Basic Elements

The behavior of a stochastic general class discrete time system with input $\mathbf{u}^{op}(k)$ and measurable state vector $\mathbf{x}(k)$ is described by a stochastic model of the following form

$$\mathbf{x}(k+1) = g[\mathbf{x}(k), \mathbf{u}^{op}(k)] + \tilde{\boldsymbol{\eta}}(k+1) \qquad (1)$$

where $\tilde{\boldsymbol{\eta}}(k+1)$ is random independent noise which has zero mean and covariance $\tilde{\mathbf{P}}$.

This can generally be expressed as:

$$\mathbf{x}(k+1) = f[\mathbf{x}(k), \mathbf{u}^{op}(k), \tilde{\boldsymbol{\eta}}(k+1)]. \qquad (2)$$

The randomized controller to be designed is described by the following stochastic model

$$\mathbf{u}^{op}(k) = \quad c[\mathbf{x}(k)] + \tilde{\mathbf{e}}(k) \qquad (3)$$

where $\tilde{\mathbf{e}}(k)$ represents random independent noise of zero mean and $\tilde{\mathbf{Q}}$ covariance matrix. Notice that only state dependent controllers are considered. However, assuming state dependent controller can be shown to represent no real restrictions (Mine and Osaki, 1970) provided that the state can be measured. The stochastic model of the controller can be reexpressed in the following general form:

$$\mathbf{u}^{op}(k) = \quad h[\mathbf{x}(k), \tilde{\mathbf{e}}(k)]. \qquad (4)$$

All probability density functions in this paper are assumed to be unknown and need to be estimated. The estimation method of these probability density functions will be discussed in Section 2.3, but first we introduce the aim of designing a probabilistic control.

2.2 Problem Formulation

In dynamic programming, the randomized controller of the above stochastic control problem is obtained by minimizing the expected value of the Bellman equation

$$J[(\mathbf{x}(k)] = \left\langle \left\{ U(\mathbf{x}(k), \mathbf{u}^{op}(k)) + \gamma J[\mathbf{x}(k+1)] \right\} \right\rangle \qquad (5)$$

where $< . >$ is the expected value, $J[\mathbf{x}(k)]$ is the cost to go from time k to the final time, $U(\mathbf{x}(k), \mathbf{u}^{op}(k))$ is the utility which is the cost from going from time k to time $k+1$, and $J[\mathbf{x}(k+1)]$ is assumed to be the minimum cost from going from time $k+1$ to the final

time. The term γ is a discount factor $(0 \leq \gamma \leq 1)$ which allows the designer to weight the relative importance of present versus future utilities. The objective is then to choose the control sequence $\mathbf{u}(k)$, $k = 1, 2, \ldots$, so that the function J in (5) is minimized.

The critic network in the DHP scheme, estimates a variable called $\lambda[\mathbf{x}(k)]$ as the derivatives of $J(\mathbf{x}(k))$ with respect to the vector $\mathbf{x}(k)$.

$$\begin{aligned}\lambda[\mathbf{x}(k)] &= \frac{\partial U[\mathbf{x}(k), \mathbf{u}^{op}(k)]}{\partial \mathbf{x}(k)} \\ &+ \frac{\partial U[\mathbf{x}(k), \mathbf{u}^{op}(k)]}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{x}(k)} \\ &+ < \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{x}(k)} > \\ &+ < \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{x}(k)} > \end{aligned} \tag{6}$$

where γ has been given the value of 1. Since $\langle \lambda[\mathbf{x}(k+1)] \rangle$, $U[\mathbf{x}(k), \mathbf{u}^{op}(k)]$ and the system model derivatives are known, then $\lambda[\mathbf{x}(k)]$ can be calculated. The optimality equation is defined as

$$\begin{aligned}\frac{\partial J[\mathbf{x}(k)]}{\partial \mathbf{u}^{op}(k)} &= \frac{\partial U[\mathbf{x}(k), \mathbf{u}^{op}(k)]}{\partial \mathbf{u}^{op}(k)} \\ &+ < \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{u}^{op}(k)} > \\ &= 0 \end{aligned} \tag{7}$$

The above two equations are usually used in dynamic programming to solve an infinite or finite horizon control policy.

If the nonlinear function $f[\mathbf{x}(k), \mathbf{u}^{op}(k), \tilde{\eta}(k+1)]$ was known or the system was noiseless, and given a deterministic function for the inverse controller, the optimal control law which achieves the above objective is shown to be derived using techniques of dynamic programming or DHP adaptive critic methods as an approximation methods to dynamic programming (Herzallah, 2007). Even if the function $f[\mathbf{x}(k), \mathbf{u}^{op}(k), \tilde{\eta}(k+1)]$ was unknown, researchers in the model based adaptive critic field would simply adapt a forecasting network which predicts the conditional mean of the state vector. This means that only deterministic models were considered in the conventional theory of the adaptive critic methods. Recently (Herzallah, 2007), it has been proved that the control law of the DHP adaptive critic methods which is derived based on the assumption of deterministic forward model is suboptimal. It has been shown in (Herzallah, 2007) that if the function of the controlled system was unknown then the problem should be formulated in an adaptive control scheme which is known to have functional uncertainty. Therefore, forward model uncertainty was quantified and used in their developed control algorithm. However, only forward model uncertainty was considered in (Herzallah, 2007). This forward model uncertainty was assumed to follow Gaussian distribution. The inverse controller on the other hand was assumed to be deterministic function.

In the current paper, the forward model and the inverse controller are described by probability density functions as shown in Equations (2) and (4). These probability density functions are not limited to Gaussian density, they can be of any shape. As mentioned in Section 2.1, the probability density functions of the forward model and the inverse controller are assumed to be unknown and need to be estimated in this paper. The objective of the current paper is then to develop an appropriate method for estimating the non Gaussian distributions of both the forward model and the inverse controller and then use these probabilistic information in the derivation of the optimal control law. This yields a novel DHP adaptive critic control algorithm which we refer to as probabilistic DHP adaptive critic method. The developed theory will be illustrated on linear infinite horizon quadratic control problem. The Riccati solution for this linear problem will also be derived.

2.3 Stochastic Model Estimation

In the neurocontrol field researchers usually adapt forecasting networks to predict the conditional mean of the system output or state vector, $\hat{\mathbf{x}}(k+1)$. In most control applications this is probably enough. However; with the growing complexity of control systems and because of the inherent uncertainty in modeling and estimation, researchers recently considered modeling the conditional distribution of the stochastic systems rather than relying on the single estimate of the neural networks.

To estimate the conditional distribution of the system output, a neural network model is optimized such that its output approximates the conditional expectation of the system output. Once the output of the neural network model has been optimized the stochastic model of the system is simply shown to be given by (Herzallah and Lowe, 2007),

$$\mathbf{x}(k+1) = \hat{\mathbf{x}}(k+1) + \eta(k+1), \tag{8}$$

where $\hat{\mathbf{x}}(k+1) = \hat{g}[\mathbf{x}(k), \mathbf{u}^{op}(k)]$, and $\eta(k+1)$ represents an input dependent random noise. The stochastic model in Equation (8) can in turn be reexpressed in the following general form:

$$\mathbf{x}(k+1) = \hat{f}[\mathbf{x}(k), \mathbf{u}^{op}(k), \eta(k+1)]. \tag{9}$$

Usually the noise $\eta(k+1)$ is assumed to follow Gaussian distribution of zero mean and covariance matrix **P**. In this work the assumption of Gaussian distribution is relaxed. In other words $\eta(k+1)$ is an input dependent random noise which could follow any non-Gaussian distribution of zero mean. This is a more realistic assumption, since a nonlinear mapping of random variable is non-Gaussian. This non-Gaussian distribution will be identified by evaluating the expectation and moments of the distribution. For example the second moment of the distribution is represented by its covariance matrix **P**. This covariance matrix represents the covariance of the error in predicting $\mathbf{x}(k+1)$.

The method proposed in (Herzallah and Lowe, 2007) estimates the conditional distribution of the system output by using another neural network model to provide a prediction for the input dependent covariance matrix $\mathbf{P} =< \eta(k+1)\eta^T(k+1) >$. In the current paper we propose a different method for estimating the conditional distribution of the system output which could be non-Gaussian as well. This novel proposed method is based on estimating the distribution of the input dependent error $\eta(k+1)$ and not the input dependent covariance matrix **P**. Since the covariance matrix **P** can be evaluated the distribution of the input dependent error is estimated by using a Gaussian Radial Basis Function neural network which has the important property of linear transformation.

$$\eta(\mathbf{x}(k),\mathbf{u}^{op}(k)) = \mathbf{w}\,\phi(\mathbf{x}(k),\mathbf{u}^{op}(k)). \qquad (10)$$

where $\mathbf{w}_i$ is a random vector which has zero mean and a covariance matrix $\Sigma_i =< \phi^{\dagger^T}\eta_i^T\eta_i\phi^{\dagger} >$, and i is the output index. Here the RBFNN is taken to be a probabilistic rather than deterministic model. To adapt this probabilistic neural network model the following conditions are assumed to hold for the neural network:

Assumption 1. The state and control is always confined within the network approximation region defined by subset $\mathcal{Z}$ whose boundaries are known. This approximation region is a design parameter and could be made arbitrarily large.

Assumption 2. The basis function centers and width parameters ensuring that condition 1 is satisfied are known a priori.

The second assumption is justified by the universal approximation property of neural networks with well known developed methods of choosing appropriate basis function centers and width parameters a priori (Sanner and Slotine, 1992).

Using the neural network as a probabilistic model for the input dependent error allows us to consider multiplicative noise on both the state and control. Besides, it ensures the positivity of the error covariance matrix **P**. Following the same procedure of the forward model, the stochastic model of the inverse controller is given by

$$\mathbf{u}^{op}(k) = \mathbf{u}(k) + \mathbf{e}(k). \qquad (11)$$

The distribution of the error in predicting the control law is also estimated using the same method of predicting the distribution of the error of the forward model.

To reemphasize, the method proposed in this section for estimating the conditional distributions of the models: ensures the positivity of the covariance matrix of the errors, it uses the neural network as a probabilistic models, and allows considering multiplicative noise on both the state and the control. However, the method proposed in (Herzallah and Lowe, 2007) does not guarantee the positivity of the covariance matrix, and it uses the neural network a deterministic model.

The theory developed in this section will be used in the next section for developing the theory behind the probabilistic DHP adaptive critic method proposed in this paper.

3 PROPOSED PROBABILISTIC ADAPTIVE CRITIC METHOD

In this section we propose a probabilistic type DHP adaptive critic controller which takes uncertainty of the forward model and the inverse controller into consideration when calculating the control law. The proposed controller can be obtained directly by optimally solving the adaptive critic problem which considers stochastic models rather than deterministic models. In the proposed probabilistic DHP adaptive critic method the control law is derived such as to minimize the expected value of the cost–to–go $J[\mathbf{x}(k)]$ given in (5) using $\gamma = 1$, but with the uncertainty of the models' estimates being taken into consideration. This is accomplished by treating the forward model and the inverse controller as random variables.

Following the procedure presented in Section 2.3 the conditional distributions of the forward model and the inverse controller are estimated. Using this in equation (5), Bellman's equation could be reexpressed as:

$$\begin{aligned} J[\mathbf{x}(k)] &=< U(\mathbf{x}(k),\mathbf{u}^{op}(k)) > + < J[\mathbf{x}(k+1)] > \\ &=< U(\mathbf{x}(k),\mathbf{u}^{op}(k)) > \\ &+ < J[\hat{f}(\mathbf{x}(k),\mathbf{u}^{op}(k),\eta(k+1))] > \end{aligned} \qquad (12)$$

Since the errors $\eta(k+1)$ and $\mathbf{e}(k)$ of the forward model and the inverse controller respectively are state dependent, the variable $\lambda[\mathbf{x}(k)]$ is shown to be given by the following theorem.

Theorem 1. The variable $\lambda[\mathbf{x}(k)]$ of the cost function of equation (12) subject to the stochastic models of equations (9) and (11), is given by

$$\begin{aligned}
\lambda[\mathbf{x}(k)] =< & \frac{\partial U[\mathbf{x}(k),\mathbf{u}^{op}(k)]}{\partial \mathbf{x}(k)} \\
& + \frac{\partial U[\mathbf{x}(k),\mathbf{u}^{op}(k)]}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{u}(k)} \frac{\partial \mathbf{u}(k)}{\partial \mathbf{x}(k)} \\
& + \frac{\partial U[\mathbf{x}(k),\mathbf{u}^{op}(k)]}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{e}(k)} \frac{\partial \mathbf{e}(k)}{\partial \mathbf{x}(k)} > \\
& + < \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{x}(k)} \\
& + \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{u}(k)} \frac{\partial \mathbf{u}(k)}{\partial \mathbf{x}(k)} \\
& + \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{e}(k)} \frac{\partial \mathbf{e}(k)}{\partial \mathbf{x}(k)} \\
& + \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \eta(k+1)} \frac{\partial \eta(k+1)}{\partial \mathbf{x}(k)} \\
& + \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \eta(k+1)} \frac{\partial \eta(k+1)}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{u}(k)} \frac{\partial \mathbf{u}(k)}{\partial \mathbf{x}(k)} \\
& + \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \eta(k+1)} \frac{\partial \eta(k+1)}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{e}(k)} \frac{\partial \mathbf{e}(k)}{\partial \mathbf{x}(k)} > \quad (13)
\end{aligned}$$

Proof. To prove the above theorem we simply derive the cost function of equation (12) with respect to the state $\mathbf{x}(k)$ at time k.

The error in predicting the state vector $\eta(k+1)$ is dependent on the control signal as well, so the optimality equation can be seen to be given by the following theorem.

Theorem 2. The optimality equation of the cost function of equation (12) subject to the stochastic models of equations (9) and (11), is given by

$$\begin{aligned}
\frac{\partial J[\mathbf{x}(k)]}{\partial \mathbf{u}(k)} =< & \frac{\partial U[\mathbf{x}(k),\mathbf{u}^{op}(k)]}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{u}(k)} \\
& + \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{u}(k)} \\
& + \lambda[\mathbf{x}(k+1)] \frac{\partial \mathbf{x}(k+1)}{\partial \eta(k+1)} \frac{\partial \eta(k+1)}{\partial \mathbf{u}^{op}(k)} \frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{u}(k)} >= 0
\end{aligned} \quad (14)$$

Proof. To prove the above theorem we simply derive the cost function of equation (12) with respect to the optimal control $\mathbf{u}(k)$ at time k.

The training process for the probabilistic type DHP adaptive critic proposed in this section is exactly the same as that for the conventional DHP adaptive critic. It consists of training the action network which outputs the optimal control policy $\mathbf{u}[\mathbf{x}(k)]$ and the critic network which approximates the derivative of the cost function $\lambda[\mathbf{x}(k)]$. As a first step both networks' parameters are initially randomized. Next, the difference between the target value of the critic, $\lambda^*[\mathbf{x}(k)]$ calculated from Equation (13) and the critic network output $\lambda[\mathbf{x}(k)]$ is used to correct the critic network until it converges. The output from the converged critic is used in (14) solving for the target $\mathbf{u}^{op}(k)$ which is then used to correct the action network. These two steps continue until a predetermined level of convergence is reached.

Because the proposed probabilistic DHP adaptive critic method takes model uncertainty into consideration, it is recommended to be implemented on–line. The forward model of the plant to be controlled the controller and the critic networks can all be adapted on–line.

4 LINEAR QUADRATIC MODEL

Stochastic linear quadratic models is one of the most widely used models in modern control engineering and finance. To understand and prove the validity of the proposed probabilistic DHP adaptive critic methods, the theory developed in the previous section is applied here to infinite horizon linear quadratic control problem. Before we evaluate the proposed methods themselves, the correct values of various functions in this problem will be calculated, so that we have something to check the proposed method against. Besides evaluating the correct values of various functions, we also derive the Riccati solution of this nonstandard stochastic control problem.

4.1 Dynamic Programming Solution for the Linear Quadratic Model

Suppose that the vector of observable, $\mathbf{x}(k)$ is the same as the state vector of the plant. Since we consider an infinite horizon problem, the objective is to minimize a measure of utility, $U(k)$, summed from the present time to the infinite future, which is defined by:

$$U(k) = \mathbf{x}^T \mathbf{O}\mathbf{x} + \mathbf{u}^{op^T} \mathbf{G}\mathbf{u}^{op}. \quad (15)$$

Suppose that the plant is described by the following stochastic model:

$$\mathbf{x}(k+1) = \mathbf{S}\mathbf{x}(k) + \mathbf{R}\mathbf{u}^{op}(k) + \eta(k+1), \quad (16)$$

where the error of the prediction, $\eta(k+1)$ is estimated as described in Section 2.3. This error is shown in Section 2.3 to be control signal and state dependent.

Since it should have the same structure and same inputs as the forward model of the plant, it is taken in this linear quadratic problem to be linear with two inputs, the state vector and the control signal:

$$\eta(k+1) = \mathbf{D}\mathbf{x}(k) + \mathbf{E}\mathbf{u}^{op}(k). \quad (17)$$

where **D** and **E** are matrices of random numbers which contain the parameters of the error model. Suppose that the action network is described by the following stochastic model:

$$\mathbf{u}^{op}(k) = \mathbf{u}(k) + \mathbf{e}(k), \quad (18)$$

where

$$\mathbf{u}(k) = \mathbf{A}\mathbf{x}(k) \quad (19)$$

and where **A** is the matrix of the controller parameters, and $\mathbf{e}(k)$ is the error in predicting the optimal control estimated as discussed in Section 2.3 and assumed to have the following form

$$\mathbf{e}(k) = \mathbf{Q}\mathbf{x}(k) \quad (20)$$

where **Q** is a matrix of random numbers that describes the mapping from the state space to the error in predicting the optimal control law. Using the control expression of Equation (19) in Equation (18) and substituting back in Equation (16) yields:

$$\mathbf{x}(k+1) = \tilde{\mathbf{S}}\mathbf{x}(k) + \mathbf{R}\mathbf{e}(k) + \eta(k+1), \quad (21)$$

where

$$\tilde{\mathbf{S}} = \mathbf{S} + \mathbf{R}\mathbf{A}. \quad (22)$$

Similarly the expression of the error in predicting the state vector as defined in Equation (17) can be rewritten in the following form

$$\eta(k+1) = \tilde{\mathbf{D}}\mathbf{x}(k), \quad (23)$$

where we have used Equations (18), (19), and (20) and where

$$\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{E}\mathbf{A} + \mathbf{E}\mathbf{Q}. \quad (24)$$

As a preliminary step to calculating the correct value of the cost function of Bellman's equation let us define **M** as the matrix that solves the following equation:

$$\begin{aligned}\mathbf{M} = \quad & \mathbf{O} + \mathbf{A}^T\mathbf{G}\mathbf{A} + <\mathbf{Q}^T\mathbf{G}\mathbf{Q}> + \tilde{\mathbf{S}}^T\mathbf{M}\tilde{\mathbf{S}} \\ + \quad & <\mathbf{Q}^T\mathbf{R}^T\mathbf{M}\mathbf{R}\mathbf{Q}> + <\tilde{\mathbf{D}}^T\mathbf{M}\tilde{\mathbf{D}}>. \end{aligned} \quad (25)$$

Following all the above definitions, the true value of the cost function *J* satisfying the Bellman equation is given in the following theorem:

Theorem 3. The true value of the cost function *J*, satisfying the Bellman equation (with $\gamma = 1$) subject to the system of equation (16) and uncertainty models of the forward model and the inverse controller defined in Equations (17) and (20) respectively and all other definitions previously mentioned is given by:

$$J(\mathbf{x}) = \mathbf{x}^T\mathbf{M}\mathbf{x}. \quad (26)$$

Proof. To prove the above theorem we simply substitute into Bellman's equation (12) and verify that it is satisfied. For the left hand side of the equation, we get:

$$J[\mathbf{x}(k)] = \mathbf{x}^T(k)\mathbf{M}\mathbf{x}(k). \quad (27)$$

For the right hand side, we get:

$$\begin{aligned} &< U(\mathbf{x}(k), \mathbf{u}^{op}(k)) + J[\mathbf{x}(k+1)] >= \\ &\quad < \mathbf{x}^T\mathbf{O}\mathbf{x} + (\mathbf{A}\mathbf{x}+\mathbf{e})^T\mathbf{G}(\mathbf{A}\mathbf{x}+\mathbf{e}) > \\ &\quad + < (\tilde{\mathbf{S}}\mathbf{x} + \mathbf{R}\mathbf{e} + \eta)^T\mathbf{M}(\tilde{\mathbf{S}}\mathbf{x} + \mathbf{R}\mathbf{e} + \eta) > \\ &= \mathbf{x}^T\mathbf{O}\mathbf{x} + \mathbf{x}^T\mathbf{A}^T\mathbf{G}\mathbf{A}\mathbf{x} + \mathbf{x}^T <\mathbf{Q}^T\mathbf{G}\mathbf{Q}> \mathbf{x} + \mathbf{x}^T\tilde{\mathbf{S}}^T\mathbf{M}\tilde{\mathbf{S}}\mathbf{x} \\ &\quad + \mathbf{x}^T <\mathbf{Q}^T\mathbf{R}^T\mathbf{M}\mathbf{R}\mathbf{Q}> \mathbf{x} + \mathbf{x}^T <\tilde{\mathbf{D}}^T\mathbf{M}\tilde{\mathbf{D}}> \mathbf{x}, \end{aligned} \quad (28)$$

where we used Equations (20) and (23) and where we made use of the fact that η, and **e** are uncorrelated random variables of zero mean. Making use of Equation (25) in Equation (28), yields

$$J(\mathbf{x}) = \mathbf{x}^T\mathbf{M}\mathbf{x}. \quad (29)$$

Comparing Equations (26) and (29) we can see that Bellman's equation is satisfied. Howards has proven (Howard, 1960) that the optimal control law, based on the policy iteration method, can be derived by alternately calculating the cost function *J* for the current control law, modify the control law so as to minimize the cost function *J*, recalculate *J* for the new control law, and so on.

4.2 Proposed Probabilistic DHP Adaptive Critic in the Linear Quadratic Model

The objective of this section is to calculate the targets for the output of the critic network $\lambda^*[\mathbf{x}(k)]$ as they would be generated by the proposed probabilistic DHP adaptive critic, and then check them against the correct values. In other words we need to check that if the critic was initially correct it will stay correct after one step of adaptation.

From (29), the correct value of $J(\mathbf{x})$ is $\mathbf{x}^T\mathbf{M}\mathbf{x}$, and consequently the correct value of $\lambda(\mathbf{x})$ is simply the gradient of $J(\mathbf{x})$, i.e $\lambda(\mathbf{x}) = 2\mathbf{M}\mathbf{x}(k)$. Hence $\lambda(k+1)$ is given by,

$$\lambda(k+1) = 2\mathbf{M}\mathbf{x}(k+1).$$

Next we carry out the calculations implied by equation (13) but with the expectation of the derivatives being evaluated at the end.

To calculate the first term on the right hand side of (13), we simply calculate the gradient of $U(\mathbf{x}(k),\mathbf{u}^{op}(k))$ with respect to $\mathbf{x}(k)$:

$$\frac{\partial U[\mathbf{x}(k),\mathbf{u}^{op}(k)]}{\partial \mathbf{x}(k)} = 2\mathbf{O}\mathbf{x}(k).$$

For the second term the value of the partial derivatives of $\mathbf{u}(k)$, $\mathbf{u}^{op}(k)$ and $U(\mathbf{x}(k),\mathbf{u}^{op}(k))$ with respect to $\mathbf{x}(k)$, $\mathbf{u}(k)$and $\mathbf{u}^{op}(k)$ respectively need to be evaluated:

$$\frac{\partial U[\mathbf{x}(k),\mathbf{u}^{op}(k)]}{\partial \mathbf{u}^{op}(k)}\frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{u}(k)}\frac{\partial \mathbf{u}(k)}{\partial \mathbf{x}(k)} = 2\mathbf{G}(\mathbf{u}+\mathbf{e})\mathbf{A}.$$

The third term can be evaluated by calculating the partial derivatives of $\mathbf{e}(k)$, $\mathbf{u}^{op}(k)$ and $U(\mathbf{x}(k),\mathbf{u}^{op}(k))$ with respect to $\mathbf{x}(k)$, $\mathbf{e}(k)$and $\mathbf{u}^{op}(k)$ respectively:

$$\frac{\partial U[\mathbf{x}(k),\mathbf{u}^{op}(k)]}{\partial \mathbf{u}^{op}(k)}\frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{e}(k)}\frac{\partial \mathbf{e}(k)}{\partial \mathbf{x}(k)} = 2\mathbf{G}(\mathbf{u}+\mathbf{e})\mathbf{Q}.$$

The fourth term requires propagating $\lambda(k+1)$ through the model of equation (16) back to $\mathbf{x}(k)$, which yields

$$\lambda[\mathbf{x}(k+1)]\frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{x}(k)} = 2\mathbf{M}\mathbf{x}(k+1)\mathbf{S}.$$

The fifth term requires propagating $\lambda(k+1)$ through the model of the plant, $\mathbf{x}(k+1)$, back to $\mathbf{u}^{op}(k)$ and then through the action network, which yields

$$\lambda[\mathbf{x}(k+1)]\frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{u}^{op}(k)}\frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{u}(k)}\frac{\partial \mathbf{u}(k)}{\partial \mathbf{x}(k)} = 2\mathbf{M}\mathbf{x}(k+1)\mathbf{R}\mathbf{A}.$$

The sixth term can be calculated by propagating $\lambda(k+1)$ through the model of the plant, $\mathbf{x}(k+1)$, back to $\mathbf{u}^{op}(k)$ and then through the error network of the controller, which yields

$$\lambda[\mathbf{x}(k+1)]\frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{u}^{op}(k)}\frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{e}(k)}\frac{\partial \mathbf{e}(k)}{\partial \mathbf{x}(k)} = 2\mathbf{M}\mathbf{x}(k+1)\mathbf{R}\mathbf{Q}.$$

The seventh term can also be calculated by propagating $\lambda(k+1)$ through the model of the plant, $\mathbf{x}(k+1)$, back to the error network of the forward model, which yields

$$\lambda[\mathbf{x}(k+1)]\frac{\partial \mathbf{x}(k+1)}{\partial \eta(k+1)}\frac{\partial \eta(k+1)}{\partial \mathbf{x}(k)} = 2\mathbf{M}\mathbf{x}(k+1)\mathbf{D}.$$

The eighth term is calculated by propagating $\lambda(k+1)$ through the model of the plant, $\mathbf{x}(k+1)$, back to the error network of the forward model and then $\mathbf{u}^{op}(k)$ and then through the action network. This yields

$$\lambda[\mathbf{x}(k+1)]\frac{\partial \mathbf{x}(k+1)}{\partial \eta(k+1)}\frac{\partial \eta(k+1)}{\partial \mathbf{u}^{op}(k)}\frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{u}(k)}\frac{\partial \mathbf{u}(k)}{\partial \mathbf{x}(k)} = 2\mathbf{M}\mathbf{x}(k+1)\mathbf{E}\mathbf{A}.$$

Finally the last term is calculated as follows

$$\lambda[\mathbf{x}(k+1)]\frac{\partial \mathbf{x}(k+1)}{\partial \eta(k+1)}\frac{\partial \eta(k+1)}{\partial \mathbf{u}^{op}(k)}\frac{\partial \mathbf{u}^{op}(k)}{\partial \mathbf{e}(k)}\frac{\partial \mathbf{e}(k)}{\partial \mathbf{x}(k)} = 2\mathbf{M}\mathbf{x}(k+1)\mathbf{E}\mathbf{Q}.$$

Adding all terms together and taking the expectation, yields

$$\begin{aligned}\lambda^*(k) =& < 2\mathbf{O}\mathbf{x}(k) + 2\mathbf{A}^T\mathbf{G}\{\mathbf{A}\mathbf{x}(k)+\mathbf{Q}\mathbf{x}(k)\} \\ &+ 2\mathbf{Q}^T\mathbf{G}\{\mathbf{A}\mathbf{x}(k)+\mathbf{Q}\mathbf{x}(k)\} + 2\mathbf{S}^T\mathbf{M}\{\tilde{\mathbf{S}}\mathbf{x}(k)+\mathbf{R}\mathbf{Q}\mathbf{x}(k) \\ &+ \tilde{\mathbf{D}}\mathbf{x}(k)\} + 2\mathbf{A}^T\mathbf{R}^T\mathbf{M}\{\tilde{\mathbf{S}}\mathbf{x}(k)+\mathbf{R}\mathbf{Q}\mathbf{x}(k)+\tilde{\mathbf{D}}\mathbf{x}(k)\} \\ &+ 2\mathbf{Q}^T\mathbf{R}^T\mathbf{M}\{\tilde{\mathbf{S}}\mathbf{x}(k)+\mathbf{R}\mathbf{Q}\mathbf{x}(k)+\tilde{\mathbf{D}}\mathbf{x}(k)\} + 2\mathbf{D}^T\mathbf{M}\{\tilde{\mathbf{S}}\mathbf{x}(k) \\ &+ \mathbf{R}\mathbf{Q}\mathbf{x}(k)+\tilde{\mathbf{D}}\mathbf{x}(k)\} + 2\mathbf{A}^T\mathbf{E}^T\mathbf{M}\{\tilde{\mathbf{S}}\mathbf{x}(k)+\mathbf{R}\mathbf{Q}\mathbf{x}(k) \\ &+ \tilde{\mathbf{D}}\mathbf{x}(k)\} + 2\mathbf{Q}^T\mathbf{E}^T\mathbf{M}\{\tilde{\mathbf{S}}\mathbf{x}(k)+\mathbf{R}\mathbf{Q}\mathbf{x}(k)+\tilde{\mathbf{D}}\mathbf{x}(k)\} >,\end{aligned} \tag{30}$$

where we used equations (21), (19), (20) and (23). Evaluating the expectation of Equation (30) yields,

$$\begin{aligned}\lambda^*(k) &= 2\mathbf{O}\mathbf{x}(k) + 2\mathbf{A}^T\mathbf{G}\mathbf{A}\mathbf{x}(k) \\ &+ 2 < \mathbf{Q}^T\mathbf{G}\mathbf{Q} > \mathbf{x}(k) + 2\tilde{\mathbf{S}}^T\mathbf{M}\tilde{\mathbf{S}}\mathbf{x}(k) \\ &+ 2 < \mathbf{Q}^T\mathbf{R}^T\mathbf{M}\mathbf{R}\mathbf{Q} > \mathbf{x}(k) + 2 < \tilde{\mathbf{D}}^T\mathbf{M}\tilde{\mathbf{D}} > \mathbf{x}(k),\end{aligned} \tag{31}$$

where we made use of the fact that the expected value of the random variables $\mathbf{Q}$, $\mathbf{E}$, and $\mathbf{D}$ is zero, that η and $\mathbf{e}$ are uncorrelated and finally that $\mathbf{Q}$ and $\mathbf{E}$ are uncorrelated random variables. Using equation (25) in (31) yields,

$$\lambda^*(k) = 2\mathbf{M}\mathbf{x}(k). \tag{32}$$

From (32) it can be clearly seen that the target vector of the proposed probabilistic critic network is equal to the correct value. This validates the theoretical development of the probabilistic DHP adaptive critic method proposed in this paper.

5 CONCLUSIONS

The nonstandard formulation of the stochastic control design presented in this paper leads to a different form of optimal controller that depends on the solution of stochastic functional equations. It provides the complete solution for designing a stochastic controller for complex control systems accompanied by high levels of inherent uncertainty in modeling and estimation. All probability density functions needed in the proposed methods are assumed to be unknown. To estimate these probability density functions we propose using probabilistic neural network models to estimate errors in predicting conditional expectations of the

functions. This proposed method always guarantees the positivity of the covariance of the errors and allows for considering multiplicative noise on both the state and control of the system.

The proposed probabilistic DHP critic method is suitable for deterministic and stochastic control problems characterized by functional uncertainty. Unlike current established control methods, it takes uncertainty of the forward model and inverse controller into consideration when deriving the optimal control law.

Theoretical development in this paper is demonstrated through linear quadratic control problem. There, the correct value of the cost function which satisfies the Bellman equation is evaluated and shown to be equal to its corresponding value produced by the proposed probabilistic critic network.

REFERENCES

Botto, M. A., Wams, B., van den Boom, and da Costa, J. M. G. S. (2000). Robust stability of feedback linearised systems modelled with neural networks: Dealing with uncertainty. *Engineering Applications of Artificial Intelligence*, 13(6):659–670.

Fabri, S. and Kadirkamanathan, V. (1998). Dual adaptive control of nonlinear stochastic systems using neural networks. *Automatica*, 34(2):245–253.

Ge, S. S., Hang, C. C., Lee, T. H., and Zhang, T. (2001). *Stable Adaptive Neural Network Control*. Kluwer, Norwell, MA.

Ge, S. S. and Wang, C. (2004). Adaptive neural control of uncertain mimo nonlinear systems. *IEEE Transactions on Neural Networks*, 15(3):674–692.

Herzallah, R. (2007). Adaptive critic methods for stochastic systems with input-dependent noise. *Automatica*. Accepted to appear.

Herzallah, R. and Lowe, D. A Bayesian perspective on stochastic neuro control. *IEEE Transactions on Neural Networks*. re-submited 2006.

Herzallah, R. and Lowe, D. (2007). Distribution modeling of nonlinear inverse controllers under a Bayesian framework. *IEEE Transactions on Neural Networks*, 18:107–114.

Hovakimyan, N., Nardi, F., and Calise, A. J. (2001). A novel observer based adaptive output feedback approach for control of uncertain systems. In *Proceedings of the American Control Conference*, volume 3, pages 2444–2449, Arlington, VA, USA.

Howard, R. A. (1960). *Dynamic Programming and Markov Processes*. The Massachusetts Institute of Technology and John Wiley and Sons, Inc., New York. London.

Karny, M. (1996). Towards fully probabilistic control design. *Automatica*, 32(12):1719–1722.

Lewis, F. L., Yesildirek, A., and Liu, K. (2000). Robust backstepping control of induction motors using neural netwoks. *IEEE Transactions on Neural Networks*, 11:1178–1187.

Mine, H. and Osaki, S., editors (1970). *Markovian Decision Processes*. Elsevier, New York, N.Y.

Murray-Smith, R. and Sbarbaro, D. (2002). Nonlinear adaptive control using non-parametric gaussian process prior models. In *15th IFAC Triennial World Congress*, Barcelona.

Sanner, R. M. and Slotine, J. J. E. (1992). Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks*, 3(6).

Sastry, S. S. and Isidori, A. (1989). Adaptive control of linearizable systems. *IEEE Transactions on Automatic Control*, 34(11):1123–1131.

Wang, D. and Huang, J. (2005). Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form. *IEEE Transactions on Neural Networks*, 16(1):195–202.

Wang, H. (2002). Minimum entropy control of non-gaussian dynamic stochastic systems. *IEEE Transactions on Automatic Control*, 47(2):398–403.

Wang, H. and Zhang, J. (2001). Bounded stochastic distribution control for pseudo armax stochastic systems. *IEEE Transactions on Automatic Control*, 46(3):486–490.

Werbos, P. J. (1992). Approximate dynamic programming for real-time control and neural modeling. In White, D. A. and Sofge, D. A., editors, *Handbook of Intillegent Control*, chapter 13, pages 493–526. Multiscience Press, Inc, New York, N.Y.

Zhang, Y., Peng, P. Y., and Jiang, Z. P. (2000). Stable neural controller design for unknown nonlinear systems using backstepping. *IEEE Transactions on Neural Networks*, 11:1347–1359.

DESIGN OF NEURONAL NETWORK TO CONTROL SPIRULINA AQUACULTURE

Ernesto Ponce, Claudio Ponce and Bernardo Barraza
Mechanical Engineering Department, Electronic Engineering Department, Tarapacá University
18 de Septiembre 2222, Casilla 6-D, Arica, Chile
eponce@uta.cl

Keywords: Neural Network, Spirulina, Control, Aquaculture, Signal Processing System.

Abstract: A neural network that was designed to control a Spirulina aquaculture process in a pilot plant in the north of Chile, is presented in this work. Spirulina is a super food, but is a delicate alga and its culture may be suddenly lost by rapid changes in the weather that can affect its temperature, salinity or pH. The neural network control system presented is complex and non linear, and has several variables. The previous automatic control system for the plant proved unable to cope with large climatic variations. The advantage of this new method is the improvement in efficiency of the process, and a reliable control system that is able to adapt to climatic changes. The future application of this work is related to the industrial production of food and fuel from micro algae culture, for the growing world population.

1 INTRODUCTION

Spirulina Platensis is a single cell micro alga that belongs to the cianobacteria group. It has a blue-green colour and spiral shape and reproduces by intracellular rupture. Its length varies between 20 and 50 microns. Spirulina can survive at temperatures between 13° and 33°C and at a pH between 8.5 and 10.5. This micro alga is 60% all-vegetable protein, rich in beta carotene, iron, vitamin B-12 and the rare essential fatty acid, GLA. It is considered the super food of the future (Henrikson, 1994), (Spirulina.com, 2004), (Jourdan, 2002).

In 2002 a sudden weather change destroyed the Spirulina culture in a pilot plant (raceway system) in Azapa Valley, Arica, Chile, by causing variations in density, salinity, pH and temperature. The classic automatic control in place could not manage the changes and the culture was lost. It was therefore necessary to design an intelligent control system able to adapt to adverse weather changes.

In Chilean modern water farms a complete system that controls Spirulina alga cultivation does not exist. The control system designed and presented here was based on a neuronal network. The variables are pH, temperature, salinity (electrical conductivity, directly related to the density of the solution) and population density. The independent variables of the system were entered. The exit of the network was forced with the evolution state of the culture in time. This procedure was repeated several times having changed the input variables (Hagan et al., 2002). Once the control system was in place, the neural network continued to learn and evolve.

2 THEORY

2.1 Neuronal Networks

Neural networks, inspired by biological nervous systems, are composed of simple elements that operate in parallel (Figure 1). As in nature, the network function is determined by the connections between elements. A neural network can be ”trained” to perform a particular function by adjusting the values of the connections between elements (Hunt et al., 2002). Thus neural networks can be trained to solve problems that are difficult for conventional computers or human beings. Moreover, they can incorporate the best techniques for pattern recognition and tendency analysis.

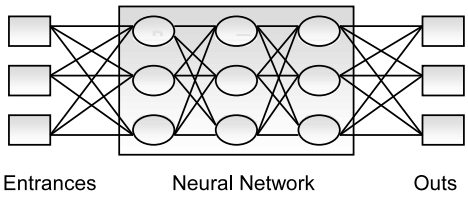


Figure 1: Basic model of a neuronal network.

For applications that demand networks with fewer than 100 neurons and little training, the software implementations are sufficient. When the problem requires over 100 neurons and 10000 synapses, is necessary to use hardware.

3 DEVELOPMENT

- Design of neuronal network
- Method
- Network training
- Operation

3.1 Design of the Neuronal Network

The variables of the system are:

- pH: This variable is very important because the alga survives within a range of pH between 8.5 and 10.5. pH can be measured with a pH meter. If the pH is low, then a valve connected with a little tank, is open to give a bicarbonate sodium solution. If it is high, another valve gives CO2 to a dome on the surface aquaculture.
- Temperature: Spirulina Platensis can tolerate temperatures between 13° to 33°C. If the temperature rises over the top limit a fan is connected and the windows open. If the temperature is low, the windows are closed (the aquaculture is in a greenhouse).
- Flow density: The density of the flow is directly related to salinity and conductivity. Thus, it can be controlled by the regulation of the doses of clean water, marine salt, nitrate and sodium bicarbonate supplied to the culture. The maximum and minimal density values are 1.05 and 1.20 g/cm^3
- Population density: When the Spirulina population reaches a maximum density, this may result in sudden death. The optimum time for collection (harvest) is therefore before maximum density is reached, at a density of 900 mg/l. Density was measured with a laser device (Ponce, 2001). At 800 mg/l a pump is connected and the harvest begins.

In order to train the neural network, the independent variables were measured and the dependent variable was forced to a desired value. During the training, time was used as an additional variable to distribute the learning into discrete cycles (Hagan and Demuth, 1999).

Independent variables:

- pH
- Temperature
- Flow density (salinity and electrical conductivity)

Dependent variable:

- Population density

After training, the system controlled 4 variables:

- pH
- Temperature
- Flow density (salinity and electrical conductivity).
- Population density

3.2 Method

It has 4 inputs (measured variables), and 4 outputs - 3 of which variables that must be controlled; the remaining output is for training the neural network. The best way to train a neuronal network is by means of variable forced learning (Gutiérrez et al., 2004). Training consists of producing successive cultures, each one under different constant conditions, maximum and minimum values. Training is completed when the learning margin error approaches zero. For a different condition the system will compute between the 2 extreme data. Offside these maximum and minimal data the control does not work.

The extreme conditions are: pH min= 8.5 ; pH max = 10.5
Temperature: min = 13°C; max = 33°C
Population density mg/l: min = 100; max = 800
Flow density g/cm^3 : min = 1.05 ; max = 1.20

The best control scheme for this problem is a NARMA2 network (Norvig and Russell, 2003), a neuronal controller that transforms nonlinear system dynamics into linear dynamics by canceling the nonlinearities during training. This optimizes the performance of the hardware. A linear system enables faster training and control in real time for microprocessors.

The control input is computed to force the plant output to follow a reference signal. The neural network plant model is trained with static back propagation and is reasonably fast. It requires minimal online computation (Rio and Molina, 2002).

The hardware has a 4*x*4 parallel pic processor array, with 4 inputs and 4 outputs. This array emulates 20 layers with 4 files, (80 neurons). This hardware is an experimental prototype.

In a NARMA2 structure, each neuron is simulated as:

$$u(k+1) = \frac{yr(k+d) - f[y(k),\ldots,y(k-n+1),u(k),\ldots,u(k-n+1)]}{g[y(k),\ldots,y(k-n+1),u(k),\ldots,u(k-n+1)]} \quad (1)$$

Where $y(n)$ is the system input, and $u(n)$ is the system output. The system function is a linear combination of $f(y(n-1),u(m-1))$ and $g(y(n-1),u(m-1))$. Each neuron can be simulated as shown in Figure 2:

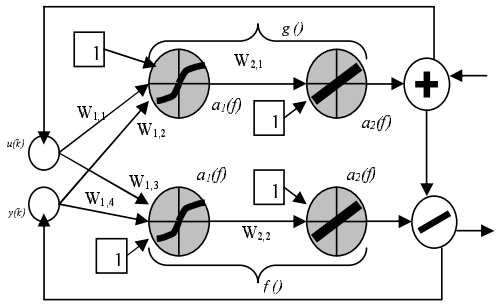


Figure 2: Neurons of NARMA2 model.

3.3 Network Training

The neural network was connected to the culture as shown in Figure 3. The training error is e_c. The inputs are y_r, and the exit variables are u.

Each line is equal to 7 lines. The controller with 80 neurons emulates the system as a linear combination of g and f (generated in the train).

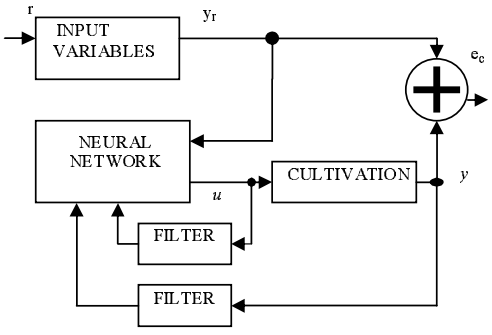


Figure 3: Connection of the Neural Network to the cultivation.

The training of the neural network is online, supervised and forced. It takes at least 6 weeks (6 cultivations in different conditions).

Employing 2x3 maximum and minimal values (2 by each parameter) a simulation with 3 set of variables was made in MATLAB.

Based on the variation of each input, the neural network built a linear model system. This enables each output to be related to the variation of all inputs. The training simulation is carried out with a forced reference. The dependent variable automatically adjusts the inputs and tries to follow the reference (Figure 4).

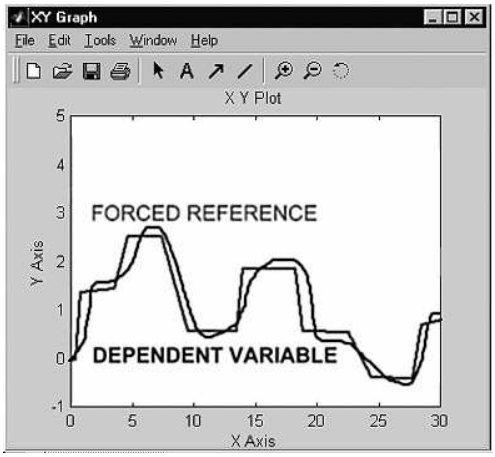


Figure 4: Simulation in MATLAB during the training stage.

Figure 5 shows the output 1 ($u1$) due to input 1 ($y1$), when the system is in operation.

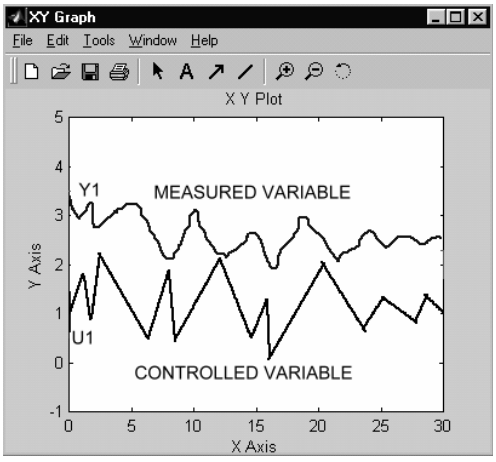


Figure 5: Simulation in MATLAB using 1 variable after training with 3 variable.

3.3.1 Operation

After training, the neural network has 3 inputs and 3 outputs. During the operation, the neural network learns new rules, it works as an expert controller.

4 CONCLUSIONS

When used to control a Spirulina alga aquaculture process, a classic automatic control system was unable to respond adequately to sudden climatic variations.

This neural network design presented here provides a good control alternative, one that is able to adapt to climatic changes and seasons, between maximum and minimal operation parameters (previously known) .

The neural network bestows a reliable, robust control system which can give a correct response to unknown situations. If part of the hardware is damaged, the information can be saved.

Furthermore, implementation is not expensive. The cost of building the prototype control system was approximately US $ 500 for a pilot plant race way system (Area= $3m^2$, volume = $0.5m^3$).

In the future it will be necessary to obtain more and more foods and fuels. The aquaculture is not an expensive solution and neither is the intelligent control proposed. A similar control system could be applied, with some changes, to another aquaculture focused to obtain fuel from glycerol (Dunaliella Salina) or petroleum (Botryococcus).

ACKNOWLEDGEMENTS

With acknowledgements to the Spirulina Control Process Project: DIPOG 8741-00, Tarapaca University (Chile) and Associate Professor Claus Brieba of the Arthur Prat University, without all of whom, this work would have been impossible.

REFERENCES

Gutiérrez, J., Cano, R., and Cofino, A. (2004). *Redes Probabilísticas y Neuronales*. Monografías del Instituto Nacional de Meteorología. Ministerio de Medio Ambiente, Madrid, Spain.

Hagan, M. and Demuth, H. (1999). Neural networks for control. pages 1642–1656, San Diego, CA.

Hagan, M., Jesus, O., and Schultz, R. (2002). *Recurrent Neural Networks: Design and Applications*, chapter 12, pages 311–340. CRC Press.

Henrikson, R. (1994). *Microalga Spirulina - Superalimento del futuro*. Urano S.A., Barcelona, Spain.

Hunt, K., Sbarbaro, D., Zbibowski, R., and Gawthrop, P. (2002). Neural networks for control system - a survey. volume 28, pages 1083–1112.

Jourdan, J. (2002). *Cultivez Votre Spiruline. Manuel de Culture Artisanale de la Spiruline*. Le Castanet, Mialet, 30140 Anduz, France.

Norvig, P. and Russell, S. (2003). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 2nd edition.

Ponce, C. (2001). Implementación de sistemas de medición de población de microalgas. Master's thesis, Tarapaca University.

Rio, B. D. and Molina, A. (2002). *Redes Neuronales Y Sistemas Borrosos*. RA-MA, 2nd edition.

Spirulina.com (2004). Spirulina. green superfood for life. http://www.spirulina.com/SPBSpirulina.html.

AN ONLINE BANDWIDTH SCHEDULING ALGORITHM FOR DISTRIBUTED CONTROL SYSTEMS WITH MULTIRATE CONTROL LOOPS

Saroja Kanchi
Department of Computer Science, Kettering University, Flint, MI 48504, U.S.A.
skanchi@kettering.edu

Juan Pimentel
Department of Electrical and Computer Engineering, Kettering University, Flint, MI 48504, U.S.A.
jpimente@kettering.edu

Keywords: Control systems, control loops, scheduling, distributed systems.

Abstract: In this paper, we present an online scheduling algorithm for communication in a distributed control system. The packet size of the communication varies for each execution of the loop within certain bounds. We consider systems with closed loops that restart immediately after the completion of an execution. Our algorithm is based on priority of the loop and size of the communication packet. We demonstrate through simulation that our algorithm generates a feasible schedule that minimizes average control delay over all the loops. Our simulations demonstrate that this online schedule reduces average delay significantly compared to a-priori schedules for distributed control systems. We demonstrate that bandwidth utilization is more efficient in case of online scheduling.

1 INTRODUCTION

Distributed Control Systems are becoming popular since they offer flexibility, modularity, speed and efficiency in designing a control system. Distributed Control System is made up of large number of components each of which performs a dedicated task and also communicates with other components. Examples of distributed control systems include office and home automation, aircraft and spacecraft systems and automotive component systems. The tasks within the Distributed Control Systems have strict timing requirements on start times and completion times.

The performance of a distributed control system depends not only on the performance of the individual components but also on the interaction between components. Timing of the task completion and communication pose significant challenges in designing a Distributed Control Systems. Added to this complexity is the issue of fault-tolerance and reliability. Designing a-priori or static algorithms for distributed control systems have been considered for a long time. These algorithms do not take into consideration the changing nature of interaction within the components of the system or application specific requirements of the control system. We propose an online bandwidth scheduling algorithm for communication in a distributed control system. This algorithm takes into consideration the priority of each task within the system and also size of each communication task. The size of communication depends on the results of the computation at each node of the system. Thus, we are proposing a dynamic algorithm that addresses the changing communication needs of the system.

The particular type of Distributed Control Systems we consider in this paper are those that consist of control loops. In (Yepez et al., 2003) an algorithm called the largest error first allocates bandwidth on the basis of the loop that contains the largest error. But the performance is not documented in the paper. In (Velasco et al., 2004) provide a dynamic bandwidth allocation algorithm based on adding a state variable to the control system model for small control systems. Their algorithm uses the performance of the control system as a feed back mechanism for scheduling the network. Our algorithm assumes an schedule that is handled by the network controller.

There were several static real-time scheduling algorithms designed for various real time tasks (Santos et al., 1997; Altenbernd and Hansson, 2004;

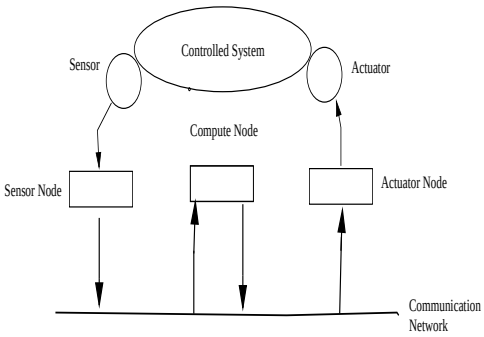


Figure 1: Model of Control Loop.

Blazewicz and Ecker, 1994; Ramamritham, 1990). Peng and Shin (Peng and Shin, 1989) provide a static allocation of processors for periodic tasks that contain both computation and communication. The communication imposes precedence constraints among the tasks. The allocation does not allocate communication link and is static allocation. Xu (Xu, 1993) provides an algorithm for multiprocessor scheduling for tasks with deadlines, release times, precedence and exclusion constraints. Again this schedule provides static schedule for processor assignment.

2 THE MODEL

We assume there are n control loops in the Distributed Control System. Each loop contains a sensor node that senses the values of certain variables. The sensor then transmits the results of the sensed variables over the network. Once the results of the sensor is available the compute node then starts to compute its algorithms. The output of the compute node is then communicated over the network to the actuator node. Since the loop is closed and the loops are multirate, we assume that after the actuator triggers the necessary components, the sensor nodes starts again at the beginning of the loop. This system is illustrated in Figure 1.

The three tasks of a control loop can be done in the order of sensing, computation and actuation and then the control loop begins at sensing again. Note that there are three dedicated nodes at each loop. The sensor node that converts the sensed signal into variable values, the compute node that is dedicated to computing values, and actuator node that produces the desired signal on the basis of computed values. Theses nodes can work in parallel if desired.

Each control loops is represented as follows: S_k, C_k and A_k represent the sensor, compute and actuator nodes for the k^{th} control loop. There are several types of possible delays in the execution of a control loop. The delay sensor node, compute node and actuator node for the k^{th} loop are denoted by s_k, c_k and a_k respectively. It is assumed that the set of control loops of the system share the same communication channel. The communication delay are of two types- The communication delay that is caused while sending the sensed variables to the compute node over the communication network is denoted by st_k and the communication delay caused due to transmission of the computed values is denoted by ct_k. Therefore, the total control delay D_k for the k^{th} loop is denoted by

$$D_k = s_k + st_k + c_k + ct_k + a_k$$

We assume that each loop in the system has a priority given by p_k. The control loops are closed and they restart at a different rates. They restart as soon as an execution of the loop is completed. The algorithm we develop for scheduling the transmission of the results of sensor and compute nodes has the goal of minimizing the average loop delay over the given set of loops in the system. We assume that higher priority loops (critical loops) execute more times than a non-critical loop over a period of time.

Unlike earlier models of the problem where communication packet size is assumed to be same, we vary the packet size between given limits for each execution For example, the packet size produced by the sensor node S_k varies between $f(s_k)$ and $g(s_k)$, where f and g are linear functions. Our scheduling therefore can take advantage of the actual packet size of the communication rather than a a-priori scheduling of the communication independent of the packet size. This avoids over allocation of the communication channel for the control loop.

2.1 Algorithm for Bandwidth Allocation for Control Loops with Priority

For each control loop k, we are given the times s_k, c_k and a_k. These values do not change over multiple executions of the loops. However, the values st_k, ct_k change depending the results produced by sensor and communication nodes. The transmission time needed on the communication channel is proportional to st_k and ct_k respectively. If a loop is ready to transmit sensor variables, we set the *next transmission length* to st_k. If the loop ready to transmit computed values we set the *next transmission length* to ct_k. At any point of time, a set of loops which we call *ready loops* will be ready to transmit either the results of the sensor node or the results of the compute node. We use the following algorithm to schedule the communication channel

At a time t,

```
For each time t, Repeat until all loops are scheduled
   1) Find the loop(s) that has the highest priority
   2) Among the loops with highest priority find the loop
      that has the lowest next transmission length
   3) Schedule the transmission of this loop
```

It is important to note that all ready loops that are ready a a particular point of time are scheduled together. That is, the same high priority loop cannot repeatedly use the channel thus reducing the average waiting time over all loops. This technique avoids the well known problem of starvation of low priority loops.

It is important to note that the selection of one among several loops of same priority is based on transmission length and the one with the shortest length is chosen. This is in tune in Shortest Job First scheduling which is the optimal algorithm for scheduling jobs on a single processor.

2.2 Algorithm for Bandwidth Allocation for Control Loops with Equal Priority

If the control loops have no assigned priority then we cannot use the transmission alone as a measure since deadline for each loop may not be met. Therefore, we use the following algorithm for control systems that do no have priority of loops.

At a time t,

```
For each time t, repeat until all loops are scheduled
    Find the loop(s) that has the lowest next burst time.
     The next-burst-time for control loop k is defined as
    next-burst-time = st(k) + c(k)
     if the communication desired is transmission of sensor values
    next-burst-time = ct(k) +a_k+s(k)  if the
      communication desired if transmission of computed values
End
```

In this case, the delay of control loops is roughly speaking, inversely proportional to the length of the loop and therefore, the average control delay is minimized for any set of loops.

3 SIMULATION

The simulation was done using a set of control loops with the following parameters for each control loop:

sensorLow - low value of the sensor node processing time
sensorHigh - high value of the sensor node processing time
computeLow - low value of the compute node processing time
computeHigh - high value of the compute node processing time
actuationLow - low value of the actuation node processing time
actuationHigh - high value of the actuation node processing time

The s_k, c_k and a_k were randomly generated to be between the corresponding low and high values. For loop, the following parameters were used:

$f(s)$ - equation that generates the packet length for sensor variable communication based on the sensor processing time. This equation was linear in s.

$g(c)$ - similar to the above and generates the packet length for communication of the compute values

The functions f and g are randomly computed online. That is, for each execution of the loop.

The most impressive results of the simulation is that the algorithm produces a feasible schedule and takes advantage of the fact that communication channel is scheduled as the packets become available. The same algorithm can also handle loops with varying priority if the control system uses a feedback mechanism for varying priority.

The simulation results specifically demonstrate the following. Figure 2 demonstrates that for loops of distinct priority, that is 1-10, 1 being the highest priority and 10 being the lowest priority, the scheduling algorithm that is feasible and that higher the priority, lower the waiting time. Figure 3 demonstrates the same for loops with random priority.

Figure 4-5 demonstrate that for a system with control loops with no priority, the waiting times were proportional to the length of the loops, thus reducing the overall control delay compared to any other algorithm.

Figure 6-7 demonstrate the execution rate of the control loops. Note that, the higher priority loops have higher execution rate than that of lower priority loops. To study the execution rate, the sensor times, the compute times and actuation times were set to be alike and f and g were set to a fixed δ.

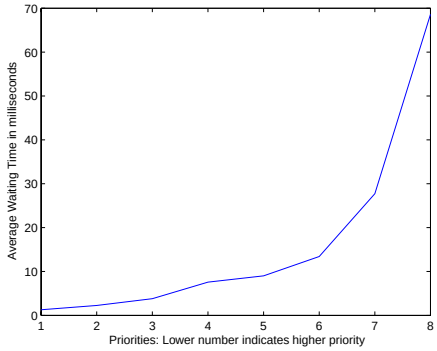


Figure 2: Control Loop priority versus Average Waiting Time.

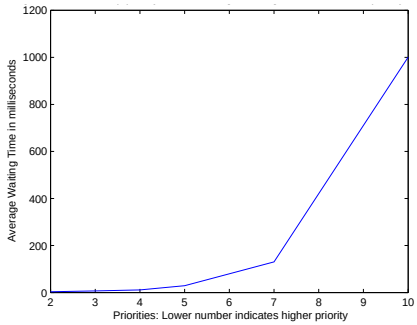


Figure 3: Control Loop priority versus Average Waiting Time with random priority for loops.

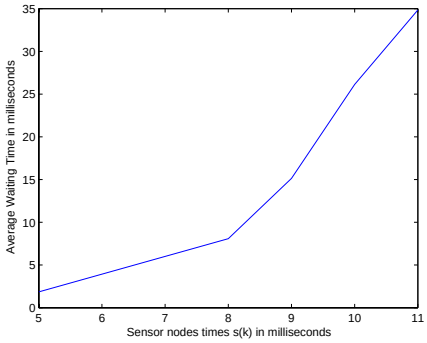


Figure 4: Sensor Node time versus average time to transmit sensor variables.

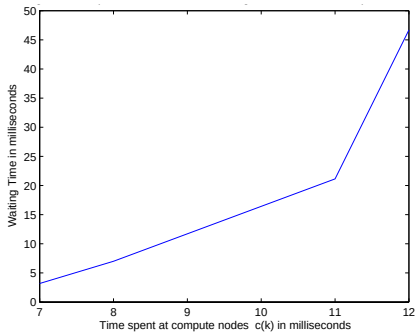


Figure 5: Compute node time versus average time to transmit computed results.

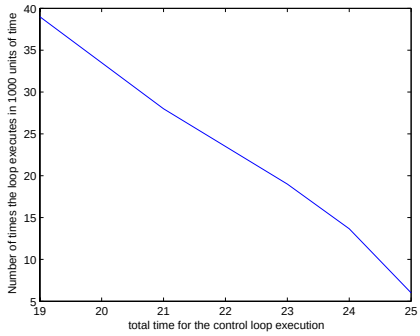


Figure 6: Control Loop Length versus Execution Rate for Equal Priority Loops.

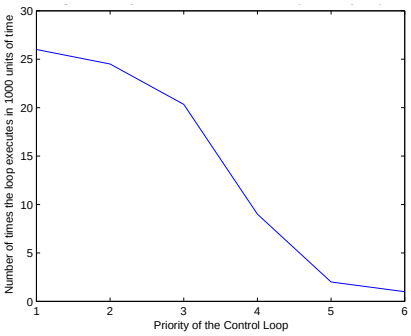


Figure 7: Priority versus Execution Rate for UnEqual Priority Loops.

4 CONCLUSIONS

In this paper we have a dynamic scheduling algorithm for control loops that takes into consideration the changing nature of communication needs within control systems. The algorithm handles loops of different priorities and different execution rates. The algorithm produces a feasible schedule and produces minimal control delay.

REFERENCES

Altenbernd, P. and Hansson, H. (2004). The slack method: A new method for static allocation of hard real-time tasks. *Real-Time Systems*, 15(2):103–130.

Blazewicz, J. and Ecker, K. (1994). Multiprocessor task scheduling with resource requirements. *Real Time Systems*, 6:37–53.

Peng, D.-T. and Shin, K. G. (1989). Static allocation of periodic tasks with precedence constraints distributed real-time systems. *9th International Conference on Distributed Computing Systems,*, pages 190–198.

Ramamritham, K. (1990). Allocation and scheduling of complex periodic tasks. *Proceedings of 10th International Conference on Distributed Computing Systems*, pages 108–115.

Santos, J., Ferro, E., Orozco, J., and Cayssials, R. (1997). A heuristic approach to the multitask-multiprocessor assignment problem using the empty-slots method. *Real-Time Systems*, 13(2):167–199.

Velasco, M., Fuertes, J. M., Lin, C., Marti, P., and Brandt, S. (2004). A control approach to bandwidth management in networked control systems. *Proc. 30th IEEE IECON*.

Xu, J. (1993). Multiprocessor scheduling of processes with release times, deadlines, precedence, and exclusion relations. *IEEE Transactions on Software Engineering*, 19(2):139–154.

Yepez, J., Marti, P., and Fuertes., J. M. (2003). Control loop scheduling paradigm in distributed control systems. *IECON '03. The 29th Annual Conference of the IEEE*, 2:1441– 1446.

FLEXIBLE ROBOT-BASED INLINE QUALITY MONITORING USING PICTURE-GIVING SENSORS

Chen-Ko Sung, Andreas Jacubasch and Thomas Müller
Fraunhofer Institute IITB, Fraunhoferstrasse 1, 76131 Karlsruhe, Germany
chen-ko.sung@iitb.fraunhofer.de, andreas.jacubasch@iitb.fraunhofer.de
thomas.mueller@iitb.fraunhofer.de

Keywords: Multi sensors, wide- and short-range sensors, intelligent sensors, sensor magazine, object detection and localization, learning-capable evaluation processes, topography, visual serving, flexible inline quality monitoring, real-time processing, robot, dynamic and automatic path planning, large-area production.

Abstract: As part of the ROBOSENS project, the IITB developed and tested a new four-step concept for multiple sensor quality monitoring. The robot-based system uses an array of test-specific short-range and wide-range sensors which make the inspection process more flexible and problem-specific. To test this innovative inline quality monitoring concept and to adapt it to customized tasks, a development and demonstration platform (DDP) was created. It consists of an industrial robot with various sensor ports - a so-called "sensor magazine" - with various task-specific, interchangeable sensors and a flexible transport system.

1 INTRODUCTION

A substantial reason for the hesitant use of picture-giving sensors for the monitoring of inspection is the insufficient flexibility of the used monitoring concepts in relation to changing setting of tasks. At present either one (user-specific) sensor or sensors with static arrangement for a certain task of inspection are used. This rigid approach is unsuitable for the inspection of variant products.

Robots with multiple intelligent sensors will be increasingly used in the future for demanding production and assembly tasks. An especially attractive area of application is the inline quality monitoring of complex, large-area production parts such as the aircraft fuselage (see Figure 1) or parts of the bodies of road and rail vehicles. For example hundred different mounting parts on an aircraft fuselage (about 4 m x 10 m in size) must be inspected, whether proper parts have been attached correctly.

The presented new four-step concept (see Figure 2 and sections 2-5) has been developed and realized at IITB for the flexible inline quality monitoring (Sung and Kuntze, 2006) with the following characteristics:

- Multiple sensor inline quality monitoring of large complex manufacturing parts;
- Complete quality assurance with minimum inspection expenditure;
- Large flexibility regarding frequently changing test tasks;
- On-line ability by minimization of the testing period.

All sensors are placed on a sensor magazine (see Figure 1) and are ready to use immediately after docking on the robot arm. The calibration of all sensors and the hand-eye calibration have to be done before the object localization task starts.

A special transport system like a monorail conveyor will probably be needed for the transportation of large objects. Such transport systems do not allow a precise positioning. The test object is free-hanging over the ground.

2 LOCALIZATION OF UNFIXED INDUSTRIAL TEST OBJECTS

As the first step of the presented quality monitoring chain, the exact position of a production piece is determined with a wide-range picture-giving sensor (see Figure 1), which is - depending on the object size - mounted in an adequate object distance, i.e. not necessarily fixed on an inspection robot's end-effector.

A marker-less localization calculates the exact object position in the scene. This procedure is based only on a 3D CAD-model of the test object or at

least a CAD-model which represents a composition of some of its relevant main parts. The CAD-model contours are projected into the current sensor images and they are matched with sub-pixel accuracy with corresponding lines extracted from the image (Müller, 2001).

Figure 3 shows a localization example. The CAD-model projection is displayed in yellow and the object coordinate system in pink color. The red pixels close to the yellow projection denote corresponding image line pixels which could automatically be extracted from the image plane. The calculated object pose (consisting of three parameters for the position in 3D scene space as well as three parameters for the orientation, see the red text in the upper part of the figure) can easily be transformed into the global scene coordinate system (displayed in green color).

Known test zones for detail inspection as well as associated sensor positions and orientations or required sensor trajectories (cf. section 3 and 4) can be defined with respect to the object coordinate system in an inspection preceding step. All the object based coordinates will be transformed online into the global scene coordinate system or the robot coordinate system with respect to the localization result, i.e. with respect to the position and orientation of the test object in the scene. The red, T-shaped overlay in Figure 3 shows an example for an optimal 3D motion trajectory (see the horizontal red line which is parallel to the object surface) together with the desired sensor's line of sight with respect to the object surface (the red line which points from a position in the middle of the trajectory towards the test object).

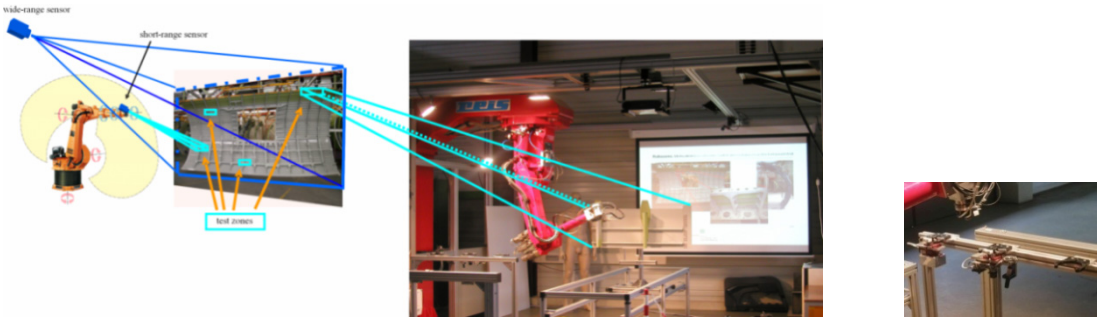


Figure 1: Quality monitoring of aircraft fuselages with wide- and short-range inspection sensors. Left: Test station and test environment. The movement of production pieces is carried out by monorail conveyors which do not allow precise positioning. Middle: Development and demonstration platform (DDP). Right: Sensor magazine.

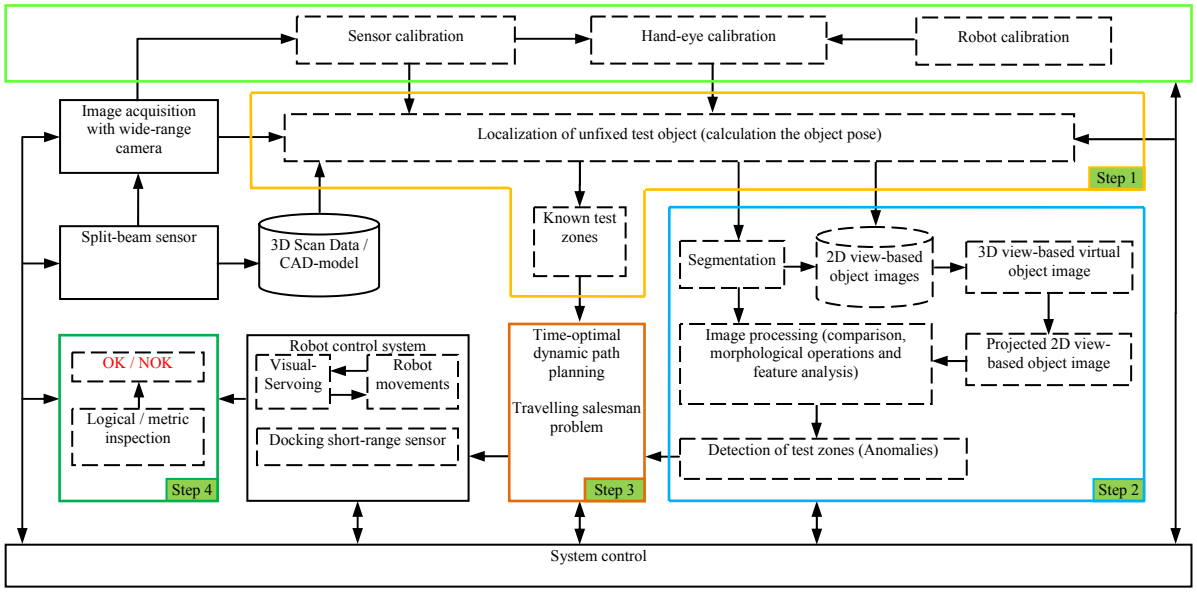


Figure 2: System overview. A new four-step concept for the flexible inline quality monitoring.

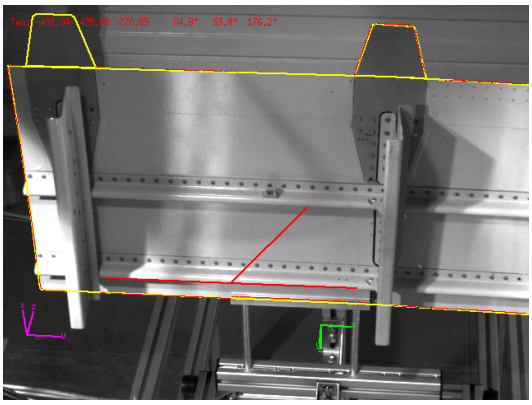


Figure 3: Localization of an object to be inspected and computation of an initial optimal inspection trajectory.

3 AUTOMATIC DETECTION OF TEST ZONES

Two approaches can be applied to find automatically anomalies on a test object. One is model-based comparison between the CAD-Model projection and the extracted image features (edges, corners, surfaces) to detect geometric differences (Veltkamp and Hagedoorn, 2001). Another one resembles probabilistic alignment (Pope and Lowe, 2000) to recognize unfamiliar zones between view-based object image and test image.

In this second step, we used purely image-based methods and some ideas of the probabilistic alignment to achieve a robust inline detection of anomalies under the assumption that the object view changes smoothly. The same wide-range camera for object localization was used for this step.

Using the result of object localization to segment an object from an image, a database with 2D object images can be built up in a separate learning step. We postulated that the views were limited either of the front side or the back side of the test object with small changes of viewing angles and furthermore postulated that we had constant lighting conditions in the environment.

We used the calibration matrix and the 2D object images to create a 3D view-based virtual object model at the 3D location where an actual test object was detected. The next process was to project the view-based virtual object model into the image plan. The interesting test zones (anomalies, see Figure 4) where detailed inspections were needed (see section 4 and 5) were detected within the segmented image area by the following steps:

- comparison between the projected view-based object image and the actual test image;
- morphological operations;
- Feature analysis.

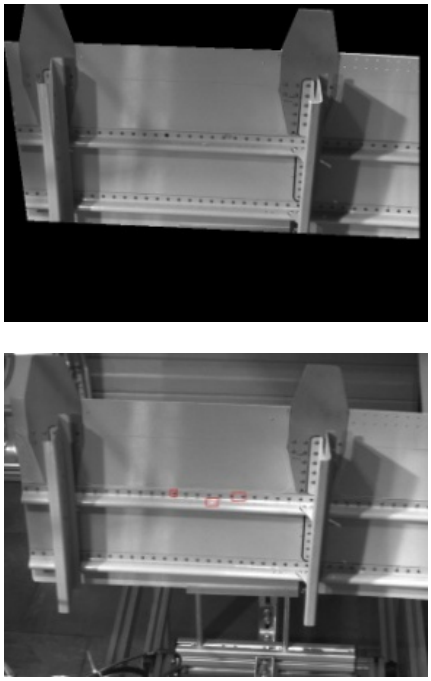


Figure 4: Upper: One of the segmented object images in the learning step. Only the segmented area in an image is relevant for the detection of anomalies. Lower: The automatic detected test zones are marked with red rectangles (overlays).

4 TIME-OPTIMAL DYNAMICAL PATH PLANNING

In the third step, an optimized inspection path plan is generated just in time, which is then carried out using various inspection-specific short-range sensors (e.g. cameras, feeler, etc.).

All the interesting test zones or the regions of interest (ROIs) have been found in the second step, but the path plan is not perfect yet. A time-optimal path has to be found from the supervising system.

The problem is closely related to the well known travelling salesman problem (TSP), which goes back to the early 1930s (Lawler et al., 1985; Applegate et al., 2006). The TSP is a problem in discrete or combinatorial optimization. It is a prominent illustration of a class of problems in computational complexity theory which are classified as NP-hard.

The total number of possible paths is calculated by: $\boldsymbol{M = (n - 1)}$. The definition of the TS-problem is based on the following assumptions:

- Modelled as a graph with nodes and edges;

- Graph is complete, this means that from each point there is a connection to any other point;
- The graph can be symmetric or asymmetric;
- The graph is metric, that means it complies the triangle inequality $C_{ij} \leq C_{ik} + C_{kj}$ (e.g. Euclidian metric, maximum metric).

Locking at the algorithms for solving TS-problems, there exist two different approaches:

Exact algorithms which guarantee a global optimal solution and heuristics, where the solution found is only locally optimal.

The most accepted exact algorithms which guarantee a global optimum are Branch-and-Cut Method, Brute-Force and Dynamic Programming. The major disadvantage of the exact algorithms mentioned above is the time consuming process finding the optimal solution. The most common heuristic algorithms used for the DSP are:

- Constructive heuristics: The Nearest-Neighbor-Heuristic chooses the neighbor with the shortest distance from the actual point. The Nearest-Insertion-Heuristic inserts in a starting path additional points;
- Iterative improvement: Post-Optimization-methods try to modify the actual sequence in order to shorten the overall distance (e.g. k-opt heuristic).

We used a heuristic algorithm with the following boundary conditions:

- The starting point has the lowest *x*-coordinate;
- The Nearest-Neighbor-Constructive heuristics look for the nearest neighbour starting with the first node and so on;
- The iterative improvement permutes single nodes or complete sub graphs randomly;
- Terminate, if there was no improvement after *n* tries.

The optimized path planning discussed above was tested at the DDP with a realistic scenario. Given a work piece of 1 by 0.5 square meter, the outputs of the second step (see section 3) are 15 detected ROIs. This would lead to a total number of about 43.6 billion possible different paths.

Starting with a 1st guess as outlined with an associated path length set to 100 %, after 15 main iteration loops the path lengths drops down to nearly 50 % of the first one, and no better achievement could be found (see Figure 5). The calculation time for the iterated optimal path was less than 1 sec. on a commercial PC, Intel Pentium 4 with 3 GHz, and took place while the robot moved to the starting position of the inspection path.

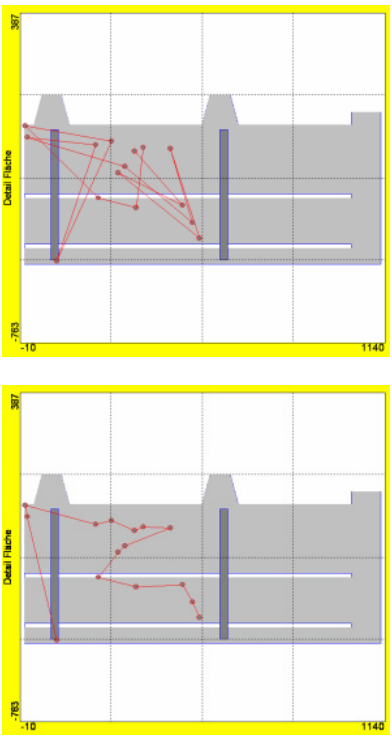


Figure 5: Upper: initial path; Lower: final path.

5 VISION-BASED INSPECTION

In the fourth step, the robot uses those sensors which are necessary for a given inspection path plan and guides them along an optimal motion trajectory into the previously-identified ROIs for detailed inspection. In these ROIs a qualitative comparison of the observed actual topography with the modeled target topography is made using image-processing methods. In addition, quantitative scanning and measurement of selected production parameters can be carried out.

For the navigation and position control of the robotic movement with regard to the imprecisely-guided production object as well as for the comparison of the observed actual topography with the target topography, reference models are required.

These models, using suitable wide-range and short-range sensors, were scanned in a separate learning step prior to the generation of the automated inspection path plan. Two sensors have been used for our work: A 3D split-beam sensor is used (Deutscher et al., 2003) for the metric test task (see Figure 6) and a short-range inspection camera with a circular lighting is used for the logical test task. For a fuselage, for example, it can be determined if construction elements are missing and/or if certain bore diameters are true to size.

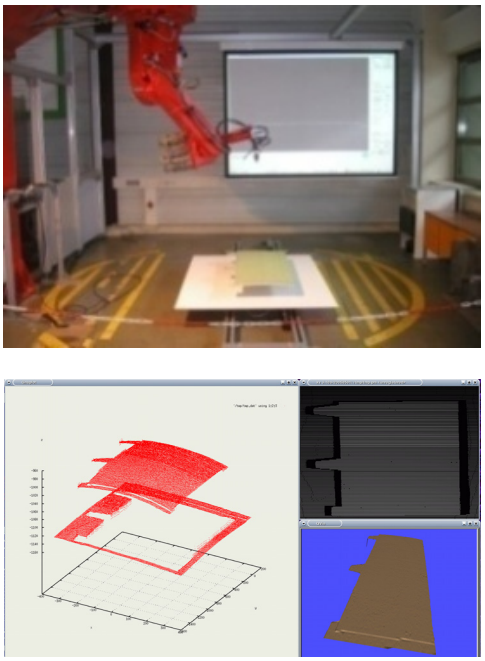


Figure 6: A split-beam technique captures the structure of a 3D object (upper part) and translates it into a graphic model (lower part).

By using the proposed, robot-based concepts of multiple sensor quality monitoring, the customary use of expensive 3D CAD-models of the test objects for high-precision CNC controlled machine tools or coordinate inspection machines becomes, in most instances, unnecessary.

An intelligent, sensor-based distance-control concept (Visual-Servoing-Principle) accurately controls the robot's movements with regard to the work piece and prevents possible collisions with unexpected obstacles.

6 CONCLUSIONS AND FUTURE WORK

A development and demonstration platform (DDP) for flexible inline quality monitoring using picture-giving sensors was created.

The primary goal of the DDP is to investigate, optimize and demonstrate to potential cooperation partners how the system can be applied to reduce effort and to increase flexibility. For example, it can be used in the robot-based coordination of short- and wide-range monitoring, for the introduction of learning-capable evaluation processes, as a tool for visualizing results and for user interaction, as well as for the flexible networking and integration of various wide- and short-range sensors.

Further inspection sensors, which are based on another measurement principle, will be developed soon on the sensor magazine and made available for the surface testing. Investigations and pre-developments for further (complex) applications can be realized with the platform at small expenditure. The applications for example can look like:

- The surface inspection of the outside and the structural examination of the inside of a car door;
- The crawler-type vehicle order supervision within the range of a car window.

ACKNOWLEDGEMENTS

This research was supported by the „Fraunhofer-Gesellschaft zur angewandten Forschung e.V." internal Program.

REFERENCES

Applegate, D. L., Bixby, R. E., Chvátal, V., Cook, W. J., (2006). *The Traveling Salesman Problem: A Computational Study*. Princeton University Press. ISBN 978-0-691-12993-8.

Deutscher, R., Munser, R., Hartrumpf M., (2003). Detection and Measurement of Damages in Sewer Pipes with a 3D-structured Light Projection Sensor, *In: tm - Technisches Messen 70,* 2003(07):338-345.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., Shmoys, D. B., (1985). *The Travelling Salesman Problem. A Guided Tour of Combinatorial Optimization*. Wiley, Chichester 1985. ISBN 0-471-90413-9.

Müller, Th., (2001). Modellbasierte Lokalisation und Verfolgung für sichtsystemgestützte Regelungen, Dissertation an der Universität Karlsruhe (TH), 8. Februar 2001.

Pope, A. R., Lowe, D. G., (2000). Probabilistic Models of Appearance for 3D Object Recognition. *International Journal of Computer Vision,* 40(2):149–167.

Sung, C.-K., Kuntze, H-B., (2006). Flexible roboter-basierte Qualitätsüberwachung mit bildgebenden Sensoren, *Sensor Magazin*, Magazin Verlag Hightech Publications KG, Bad Nenndorf, 2006(3):24-26.

Veltkamp, R.C., Hagedoorn, M., (2001). State-of-the-art in shape matching. In M. Lew (Ed.), *Principles of Visual Information Retrieval*, Springer.

REDUCED ORDER H_∞ SYNTHESIS USING A PARTICLE SWARM OPTIMIZATION METHOD

Guillaume Sandou, Gilles Duc and Patrick Boucher
Supelec Automatic Control Department, 3 rue Joliot Curie, 91192 Gif-sur-Yvette, France
guillaume.sandou@supelec.fr, gilles.duc@supelec.fr, patrick.boucher@supelec.fr

Keywords: Particle swarm optimization, Reduced-order H_∞ synthesis, metaheuristics.

Abstract: H_∞ controller synthesis is a well known design method for which efficient dedicated methods have been developed. However, such methods compute a full order controller which has often to be reduced to be implemented. Indeed, the reduced order H_∞ synthesis is a non convex optimization problem due to rank constraints. In this paper, a particle swarm optimization method is used to solve such a problem. Numerical results show that the computed controller has a lower H_∞ norm than the controller computed from a classical Hankel reduction of the full order H_∞ controller.

1 INTRODUCTION

H_∞ synthesis is an efficient tool, which aims to compute controllers in a closed loop framework, achieving high and various performances. Two principal solution methods have been developed for this purpose, based on Linear Matrix Inequalities (Gahinet and Apkarian, 1994), or on Riccatti equation solutions (Glover and Doyle, 1988). The main drawback of such approaches is the controller order: H_∞ synthesis provides a controller whose order is the same as the synthesis model. A classical way to get low order controllers is to reduce the full controller, for example with a Hankel decomposition method. However, this approach may lead to a high H_∞ norm of the closed loop system and a high sensitivity to high frequency noises. To avoid high order controllers, the H_∞ optimization problem can be solved, adding some order constraints. However, this kind of constraints is expressed with rank constraints and the reduced-order synthesis problem appears to be a non convex optimization problem, and classical algorithms may fail in the solution.

In this paper, a new approach is proposed, using Particle Swarm Optimization (PSO). With such a method, the optimality of the computed solution can never be guaranteed, but the structure of costs and constraints is not an essential point. The mathematical descriptions of the full and reduced order H_∞ synthesis are called up in section 2. PSO is presented in section 3. The proposed algorithm is used for the multivariable control of a pendulum in the cart. Results are given in section 4. Finally conclusions are drawn in section 5.

2 REDUCED-ORDER H_∞ SYNTHESIS

2.1 Full-Order H_∞ Synthesis

Consider the closed loop of figure 1, with s the Laplace variable. The transfer matrix is:

$$\begin{bmatrix}\varepsilon(s)\\ u(s)\end{bmatrix}=\begin{bmatrix}S & -SG\\ KS & -KSG\end{bmatrix}\begin{bmatrix}r(s)\\ d(s)\end{bmatrix}=T(s)\begin{bmatrix}r(s)\\ d(s)\end{bmatrix} \quad (1)$$

with $S(s)=(I+G(s)K(s)^{-1}$

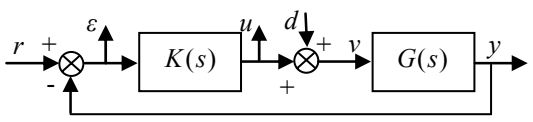


Figure 1: Classical closed loop system.

The H_∞ synthesis problem is defined as follows. Find a stabilizing controller $K(s)$ such that:

$$\gamma=\min_{K(s)}\left\|T(s)\right\|_\infty \quad (2)$$

It can be reformulated into a convex problem and solved with Riccati equations or LMI formulations. This solution is called "full order" synthesis, as the solution of problem (2) is a controller $K(s)$ whose order is equal to the order of $G(s)$. Some design filters are added to the synthesis model to tune the performances (figure 2). The new system is:

$$\begin{bmatrix} e_1(s) \\ e_2(s) \end{bmatrix} = \begin{bmatrix} W_1 S & -W_1 S G W_3 \\ W_2 K S & -W_2 K S G W_3 \end{bmatrix} \begin{bmatrix} r(s) \\ d(s) \end{bmatrix} \quad (3)$$

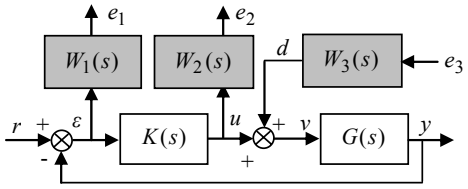


Figure 2: Synthesis model.

Finally, solving the H_∞ problem for this system induces frequency dependent constraints for each transfer of matrix (1).

2.2 Reduced-Order H_∞ Synthesis

The reduced-order H_∞ problem refers to the solution of the following optimization problem:

$$\gamma = \min_{K(s)} \left\| T(s) \right\|_\infty \text{ s.t. } \partial^\circ K(s) = n_r \quad (4)$$

where $\partial^\circ K$ denotes the order of $K(s)$, and n_r is strictly less than the order of the synthesis model. It can be reformulated into LMI equations by adding rank constraints on matrices, loosing the property of convexity (El Ghaoui et al., 1997).

3 PSO ALGORITHM

PSO was introduced by Russel and Eberhart (Eberhart and Kennedy, 1995). P particles are moving in the search space. x_p^k (v_p^k) is the position (velocity) of particle p at iteration k, b_p^k is the best position found by particle p until iteration k, $V(x_p^k) \subset \{1,2,\ldots,P\}$ is the set of "friend particles" of particle p at iteration k, g_p^k best position found by the friend particles of particle p until iteration k, and $\otimes$ element wise multiplication of vectors. The particles move in the search space according to the following transition rule:

$$\begin{aligned} v_p^{k+1} &= w.v_p^k + c_1 \otimes (b_p^k - x_p^k) \\ &\quad + c_2 \otimes (g_p^k - x_p^k) \\ x_p^{k+1} &= x_p^k + v_p^{k+1} \end{aligned} \quad (5)$$

In this equation, w is the inertia factor and c_1, c_2 are random vectors in the range $[0, \bar{c}]$. The choice of parameters is very important to ensure the satisfying convergence of the algorithm, see (Eberhart and Shi, 2000). However, it is not in the scope of this study to look for fine strategies of tuning. Thus, standard values, given in (Kennedy and Clerc, 2006) will be used P: $P = 10 + \sqrt{n}$ (n is the number of optimization variables), $w = 1/(2\ln(2))$, $\bar{c} = 0{,}5 + \ln(2)$, $\dim(V(x_p^k)) \le 3$.

4 NUMERICAL RESULTS

4.1 Case Study

The proposed method has been tested for a pendulum in the cart (figure 3).

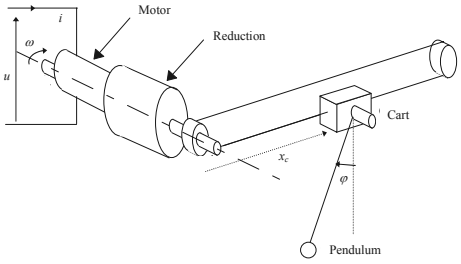


Figure 3: Pendulum in the cart.

The system can be modelled by:

$$\begin{aligned} L\frac{d\,i(t)}{d\,t} + R\,i(t) + K_e\,\omega(t) &= u(t) \\ J\frac{d\,\omega(t)}{d\,t} + f\,\omega(t) + d(t) &= K_e i(t) \\ \frac{d\,x_c(t)}{d\,t} &= \frac{r}{N}\omega(t) \\ \cos(\varphi)\frac{d^2 x_c}{d\,t^2} + l\frac{d^2\varphi}{d\,t^2} + \alpha\frac{d\,\varphi}{d\,t} + g\sin(\varphi) &= 0 \end{aligned} \quad (6)$$

Variables are i and u (current and input voltage of the motor), ω (rotation speed), x_c (position of the cart), φ (angle of the pendulum), d (disturbance moment). Constants are L, R, J (motor inductor, resistance, inertia), K_e (electromagnetic constant), f (friction coefficient), r (pulley radius), N (gear reduction), l (pendulum length), α (pendulum friction coefficient) and g (weight acceleration).

Specifications are: tracking of the reference of figure 4, no steady state error, time response ≤ 6s, rejection of disturbance d and $|\varphi(t)| \leq 0.05$rad ;

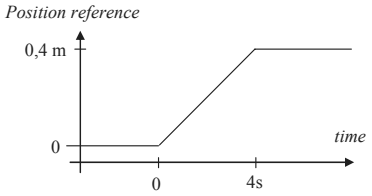


Figure 4: Position reference.

4.2 Three Outputs H_∞ Synthesis

To show the versatility of the method, a three measurement controller is designed (synthesis model of figure 5). The filters are defined as:

$$W_1 = \frac{1}{2} \cdot \frac{s+1.7}{s+0.0009},\ W_2 = 100 \cdot \frac{s+2}{s+2000} \qquad (8)$$
$$W_3 = 0.01,\ W_4 = 2,\ W_5 = 1,\ W_6 = 0.1$$

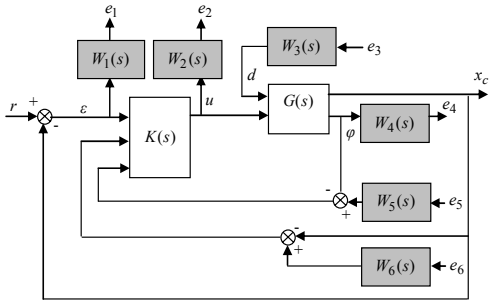


Figure 5: Synthesis model for the "3 output" case.

The solution of the full order synthesis leads to a H_∞ norm $\gamma = 1.06$. The full-order controller is of order 6. The Hankel reduction leads to a very large H_∞ norm $\gamma = 56.7$ for the order 2 controller. A controller is computed by the PSO algorithm, with the filters of the full order synthesis. Results are given in table 1 for 100 tests. Computation times are 30s (Pentium IV, 2GHz; Matlab 6.5).

Table 1: Optimisation results for the three output case.

Worst	Best	Mean
$\|\ \|_\infty = 4.53$	$\|\ \|_\infty = 2.60$	$\|\ \|_\infty = 3.50$

Figure 6 gives the Bode diagram of the transfers of matrix (1) (full order, Hankel reduction controller, and PSO). Figure 7 represents the corresponding time responses. As can be seen, results of the Hankel reduction controller are quite similar as for the full order controller, except at high frequencies. Figure 8 and 9 give the same results obtained with the mean controller of the PSO method. Note first that the response of $\varphi(t)$ is quite similar as the previous ones and remains therefore satisfying. A slight overshoot is observed on the reference tracking.

However, consider figure 10, where a measurement noise d_m has been added on the cart position. The control input u is represented both for Hankel reduction and PSO controllers. As can be seen from figure 6, Hankel reduction leads to a modification of the closed loop transfers for high frequencies. As a result, high gains for high frequencies lead to an amplification of measurement noises and thus to chattering control inputs. On the contrary, the reduced order synthesis leads to closed loop systems with smaller H_∞ norm. The system is more robust against measurement disturbances.

5 CONCLUSIONS

In this paper, a metaheuristic method based on Particle Swarm Optimization has been presented. PSO is a stochastic optimization method which does not require any particular structure for costs and constraints. As a result, the method can be used to optimize many kinds of criterions and solve non convex, non linear or non analytic problems. In this paper, the method is used to solve a well known problem of modern Automatic Control, namely the reduced order H_∞ synthesis. The problem is known to be a non convex problem, for which the traditional approach is an a posteriori reduction of the full order synthesis. Results, computed for a pendulum in the cart have shown the viability of the approach. Computed controllers lead to a slight decrease of nominal performances but to a more

robust controller with an important decrease of the closed loop H_∞ norm.

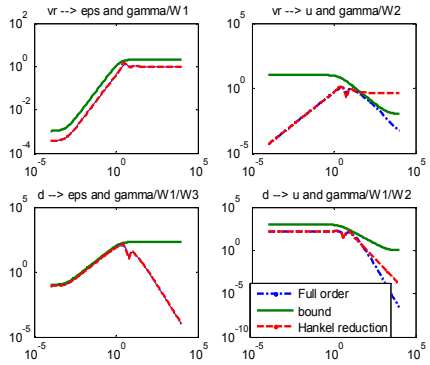


Figure 6: Bode transfer of full order and Hankel reduction.

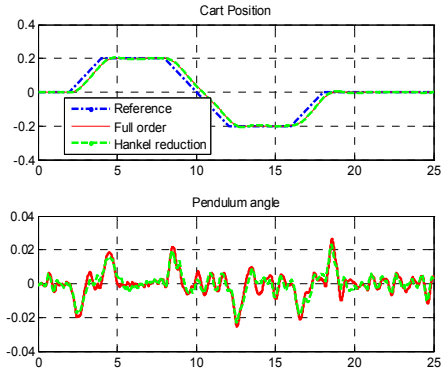


Figure 7: Time response - full order and Hankel reduction.

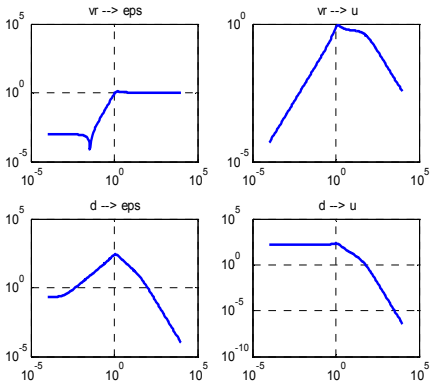


Figure 8: Bode transfer for PSO controller.

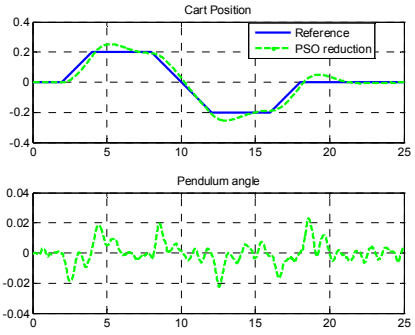


Figure 9: Time response for PSO controller.

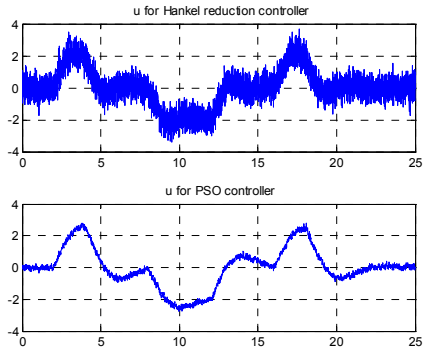


Figure 10: Control input for Hankel reduction and PSO controllers.

REFERENCES

Eberhart, R. C, Kennedy, J., 1995. A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan. pp. 39-43.

Eberhart, R. C., Shi, Y., 2000. Comparing inertia weigthts and constriction factors in particle swarm optimization. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2000), San Diego, CA, USA, p 84-88.

El Ghaoui, L., Oustry, F., AitRami, M., 1997. A cone complementary linearization algorithm for static output feedback and related problems. In: *IEEE Transactions on Automatic Control*, Vol. 42(8), pp. 1171-1176.

Gahinet, P., Apkarian P., 1994. A linear matrix inequality approach to H_∞ control. In *International Journal of Robust and Nonlinear Control*, Vol. 4, pp. 421-448.

Glover, K., Doyle, J.C., 1988. State-state formulae for all stabilizing controllers that satisfy an H_∞-norm bound and relations to risk sensitivity. In *Systems and Control Letters*, Vol.11, pp. 167-172.

Kennedy, J. and M. Clerc, 2006. Standard PSO. http://www.particleswarm.info/Standard_PSO_2006.c

EVOLUTION OF A MOBILE ROBOT'S NEUROCONTROLLER ON THE GRASPING TASK

Is Genetic also Generic?

Philippe Lucidarme
Lisa, University of Angers, France
philippe.lucidarme@univ-angers.fr

Keywords: Evolutionary algorithm, mobile robot, artificial neural networks controller, grasping task.

Abstract: This paper presents a survey on the generic evolution of mobile robot' neurocontrollers with a particular focus on the capacity to adapt these controllers in several environments. Several experiments on the example of the grasping task (autonomous vacuum cleaner for example) are performed and the results show that the produced neurocontroller is dedicated to the trained conditions and cannot be considered as generic. The last part of the paper discusses of the necessary changes in the fitness function in order to produce generic neurocontrollers.

1 INTRODUCTION

In the past years, biology has been an inspiration for computer science researches. Genetics algorithms and artificial neural networks are probably the best illustration. Evolutionary algorithms are today used to optimize, classify or control in a large number of problems. The efficiency of the methods has been experimentally and sometime theoretically proven on given problems (Jansen, 2002), (Bäck, 1996) and (Floreano, 1994). In the particular field of robotics, especially in the case of real robots, proving the efficiency of the methods is very hard due to the complexity of the interactions between the environment and the robot. In spite of this, evolutionary algorithms are commonly used to optimize the parameters of robot's controllers.

The first experiment in the field of robotics has been performed in 1994 by Dario Floreano and Francesco Mondada (Floreano, 1994). Based on the used of a genetic algorithm, the aim of this experiment was to optimize the parameters of an artificial neural controller in order to generate an obstacle avoidance behavior. The experiment was performed on a real Kephera robot (Mondada 1993) and required 100 generations of 80 individuals. Each generation lasts 39 minutes. The results were remarkable, after 50 generations (32 hours) the robot already performed a behavior close to the optimal. Note that the Khepera robot is not a symmetrical robot: front face has 6 proximity sensors versus 2 for the rear face. During the evolution, the robot naturally selected the front face as the best direction. Few years later, the same team extended the experiment on a bigger robot with a different proximity sensors disposition (Floreano 1998). They continued the evolution on a Koala robot (described in section 4.4 of (Nolfi 2000)). In approximately thirty generations the best individuals reported fitness values similar to the experiment previously described with the Khepera robot (Floreano, 1994).

The first experiment proved the possibility of using evolutionary algorithm in order to learn basic behaviors on real robots. The second experiment proved that the previous results are platform independent, and this result can even be extended: from the neurocontroller point of view, inputs are the proximity sensors, and outputs are motor's commands. It means that the geometry and the kinematics of the robot are external to the controller. Then, the results of this second experiment can be extended to the environment, as the authors explained in (Nolfi 2000): *"From the point of view of the neurocontroller, changing the sensory motor characteristics of the robot is just another way of modifying the environment"*.

These results have been exploited on several experiments like motion planning (Ahuactzin, 1992) or humanoid walking (. Yamasaki 2002) with a recurrent argument: the adaptive capacity of the evolutionary algorithms. According to the previous explanations, this capacity cannot be contested, but what about the neurocontroller? In fact, the

neurocontroller is not generic; it has been optimized for the environment where the experiment has been performed. It means that if the robot needs to evolve in a different environment the evolutionary process needs to be restarted like for the Koala robot. In practice, and especially on real robots, the evolutionary algorithms need to be stopped to avoid performing dump behaviors due to unfortunate crossovers. The study presented in this paper evaluates the faculty of a neurocontroller to be adapted in several environments.

Second section of the paper describes the context of the experiment, based on a grasping task (Arkin, 1992). Introduced by R.C. Arkin in the 90's, this task consists in exploring a given environment (for example for mowing or painting the floor).

From section three, the following of the survey is based on a methodology inspired from biology: evolutionary algorithms are used to generate eight neurocontrollers in eight different environments. These neurocontrollers are stored as standard behavior and are compared in the seven other environments to evaluate there performances in different contexts.

The fourth part of the paper introduces a new experiment where the robot is trained in the eight environments: the fitness function is the average performance. Results are analyzed and compared with the standards previously defined.

The last part of the paper introduces a new fitness function based on the performance in the worst environment. A general conclusion ends the paper.

2 EVOLUTIONARY ALGORITHMS ON THE GRASPING TASK

2.1 Grasping Task

As explained in the introduction, the grasping task has been introduced in a paper on multi-agents systems in 1992 by R.C. Arkin. This task has been choosen because the duration of the evaluation of one individual can be bounded and also because the fitness function is easy to evaluate in simulation (explored area divided by total surface). Note that another kind of tasks would have been used like obstacles avoidance or target tracking.

Due to the duration of the experiments, all the presented results are obtained by simulation. Note also that our purpose in not to obtain an efficient neurocontroller on real robots but to compare the results in several contexts. Real experiments suffer from noise on sensors, wheels slipping or battery discharge that make the comparison sometime difficult. In spite of this, the simulator computes the model of the real robot Type 1 (described in the next section) and the environments are scaled around this robot. The simulator is designed with a library of eight environments. Each environment has a squared shape (length of the side: 3 meters). The disposal of walls has been chosen so as to do three kinds of environments: with large spaces (Figure 1.a,b,c and f) , with narrow corridors (Figure 1.d and g.) and mixed (Figure 1.e and h).

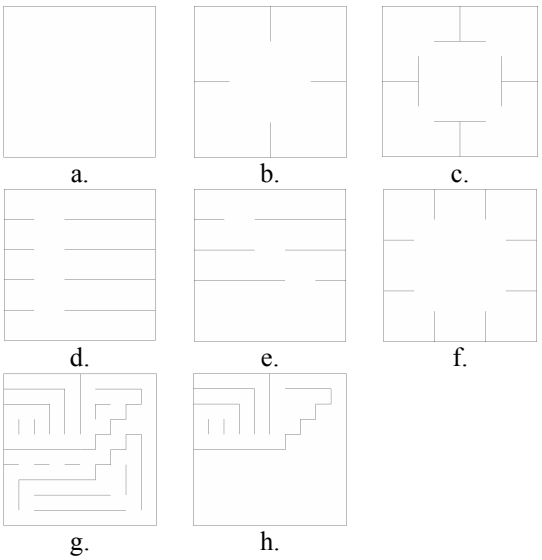


Figure 1: The eight environments used to compare the learned behaviors.

2.2 The Mobile Robot Type 1

The kinematics model of the robot and sensor's disposal are similar to the robot Type 1 described in (Lucidarme, 2006). It has a 10 cm-height and 13 cm-diameter cylindrical shape (figure 5). Two wheels actuate it. Two small passive ball-in-socket units ensure the stability in place of usual castor-wheels. DC motors equipped with incremental encoders (352 pulses per wheel's revolution) control the wheels. The encoders may be used for both speed control and robot localization by odometry. The robot is surrounded with 16 infrared emitters and 8 receivers (shown on figure 2 and 3). The sensors use a carrier frequency of 40 kHz for a good noise rejection. An embedded PC (80486 DX with 66 MHz clock) manages the robot.

Figure 3. shows the model used in the simulator, especially the sensor's positions. The kinematics model used in the simulator is described by equation 1. This robot has been chosen because in the case where real experiments would have been necessary,

Type 1 has many of the characteristics required by the evolutionary approach: fully embedded computation power (x86 processor), up to two hours of autonomy (Li-ion batteries) and large memory capacity (compact-flash).

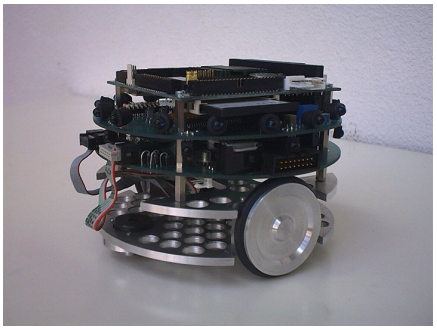


Figure 2: Picture of the mobile robot Type 1 equipped with infrared proximity sensors and the embedded PC104 visible on the top.

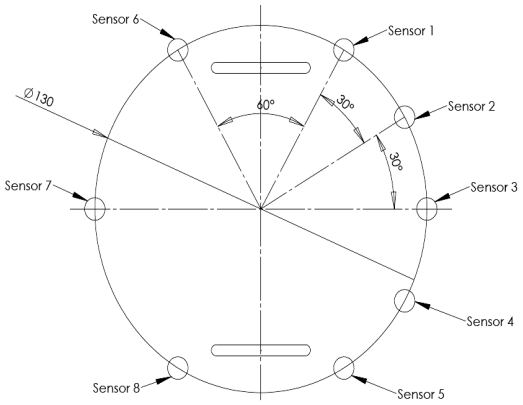


Figure 3: Description of the model used in the simulator and position of the sensors (Dimensions are in mm).

$$\begin{bmatrix} \frac{dX}{dt} \\ \frac{dY}{dt} \\ \frac{d\Psi}{dt} \end{bmatrix} = \frac{r}{2} \begin{bmatrix} \cos(\Psi) & \cos(\Psi) \\ \sin(\Psi) & \sin(\Psi) \\ \frac{1}{l} & -\frac{1}{l} \end{bmatrix} \cdot \begin{bmatrix} \Omega_r \\ \Omega_l \end{bmatrix} \quad (1)$$

where :

- X and Y are the coordinate of the robot in the environment frame,
- Ψ is the orientation in the same frame,
- Ω_l and Ω_r are the angular speed of the left and right wheels,
- r is the wheel's rayon,
- l is the distance between two wheels.

2.3 The Neurocontroller

As explain in (Floreano, 1994) and (Haussler, 1995) genetic algorithms can be used to train and optimize artificial neural networks (ANNs). Such solution has been selected here for its interesting link with biology and its anteriority in the field of robotics. Previous works (Braitenberg, 1986) prove that simple neurocontroller can be used to performed obstacle avoidance. Assuming that the grasping task is similar to an obstacle avoidance behavior from the neurocontroller point of view, the same structure for the neural network without hidden layer has been chosen (Figure 4 shows the neural network).

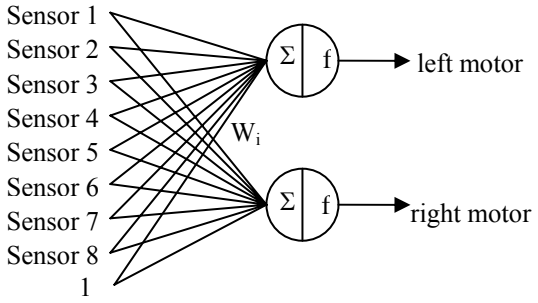


Figure 4: Neurocontroller's structure.

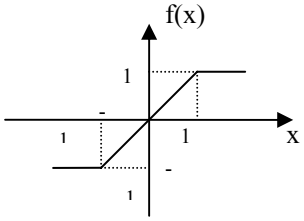


Figure 5: Transfer function for each perceptron.

In order to homogenize the simulation and the result analysis, all the data are scaled in the interval [-1;1]. For example, in the case of the wheel's angular speed: +1 applied to the motors is equivalent to the maximum speed (0.1 rad.s^{-1}). For the same reason, the synaptic weights are bounded in the same interval and the output of the perceptron is also bounded (Figure 5) to reproduce the mechanical characteristics of the motors. Proximity sensor's data is applied on the input of the network that computes the command on each motor. Note that a synaptic link with a constant value applied to the input (equal to one) has been added allowing the robot to move when none of the sensors are providing a value, i.e. when all the c_i are equal to zero. As usual, the neural network is just a friendly representation of a mathematical expression (equation 2). In this

equation, all the parameters are known except the 18 synaptic weights, optimized by the evolutionary algorithm.

$$\Omega_l = f(\sum_{i=1}^{8} w_i.c_i + w_9)$$
$$\Omega_r = f(\sum_{i=1}^{8} w_{i+9}.c_i + w_{18}) \quad (2)$$

where :

- Ω_l and Ω_r : command applied on the motors,
- w_i , $i \in [1;9]$: synaptic weight for the left motor,
- w_i , $i \in [9;18]$: synaptic weight for the right motor,
- c_i : distance detected by the sensor i.

2.4 The Evolutionary Algorithm

The algorithm is based on a classical genetic approach described as follow.

2.4.1 Chromosome

As the structure of the neurocontroller has been fixed, only the synaptic weights have to be optimized. The chromosomes are only containing these weights. An intuitive approach consists in coding the weights in binary, providing a series of 0 and 1. The drawback of this approach results in the most important influence of most significant bit during crossovers and mutations. To avoid this problem, an elementary component of the chromosome is not a 0 or a 1, but the weight himself as described in table 1.

Table 1: Structure of a chromosome.

w_1	W_2	…	w_i	…	w_{17}	w_{18}

2.4.2 Crossovers

Crossovers are performed with two individuals selected from previous generation. The selection of the individual is based on the roulette-wheel reproduction as described in (Nolfi, 2000) that allows the best individual to be statistically selected more frequently. The probability for an individual *n* to be selected is given by the equation 3.

Once the two parents are selected, one of them is randomly selected (each with a probability of 0.5) to provide the first gene, and this process is repeated 18 times (one for each weight). The crossover's strategy is multipoint as described in (Mitchell, 1997).

$$P_n = \frac{f_n}{\sum_{i=0}^{N} fi} \quad (3)$$

where :

- P_n : probability of selection for the individual n,
- f_j : fitness of the individual *j*,
- N : number of individuals in the previous generation

2.4.3 Mutation

According to the strategy chosen, the mutation process is very important. Assume that the optimal weights for a given gene aren't present in none of the 100 individuals: without mutation, it is strictly impossible to find the optimal solution. Mutations are indeed very important and the mutation rate must be high enough to ensure a good exploration of the space. The mutation process is performed after the crossovers; 10% of the new individuals are randomly selected. For each individual a gene is randomly selected and replaced by a random value in the range [-1,1]. For each draw, the probability is uniform.

2.4.4 Fitness Function

As our goal is a grasping task, the fitness function must be linked with the explored area. The environment is sampled with a sampling rate of 30 cm for each axis. As the size of the environment is 3m x 3m the space is divided into 100 squares. At the end of the evaluation of each individual, the fitness function is computed with the equation 4. Figure 6 shows a snapshoot of the simulator.

$$f_j = \frac{N_{\exp lored}}{N_{Total}} \quad (4)$$

where :

- f_j : fitness of the individual j,
- $N_{explored}$: number of squares explored
- N_{Total} : total number of squares

2.4.5 Parameters

In the first version of the simulator, a noise was added on the motors. After analyzing the results we discovered that this noise prevented from comparing

the results. We decided to eliminate this noise in order to make the simulator deterministic. For the same reason, the initial position of the robot is always located at the same place (at the top left, c.f. Figure 6) to prevent from favoring individuals.

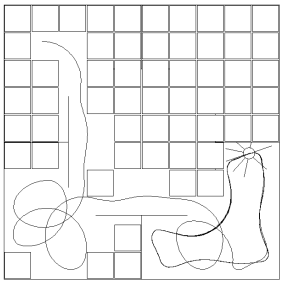


Figure 6: Snapshoot of the simulator showing the trajectory of the robot and the explored area. The current fitness of the robot is 0.43 (43 explored squares divided by 100 total squares).

Note that collisions between robot and walls are considered. As the neurocontroller has been designed without hidden layer, it is impossible for a jammed robot to escape from a collision. To decrease the computation time, when a robot is jammed its evaluation is stopped and the fitness is computed. Table 2 describes the parameters of the simulator.

Table 2: Parameters of the simulations.

Description	Name	Value
Wheel's rayon	r	0.05 m
Distance between wheels	l	0.1 m
Sensor's range	Sr	0.2 m
Maximum angular speed of the wheels	Ω_{max}	0.1 rad.s^{-1}
Size of the environment	-	3x3 m
Sampling rate (space)	-	0.3m
Size of the population	$N_{individual}$	100 ind.
Mutation rate	-	10%
Sampling rate (time)	Δ_t	0.01 s
Duration of an evaluation	-	30 s
Number of generations	-	250 gen.

3 STANDARD BEHAVIORS

3.1 Standard Experiments

For each environment a neurocontroller has been trained. This controller is considered as the standard behavior in the following. The evolution of the 250 generations lasts around two hours on a desktop computer. Figure 7 shows the evolution of an individual in the environment c. Figure 8 shows the evolution of the fitness. For each environment, an efficient strategy has been generated that confirms the relevance of the used parameters.

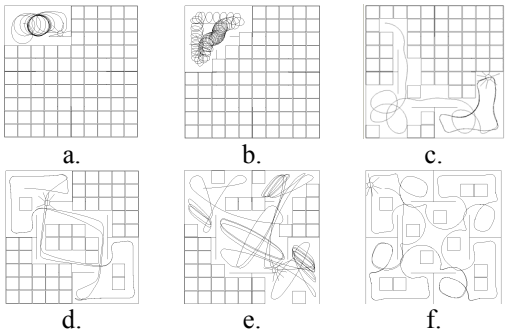


Figure 7: Evolution of the robot's behavior in the environment c. (see Fig.1) at generations 2,3,6,13,17 and 162 with respective fitness: 0.15, 0.19, 0.43, 0.51, 0.66 and 0.93.

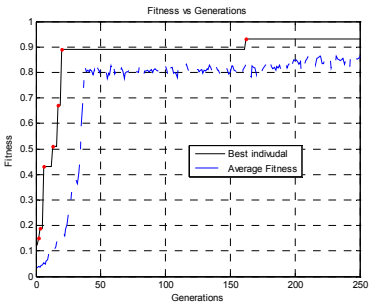


Figure 8: The fitness of the individuals (Fitness of the best individual (continuous line) for each generation and average (dotted line) of the whole population).

Analyzing the results shows that efficient behaviors can be classified in three categories:

- performing epicycloidals trajectories (Figure 9.a),
- moving straight and avoiding obstacles (Figure 9.b)
- wall following (Figure 9.c)

Best strategies are usually a mixed of the previous behaviors (Figure 9.d)

3.2 Swapping the Environments

In order to evaluate how generic are the produced behaviors, each individual is placed in the seven other environments. Note that the genetic process is stopped. Table 3 summarizes the results. For example bolded 38% presents the performance in the environment a of the individual trained in the

environment d. Grey cells show the best performance for each environment. This agrees with the diagonal that represent the performance of each individual in "its" environment except for environment d. and e. where individuals are equally ranked, probably due to the fact that environments d. and e. are quiet similar.

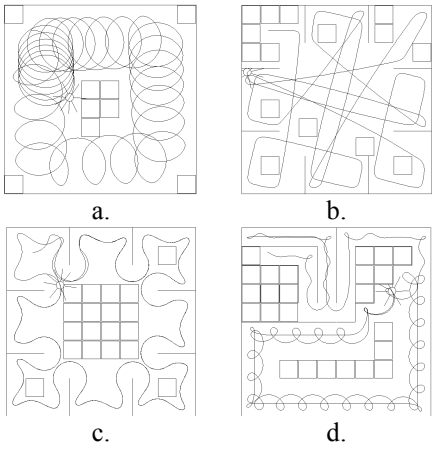


Figure 9: Examples of strategies used for exploring the environment.

This preliminary result allows us two conclusions: the neurocontroller produced by the genetic process cannot be considered as generic. An nice illustration is shown on table 3.: the performance in the environment g. is very poor except for the individuals trained in this kind of environment. These results also confirm that genetic algorithm may be considered as generic: the produced behavior is nicely adapted to the trained environment, on table 3. the best fitness are always located on the diagonal. The last part of the paper will discuss about the best strategy for generating generic neurocontrollers.

4 GENERIC NEUROCONTROLLERS

4.1 Random Selection of the Environment

The first idea for building generic neurocontrollers consists in mixing the environments during the evolution. A new simulation has been performed, but the environment is now randomly selected for each generation. After 1500 generations, the synaptic weights never converge to a stable value. Figure 10 shows the evolution of the fitness during the 200 first generations (no changes were observed after). Compared to Figure 8 this evolution cannot be considered as satisfying. Results have shown that random selection of the environment makes the global system non-deterministic and prevents the genetic algorithm from finding the optimal solution. To avoid this problem, each individual is now trained in the eight environments.

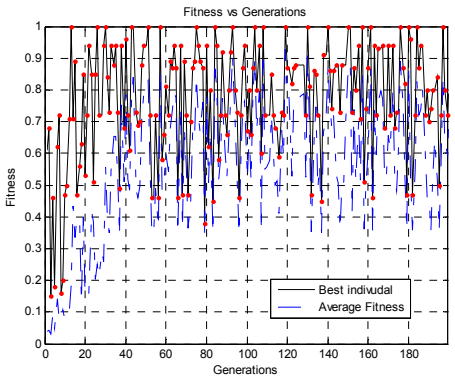


Figure 10: Fitness versus generations.

Table 3: Performance of the best individual of the final generation in the seven other environments.

Trained environment \ Explored environment	a.	b.	c.	d.	e.	f.	g.	h.
a.	96 %	70 %	6 %	9 %	12 %	10 %	1 %	1 %
b.	94 %	98 %	4 %	11 %	14 %	11 %	2 %	2 %
c.	14 %	13 %	93 %	92 %	33 %	20 %	15 %	15 %
d.	**38 %**	62 %	9 %	100 %	86 %	81 %	11 %	11 %
e.	7 %	7 %	7 %	100 %	86 %	5 %	4 %	4 %
f.	41 %	65 %	8 %	9 %	9 %	85 %	14 %	14 %
g.	38 %	62 %	17 %	23 %	84 %	77 %	77 %	72 %
h.	44 %	64 %	90 %	93 %	84 %	83 %	61 %	76 %

4.2 Evolution in the Eight Environments

For the reason explained in the previous section, each individual is now successively trained in the eight environments. The fitness is similar to the previously described equation 4 (equivalent to the average performance in the eight environments). The experiment lasts about fifteen hours (eight times longer than for one environment). The evolution of the fitness is now asymptotic (similar to figure 8) that proves the convergence of the genetic algorithm. Examples of trajectories are shown on figure 11.

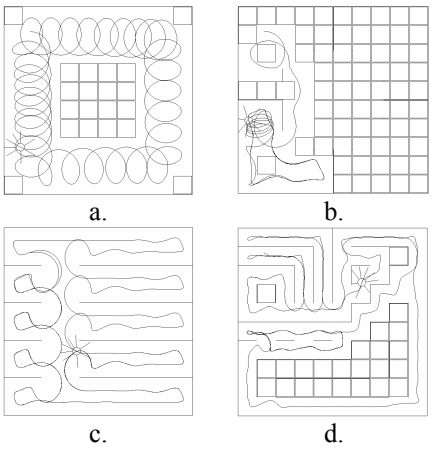


Figure 11: Example of trajectories in four environments.

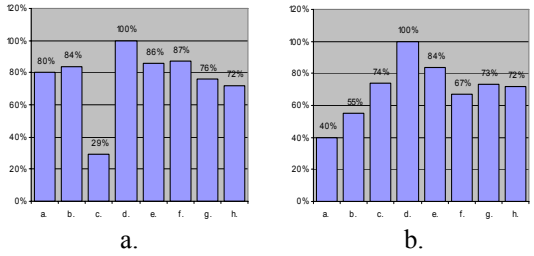


Figure 12: Performance of the best individual (last generation) in the eight environments.

Figure 12.a. shows the fitness in each environment. The global performance is satisfying (76.75%) to the detriment of the environment c. (only 29%). Figure 11.b. shows the trajectory of the robot in this environment; the robot is quickly jammed in a dead end. However it's hard to conclude about the adaptability of the neurocontroller. Indeed, it is clear that the performance in the environment c. has been sacrificed in favour of global fitness. Considering that generic means able to perform a high fitness in any situations, this goal isn't reach.

4.3 Increasing the Last

To avoid having a "sacrificed" environment, we performed the previous experiment with a new fitness function. The average isn't longer considered. Each individual is evaluated in the eight environments and the new fitness is the performance in the worst environment. For example, on figure 12.a, the performance used to compute crossovers is the weakest: 29% (environment c.).

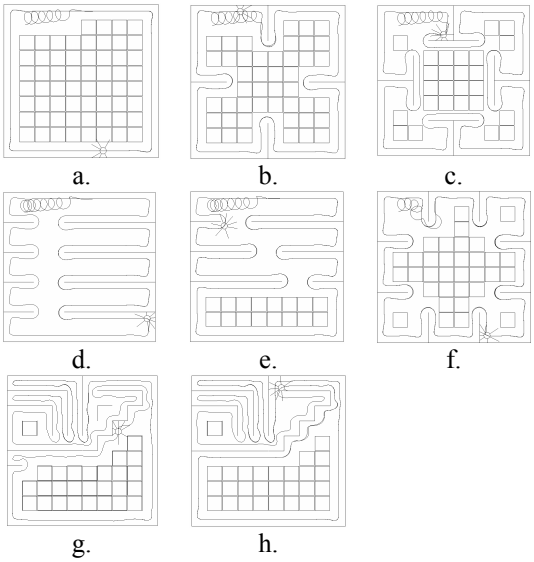


Figure 13: Trajectory of the robot in each environment.

Figure 12.b shows the performance in each environment of the best individual (last generation). The worst performance is 40% in the environment a. (slightly better than the previous 29%). The trajectories of the robot are clearly based on a wall following strategy visible on figure 13. As the environment a. has no wall (except the outline walls) the performance is poor. In spite of this, this result is encouraging. The average performance is 70.62%, not so far from the previous 76.75%. This means that the global performance isn't too much affected. Even if these results are globally worst, it stays encouraging. Probably that the chosen neurocontroller (without hidden layer) does not allow the robot to perform a high performance in the eight environments at the same time. These results tend to show that taking the performance of the worst case may provide more generic controllers than averaging the fitness

5 CONCLUSIONS

We presented in this paper some experiments based on the grasping task (for example an autonomous vacuum robot). These experiments are based on the genetic evolution of a neurocontroller without hidden layer. In the first part we evolved in simulation eight neurocontrollers (each in a given environment). The neurocontroller were swapped and the performances in the other environments were evaluated. For some researchers, there is sometime a mix-up between the genetic algorithm and the generated behavior. We've shown that genetics algorithms can be easily adapted with the same parameters to several problems. We've also shown that the generated neurocontroller is dedicated to the trained environment. It means that genetics algorithms are generic, contrary to neurocontrollers that are dedicated. This result can probably be extended to all the parameters of the evaluation: noise, robot's hardware, battery charge, etc.

In the second part of the paper, several strategies were experimented to produce generic neurocontrollers. First, the evaluation of the individual was done in the eight environments and the average performance was used for the fitness computation. This experiment provides good results except in one environment where the fitness was very poor. In the final experiment the performance in the worst environment was used to compute the fitness. The global performance is slightly smaller than in the previous experiment, but the performance is more distributed in the environments. Generating generic controllers using genetic algorithms stay a complex problem but we've shown that taking the worst case for evaluating the individual may be a first step in the automatic generation of generic neurocontrollers.

REFERENCES

Jansen T. and Wegener I., 2002, *On the analysis of evolutionary algorithms: A proof that crossover really can help.* Algorithmica, Springer New York, Volume 34, Number 1 / July, 47-46.

Bäck T., 1996, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press.

Floreano D. and Mondada F., 1994, *Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural Network Driven Robot*. 3rd International Conference on Simulation of Adaptive Behavior (SAB'94).

Mondada F., Franzi E. and Ienne, P., 1993, *Mobile Robot Miniaturization: A Tool for Investigation in Control Algorithms*, 3rd International Symposium on Experimental Robotics III, October 28-30, 501-513.

Floreano D. and Mondada F., 1998, *Evolutionary Neurocontrollers for Autonomous Mobile Robots*. Neural Networks, 11(7-8), 1461-1478.

Nolfi S. and Floreano D., 2000, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-organizing Machines.* Bradford book, MIT Press, Cambridge, Massachusetts.

Ahuactzin J.M., Talbi E-G, Bessiere P. and Mazer E.,1992 , *Using genetic algorithms for robot motion planning*. European Conference on Artificial Intelligence (ECAI92), Wien (Austria).Yamasaki F., Endo K., Kitano H. and Asada M., 2002, *Acquisition of humanoid walking motion using genetic algorithm - Considering characteristics of servo modules*. Robotics and Automation, Proceedings. ICRA '02. IEEE International Conference on Robotics and Automation, 3123-3128.

Arkin R.C., 1992, *Cooperation without Communication, Multiagent Schema-Based Robot Navigation*, Journal of Robotic Systems, Vol. 9 (3), avril , 351-364.

Lucidarme P. and Simonin O.,2006 , *Le robot mobile Type 1*, Journée des démonstrateurs en automatique, Angers, France.

Haussler A., Li Y., Ng K.C., Murray-Smith D.-J. and Sharman, K.C., 1995, *Neurocontrollers designed by a genetic algorithm*. First conference on genetic algorithms in engineering systems: innovations and applications, Sheffield, UK, Publ. No. 414.

Braitenberg V., 1986, Vehicules – *experiments in synthetic psychology*, Bradford books.

Mitchell T.M., 1997, *Machine learning*. McGraw-Hill Science.

COMBINATION OF BREEDING SWARM OPTIMIZATION AND BACKPROPAGATION ALGORITHM FOR TRAINING RECURRENT FUZZY NEURAL NETWORK

Soheil Bahrampour[1], Sahand Ghorbanpour[2] and Amin Ramezani[1]
[1]*Control and Intelligent Processing Center of Excellence, School of ECE, University of Tehran, Iran*
{s.bahrampour, a.ramezani}@ece.ut.ac.ir
[2]*Douran Company, Tehran, Iran*
sahand.ghorbanpour@bihe.org

Keywords: Recurrent fuzzy neural network, identification, breeding swarm optimization.

Abstract: The usage of recurrent fuzzy neural network has been increased recently. These networks can approximate the behaviour of the dynamical systems because of their feedback structure. The Backpropagation of error has usually been used for training this network. In this paper, a novel approach for learning the parameters of RFNN is proposed using combination of the backpropagation and breeding particle swarm optimization. A comparison of this approach with previous methods is also made to demonstrate the effectiveness of this algorithm. Particle swarm is a derivative free, globally optimizing approach that makes the training of the network easier. These can solve the problems of gradient based method, which are instability, local minima and complexity of differentiating.

1 INTRODUCTION

Fuzzy neural network (FNN) was introduced to fuse fuzzy systems and neural networks into an integrated system to reap the benefits of both (Ku, 1995). The major drawback of the FNN is its limited application domain to static problems, due to the feedforward network structure, thus it is inefficient in dealing with temporal applications.

Recurrent neural network systems learn and memorize information implicitly with weights embedded in them. A recurrent fuzzy neural network (RFNN) was proposed based on supervised learning, which is a dynamic mapping network and it is more suitable for describing dynamic systems than the FNN (Lee, 2000). Of particular interest is that it can deal with time-varying input or output through its own natural temporal operation (Williams, 1989). Ability of temporarily storing information simplifies the network structure and fewer nodes are required for system identification. Because of the complexity in back propagation (BP) learning approach, only diagonal fuzzy rules have been implemented (Ku, 1995). This limiting feature restricts users to employ a more completed fuzzy rule base.

In this paper a novel approach is proposed as a solution to this problem. We combined original BP used in previous works (Lee, 2000) with a breeding particle swarm optimization (BPSO) to train the network more easily and without the complexity of differentiating. The BPSO approach is an derivative-free, global optimizing algorithm that is a combination of genetic algorithm (GA) (Surmann, 2001) and particle swarm optimization (PSO) (Engelbrecht, 2002, Angeline 1994, and Kennedy, 1995) which was first used for training RNN (Settles, 2005).

This paper is organized as follows. In section 2, the RFNN structure is introduced and a comparison between the FNN and the RFNN is described. Section 3 briefly introduces BPSO. The training architecture of the network is presented in section 4 and simulation results are discussed in section 5. Finally, in section 6 we summarize the result of this approach.

2 NETWORK STRUCTURE

The key aspects of the RFNN are dynamic mapping capability, temporal information storage, universal approximation, and the fuzzy inference system. The

RFNN possesses the same advantages as recurrent neural networks and extend the application domain of the FNN to temporal problems. A schematic diagram of the proposed RFNN structure is shown in Fig. 1 which indicates the signal propagation and the operation functions of the nodes in each layer. In the following description, u_i^k denotes *i*-th input of a node in the *k*-th layer; o_i^k denotes the *i*-th node output in the *k*-th layer. For the sake of brevity, a brief description of the RFNN is introduced. Interested readers are referred to reference (Lee, 2005).

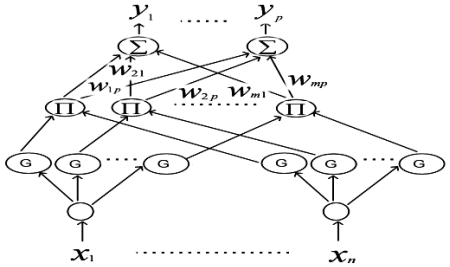


Figure 1: structure of RFNN.

Layer 1: Input Layer:

$$o_i^1 = u_i^1 \tag{1}$$

Layer 2: Membership Layer: The Gaussian function is adopted here as a membership function:

$$o_{ij}^2 = \exp\left\{-\frac{(u_{ij}^2 - m_{ij})^2}{(\sigma_{ij})^2}\right\} \tag{2}$$

where m_{ij} and σ_{ij} are the center (or mean) and the width (or standard deviation—STD) of the Gaussian membership function. The subscript *ij* indicates the *j*-th term of the *i*-th input. In addition, the inputs of this layer for discrete time *k* can be denoted by

$$u_{ij}^2(k) = o_i^1(k) + o_{ij}^f(k) \tag{3}$$

where $o_{ij}^f(k) = o_{ij}^2(k-1) \times \theta_{ij}$ and θ_{ij} denotes the link weight of the feedback unit. It is clear that the input of this layer contains the memory terms $o_{ij}^2(k-1)$, which store the past information of the network. Each node in this layer has three adjustable parameters: m_{ij}, σ_{ij}, and θ_{ij}.

Layer 3: Rule Layer:

$$o_i^3 = \prod_i u_i^3 = \exp\left\{-\left[D_i(u_i^2 - m_i)\right]^T \left[D_i(u_i^2 - m_i)\right]\right\} \tag{4}$$

Where

$$D_i = diag\left\{\frac{1}{\sigma_{1i}}, \frac{1}{\sigma_{2i}}, ..., \frac{1}{\sigma_{ni}}\right\}, \quad u_i = [u_{1i}, u_{2i}, ..., u_{ni}]^T, m_i = [m_{1i}, m_{2i}, ..., m_{ni}]^T \tag{5}$$

Layer 4: Output Layer:

$$y_p = o_p^4 = \sum_{j=1}^{m} u_j^4 w_{jp}^4 \tag{6}$$

where $u_j^4 = o_j^3$ and w_{ji}^4 (the link weight) is the output action strength of the *i*-th output associated with the *j*-th rule. w_{ji}^4 are the tuning factors of this layer. Finally, the overall representation of input x and the *p*-th output is

$$y_p(k) = o_p^4 = \sum_{j=1}^{m} w_{jp}^4 \times \prod_{i=1}^{n} \exp\left[-\frac{(x_i(k) + o_{ij}^2(k-1) \times \theta_{ij} - m_{ij})^2}{(\sigma_{ij})^2}\right] \tag{7}$$

Where

$$o_{ij}^2(k-1) = \exp\left[-\frac{(x_i(k-1) + o_{ij}^2(k-2) \times \theta_{ij} - m_{ij})^2}{(\sigma_{ij})^2}\right] \tag{8}$$

Obviously, using the RFNN, the same inputs at different times yield different outputs. The proposed RFNN can be shown to be a universal uniform approximator for continuous functions over compact sets if it satisfies a certain condition (Lee, 2000).

3 BPSO

With correct combination of GA and PSO, the hybrid can outperform, or perform as well as, both the standard PSO and GA models (Settles, 2005). The hybrid algorithm combines the standard velocity and position update rules of PSOs with the ideas of selection, crossover and mutation from GAs. An additional parameter, the breeding ratio (Ψ), determines the proportion of the population which undergoes breeding (selection, crossover and mutation) in the current generation. Values for the breeding ratio parameter range from (0.0:1.0).

In each generation, after the fitness values of all the individuals in the same population are calculated, the bottom (N · Ψ) are discarded and removed from the population where N is the population size. The remaining individual's velocity vectors are updated, acquiring new information from

the population. The next generation is then created by updating the position vectors of these individuals to fill (N · (1 − Ψ)) individuals in the next generation. The (N · Ψ) individuals needed to fill the population are selected from the individuals whose velocity is updated to undergo VPAC crossover and mutation and the process is repeated. For clarity, the flow of these operations is illustrated in Figure 1 where k = (N · (1 − Ψ)).

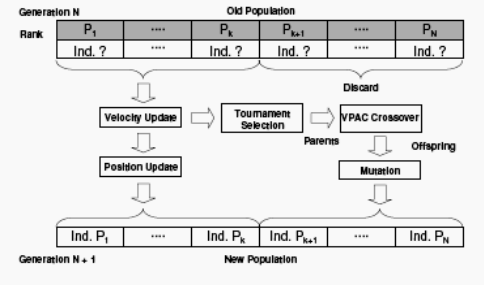


Figure 2: BPSO.

Here, we developed crossover operator to utilize information available in the Breeding Swarm algorithm, but not available in the standard GA implementation. The new crossover operator, velocity propelled averaged crossover (VPAC), incorporates the PSO velocity vector. The goal is creating two new child particles whose position is between the parent's positions, but accelerated away from the parent's current direction (negative velocity) in order to increase diversity in the population. Equations (8) show how the new child position vectors are calculated using VPAC.

$$\begin{cases} c_1(x_i) = \dfrac{p_1(x_i) + p_2(x_i)}{2.0} - \varphi_1 p_1(v_i) \\ c_2(x_i) = \dfrac{p_1(x_i) + p_2(x_i)}{2.0} - \varphi_2 p_2(v_i) \end{cases} \quad (9)$$

In these equations, $c_1(x_i)$ and $c_2(x_i)$ are the positions of child 1 and 2 in dimension *i*, respectively. $p_1(x_i)$ and $p_2(x_i)$ are the positions of parents 1 and 2 in dimension *i*, respectively. $p_1(v_i)$ and $p_2(v_i)$ are the velocities of parents 1 and 2 in dimension *i*, respectively. φ is a uniform random variable in the range [0.0:1.0]. Towards the end of a typical PSO run, the population tends to be highly concentrated in a small portion of the search space, effectively reducing the search space. With the addition of the VPAC crossover operator, a portion of the population is always pushed away from the group, increasing the diversity of the population and the effective search space.

The child particles retain their parents's velocity vector $c_1(\vec{v}) = p_1(\vec{v}),\ c_2(\vec{v}) = p_2(\vec{v})$. The previous best vector is set to the new position vector, restarting the child's memory by replacing new $c_1(\vec{p}) = p_1(\vec{x}),\ c_2(\vec{p}) = p_2(\vec{x})$. The velocity and position update rules remain unchanged from the standard inertial implementation of the PSO. The social parameters are set to 2.0 while inertia is linearly decreased from 0.7 to 0.4 and a maximum velocity (Vmax) of ±1 was allowed. The breeding ratio was set to an arbitrary 0.3. Tournament selection, with a tournament size of 2, is used to select individuals as parents for crossover. The used mutation operator is Gaussian mutation, with mean 0.0 and variance reduced linearly in each generation from 1.0 to 0.0. Each weight in the chromosome has probability of mutation 0.1.

4 NETWORK TRAINING

BP approach, as mentioned, has been mostly used for training RFNN in previous works. This approach is not easy to implement, when faced with the case of a complete or a non-diagonal fuzzy rule base. As we can see in Fig. 1, each rule of layer 3 is made by only a diagonal variables, i.e. the *i-th* rule are made by multiplication of the *i-th* outputs of layer 2. However, if we want to use complete or non-diagonal fuzzy rule base, it will make learning of parameters in layer 2 totally complicated. In this paper we propose Breeding Particle Swarm Optimization for tuning parameters of layer 2 (m_{ij} , σ_{ij} , θ_{ij}) and original BP for tuning w_{jp} . These two approaches are used simultaneously. Pseudo code of the algorithm used in this study for training RFNN parameters is shown in Fig. 3. The proposed combination has various benefits for training RFNN. First of all, there is no need to differentiate those complex derivations for training the parameters of the 2nd layer. The proposed algorithm utilizes BPSO as a derivative-free approach for training these parameters. The method is also a global optimization approach that prevents training parameters from converging to local minima. Because of simplicity and high speed convergence, the parameters of 4th layer is learned by BP. Note that using a complete fuzzy rule base doesn't affect the tuning of w_{jp} by BP and will not increase its complexity.

Initialize
While $error > \xi$
    apply BPSO for tuning m_{ij}, σ_{ij}, and θ_{ij}
    apply BP for tuning w_{jp}

Figure 3: Pseudo code of the proposed tuning algorithm.

5 SIMULATION RESULTS

Suppose the following nonlinear dynamical system:

$$y_p(k+1) = f(y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)) \quad (10)$$

Where,

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2} \quad (11)$$

In this system the output value depended on the previous values of the output and the previous values of the input. We use RFNN to identify this system. Because of dynamical characteristics of RFNN, it is not necessary to use all complete samples of the previous inputs and outputs. So just y(k) and u(k) are used for estimating y(k+1). The parameters of the 2nd layer of the RFNN is tuned by BPSO and the output weights, w_{jp} is tuned with BP simultaneous. The same input signal that was used in (Lee, 2000) is used here for testing. Fig. 4 illustrates that learning of the network is successfully done. This method leads to better identification than the previous ones. The MSE parameter was 0.00013 in original method while our proposed method converges to 0.00005.

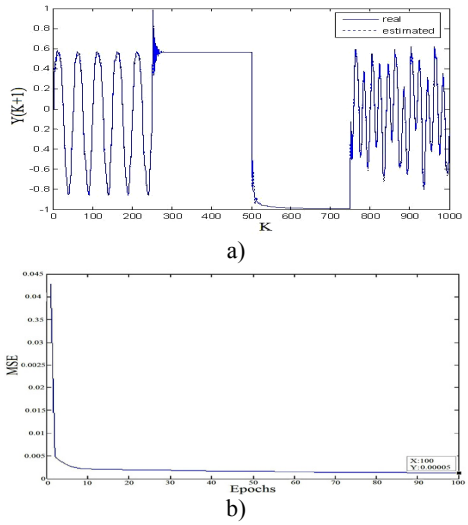


Figure 4: training RFNN parameters with cooperation of the BPSO and BP. a) Identification. b) MSE.

6 CONCLUSIONS

In this study a novel approach for training RFNN was proposed. BP algorithm suffers from complexity of differentiating and converging to local minima. Our proposed method utilizes BPSO with combination of BP. As the simulation results show, applying this algorithm improves the performance of training RFNN. This improvement is gained by globally optimizing the feature of PSO that prevents training to be entrapped in local minima. The complexity of differentiating for gradient based methods is more serious when a complete or a non-diagonal fuzzy rule base is used. This algorithm solves this problem too and one can use it more frequently.

REFERENCES

Angeline, P., Saunders, G., Pollack, J.,1994. An evolutionary algorithm that constructs recurrent neural networks.

Engelbrecht, A. P., 2002. Computational Intelligence. *John Wiley and Sons,2002.*

Kennedy, J., Eberhart, R., 1995. Particle Swarm Optimization. *IEEE International Conference on Neural Networks , pp. 1942-1948.*

Ku, C. C., Lee, K. Y., 1995. Diagonal recurrent neural networks for dynamic systems control. *IEEE Trans. Neural Networks, vol. 6, pp. 144–156.*

Lee, C. H., Teng, C., 2000. Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks. In *IEEE Transactions on fuzzy systems, vol.8, NO. 4, 349,366* .

Settles, M., Nathan, P., Soule, T, 2005. Breeding Swarms: A New Approach to Recurrent Neural Network Training. *GECCO'05, June 25–29 Washington, DC, USA.*

Soule, T., Chen, Y., Wells, R.,2002. . Evolving a strongly recurrent neural network to simulate biological eurons. *In the proceedings of The 28th Annual Conference of the IEEE Industrial Electronics Society.*

Surmann, H., Maniadakis, M., 2001. Learning feed-forward and recurrent fuzzy systems: A genetic approach. *Journal of System Architecture 47, 649-662.*

Williams, R. J., Zipser, D.,1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation., vol. 1, pp. 270–280.*

ONTOLOGY ADAPTER
Network Management System Interface Model

Lingli Meng, Lusheng Yan
Network Management Research Center, Beijing University of Posts and Telecommunications
Beijing, Popular Republic of China
mickey200002@sina.com, lsyan@bupt.edu.cn

Wenjing Li
Department of Computer science and technology, Beijing University of Posts and Telecommunications
Beijing, Popular Republic of China
wjli@bupt.edu.cn

Keywords: Ontology Adapter, Interface Model.

Abstract: This paper proposes a new method to define the interface model of network management system, that is ontology adapter. This model includes three parts, which are ontology agent, ontology knowledge base and ontology resource description. We can realize the uniform presentation of different network resource interface information using them. Therefore, we can take this model as a common data platform to offer the interface information to the network management system.

1 INTRODUCTION

The telecommunication network and business have such a great speed development, which cause more network resource needed to be managed in the managed network system (Luoming,2001). However, the traditional network management interface model of the TMN proposed by the ITU-T is not suitable for changes of the network technology, and can only satisfy the static management of the managed network resources, adjusting badly to all kinds of the existing interface technology. Therefore, the interface model is encountering the great challenges (Luoming ,2003)(Zhipeng,2007):

1) variety of managed network resources. For the complicated managed network resources are included under the background of network management integration, it is an important problem how to describe these managed resources in the interface layer, and give the uniform description of them before offering to the NMS(Network Management System).

2) diversity of interface description. At present, there exactly exists the diversity definition of private interfaces among all equipment vendors, although the ITU-T has standardized these interface descriptions. It can cause the inconsistence presentation of the data, and not share the managed resources with different systems, as well as be deficient in flexibility when NMS collects the information from different interface equipments. Therefore, how to find a better interface model to integrate all these different descriptions of network resources, realize the uniform data presentation, which will be a creative breakout of interface model in NMS.

So as to solve these problems above, we proposal an interface adapter model using ontology language. The rest of this paper is organized as follows: Section 2 introduces the existing method of interface definition in NMS. In section 3, we give a glimpse of ontology language, presenting the ontology language OWL. We describe the detail of the ontology interface adapter model in section 4. Finally section 5 would conclude our paper and give the future work.

2 INTERFACE ANALYSIS

ITU-T proposed five layers architecture of TMN in 1990's. The five layers are composed of business management, service management, network

management, network element management and network element, which are managed by NMS.
At present, it has been defined that several methods of the interface between NMS and managed resources, Such as SNMP, CORBA and XML. They design the interface model distinguishingly according to different prospective.

2.1 SNMP

SNMP is a standard protocol to manage the IP network. It can monitor and control the variable set,as well as monitor the two data format of equipments, SMI and MIB. SNMP adopts the model of "Management Process---Agent Process". When the Management Process sends an order to the Managed Object, the Managed Object receives the order by the Agent Process. Then the Agent process also sends back the response to the Management Process. That is, the communication between Management Process and Managed Object is not in direct, but using the Agent Process.

SMI is a method of defining the management object using SNMP. It based on object-oriented, not using encapsulation, inheritance, polymorphism. So it simplifies the management information models. It is on behalf of an instance of a class by using table to represent the data type, table, column, and row. (Jorge Vergara, 2003)

2.2 Corba

CORBA is proposed by the OMG. It is an object-oriented software development as well as application platform, with the function of portable, reusable and connection, under distributed heterogeneous circumstance.

IDL, which ITU-T use to define the TMN and UTRAD. It is an object-oriented description language. By defining of the IDL, classes are interfaces between different models. They have attributes and methods. IDL can be mapped to different languages, such as C++, JAVA, Ada, Cobol and SmallTalk.

2.3 XML

XML is given by the W3C. It is an information processing tool, which has on relevant to platform, software and hardware. XML has the feature of flexibility, easy-realization, easy-extension, as well as low cost. At present, we only use this tool to define format of the configuration files and the performance files in NMS.

3 ONTOLOGY DESCRIPTION

Ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of existence. In 1990's, ontology was brought into the field of the knowledge representation, knowledge sharing, semantic reason and so on.
In recent years, it has been great development of the applications and research of ontology, which NMS has been focusing on. There may exist the possibility of the application of ontology in network management information model (Jorge, Vergara, 2003), the relations among ontology mapping, network management operation (Wong, 2005), as well as special management information model based on ontology. However, it is still short of the research to design interface adapter model using ontology language in NMS. We will propose an ontology interface adapter model next section.

3.1 OWL

Generally speaking, we use a special ontology language to describe ontology. At present, the main ontology description languages are DAML, OIL, KIF, and OWL, among which, the OWL proposed by the W3C has the great effect.

The OWL (Web Ontology Language) is designed for processing the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than those supported by XML, RDF, and RDF Schema (RDF-S), by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

OWL has been applied widely to module in most kinds of domain ontology: W3C has used OWL in the ontology model of Web Service; Using OWL defined the analysis model, which can be adapted to test circumstance (Bodkin, 2005).

4 ONTOLOGY ADAPTER DESIGN

4.1 Ontology Adapter Model Analysis

We propose and discuss the ontology adapter model in the interface definition below. We divide ontology adapter model into three parts: ontology

agent, ontology knowledgebase and ontology resource description. Figure 1 shows the model.

Now we will introduce what functions the three parts have, and how to realize the uniform description of the interface information when NE management Layer sends the network resource to the NMS. First of all, the network resource information has been defined well in the NE Management Layer.

Step1, ontology agent receives resource information, represented by the SMI, IDL or XML from the NE Management Layer. Then they will be transformed to the uniform representation using OWL. If just the

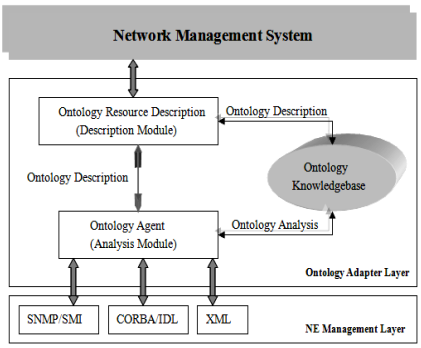


Figure 1: Ontology Adapter Model.

information describe static resources, which can be dealt with by the ontology agent; otherwise including dynamic information, which would be forwarded to the Ontology Knowledgebase. It will give a further deal.

Step2, ontology knowledgebase is used to transform the complex interface information. There exists a database, which stores more transduction rules between OWL and other interface description languages, all that the ontology agent can't deal with, will be sent to the ontology knowledgebase.

Step3, ontology resource description, in which, we can complete the interface information representation finally. After receiving the static information sent by ontology agent and the dynamic ones dealt by ontology knowledgebase, ontology resource description module will integrated both of them, and present a uniform format to NMS.

4.2 Ontology Adapter based on XML

This section we give a simple realization of ontology adapter model. We have defined some transformation rules and a simple ontology resource description frame using OWL.

4.2.1 Ontology Agent

Figure2 shows the process how ontology receives the interface information. Firstly, read interface information; Secondly, analyze language type and judge whether it present dynamic information; thirdly, transform the format, finally, the final OWL representation is formed.

In transform mode 1, take IDL as an example, there are some rules below:

Rule 1: $\forall$ module $\in$ IDL, $\rightarrow$
xmls: $\exists$ NameSpace $\in$ OWL

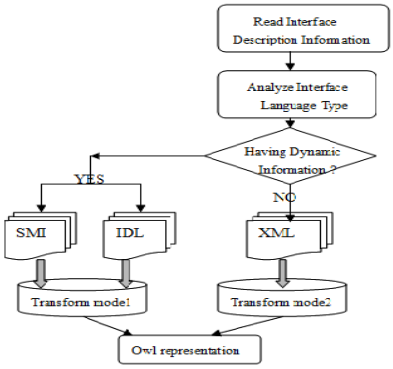


Figure 2: Ontology Agent Flow Chart.

Case study:
module TypeDemo { …… } $\rightarrow$
xmlns="http://www.w3.org/TR/2004/REC-owl-guide-20040210/TypeDemo#"
xml:base="http://www.w3.org/TR/2004/REC-owl-guide-20040210/TypeDemo#"

Rule 2: $\forall$ Interface $\in$ IDL, $\rightarrow$ $\forall$ class $\in$ OWL

Case study:
interface Father { ……. }; $\rightarrow$
<owl:Class rdf:ID="Father"/>

Rule3: $\forall$ Attribute $\in$ IDL, $\rightarrow$
$\forall$ property $\in$ OWL

Case study:
attribute string name; $\rightarrow$
<owl:DatatypeProperty rdf:ID="name">
<rdfs:domain rdf:resource="#Father"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

Role4: $\exists$ fundamental datatype $\in$ IDL, $\rightarrow$
xsd: $\exists$ datatype $\in$ OWL

Case study: String $\rightarrow$ xsd:string

Note: there are some data types that can't be mapped directly, therefore, you must do some changes. Such as, char $\rightarrow$ xsd:byte.

Rules above, which just aim at static information. There are still methods to represent dynamic

information, such as operation and context. We do such analysis in ontology knowledgebase. Simply, we use OWL text to descript these methods. Here, we only offer a sample:

e.g. Method1 (var1, var2) ∈ IDL →
Method1 has var1, var2 ∈ OWL.

In transform mode 2, we translate XML into OWL, we should distinguish the differences between the two kinds of data types, the fundamental data type and the user-defined ones. Because the user-defined data type is so complex that we must avoid the semantic loss at our best, while transforming such information.

4.2.2 Ontology Resource Description

In this part, we represent a uniform network resource description framework using OWL. When ontology agent and ontology knowledgebase send the transformed information to ontology resource description, it will integrate these two parts, and give a uniform representation using OWL. We now give a sample description, as follows:

The definition of class:

```
<owl:Class rdf:ID=Equipment>
  <rdfs:subClassOf rdf:resource=PhysicalME/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasDatatypeProperty rdf:resource=
      equipType>
        <owl:ConstraintRequirement
        rdf:datatype="&xsd;string">
        </owl:ConstraintRequirement>
      </owl:hasDatatypeProperty>
      <owl:hasObjectProperty
      rdf:resource=hasInterface/>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=
      "#hasInterface"/>
      <owl: minCardinality
      rdf:datatype="&xsd;int"> 1
      </owl: minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=
      "# hasInterface "/>
      <owl:haslValuesFrom rdf:resource=
      "# Interface "/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

The detail definition of attributes for the above:

```
<owl:DatatypeProperty rdf:ID=" equipType ">
      <rdf:type rdf:resource=
      "&owl;FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Equipment"/>
      <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="hasInterface">
      <rdfs:domain rdf:resource="#Equipment"/>
      <rdfs:range rdf:resource="#Interface"/>
      <owl:inverseOf
      rdf:resource="#belongEquipment "/>
</owl:ObjectProperty>
```

5 CONCLUSIONS

It is a challenging research of the integration of network resource interface information. The ontology adapter model we presented has solved how to integrate these different interface informations in architecture. Our future work will concentrate on the research of offering a specific method and realization, using which we can success to transform existing interface presentation language into OWL. If possible, a tool which can be used to produce this transformation automatically is expected.

REFERENCES

Luoming, Meng. Qi feng. *Modern Network Management Technology* ,M. Beijing: Press of BUPT, 2001.

Luoming, Meng. Network Management: Problems, Progress and Prospect, J.*Journal of BUPT*, 26-2: 1-8, 2003.

Zhipeng,Gao. Ontology-based Modeling Methods,Model and Application of Shared Management Information, D. Beijing: *Doctoral Dissertation of BUPT*, LAB. of BUPT,2007.

Jorge.E López de Vergara, Víctor A. Villagrá,Juan I. and Julio Berrocal. Ontologies: giving semantics to network management models, J. *IEEE Network*, 17-3:15-21, 2003

Wong, A.K.Y. Ray, P. Parameswaran, N. Strassner, J. Ontology mapping for the interoperability problem in network management, J. *Selected Areas in Communications, IEEE Journa*l on. 2005, Volume: 23 Issue: 10: 2058 – 2068.

Bodkin, M., Harris, M., Helton, A. Use of ontologies for decision support generation and analysis,A. *Autotestcon, 2005. IEEE.* 2005:684

HUNTER – HYBRID UNIFIED TRACKING ENVIRONMENT

Real-time Identification and Tracking System using RFID Technology

A. G. Foina and F. J. Ramirez Fernandez
Escola Politécnica - Universidade de São Paulo, São Paulo, SP, Brazil
aislan.foina@poli.usp.br, jramirez@lme.usp.br

Keywords: RFID, Middleware, Tracking, Real-time, Supervision, Wi-Fi, Wireless.

Abstract: This article presents a developed system that use RFID technology for trucks' cargo real-time tracking. RFID tags were settled at trucks' dump-carts and readers were spread throughout warehouses entrances, at the truck weighting scale and through unload platforms. The unload inspectors used robust PDA with camera, along with Wi-Fi access points installed in warehouses, to confirm the truck information and take a snapshot for future audits. A wireless broadband link was used to connect two weighting scale that are distant from the unloading area. All technologies communicate with a web-based middleware that manages all different devices. The system design is flexible enough to be used in very different applications like product process control, automated manufactory lines control, supply chain applications and others.

1 INTRODUCTION

The unloading of bulk cargo is a great security problem for companies that work with this product modality, since the unloading process consists in three tasks:

1. Full weight, which the truck is weighed with full loaded dump-cart;
2. Unload, which the truck goes to a designated site and unloads the product;
3. Empty weight, which the truck is once again weighed unloaded.

The amount unloaded by the truck is the difference between the second and first weight. Thus, if a load theft occurs in the period between the two weights, the company will not be able to know if the theft had occurred or even how much was stolen.

In some ports, the exporting companies' area is open to access by anyone or any vehicle, and the weight sites are approximately 2 kilometers far. This scenario makes the cargo theft possible by the following form: the truck is weighed fully loaded, and then it goes to an area outside the unloading zone; after that, it unloads the cargo at another truck, or exchange dump-carts, and then return to be weighed empty.

The estimated loss with cargo in bulk theft, at the studied company, is approximately 7% of its annual profit, which represents 750 thousand dollars. Based on that scenario, it was projected a Radio Frequency Identification (RFID) based solution to track the vehicles and works as an electronic seal at the trucks' dump-cart. So, the system control the sites where the truck is passing by, time from one point to another, and electronically detect a dump-cart seal violation.

All technologies used will be described to follow, including the managing application.

2 ACTIVE RFID SYSTEM

Present RFID systems consist basically of four components: an electronic tag, a reader, an antenna, and application software to process the data. When a tag approaches the antenna, the latter sends a signal to the reader with the tag identification. The reader receives the signal, and the information is sent to a computer executing the software. This software is normally a middleware application that processes the data packets and sends them to an end-user application or a database (Kim, 2006).

RFID tags can be active or passive. The active ones are self fed by an intern battery and the passive ones are fed with the energy from electromagnetic waves sent by the reader. Regarding passive tags, the reading ranges vary between 5 cm and 10 m. For the active tags, since they have internal battery, the reading range can reach distances in the order of 200 meters. Active tags can be connected to temperature,

tamper and movement sensors, among others. For this reason, the active tags technology was chosen here for truck access control in warehouses and unloading area. In this case, the reader used operates in the UHF frequency range, at 433MHz to transmit data and at 915MHz to receive data from the tag, with 80 meters reading range.

To work as an electronic seal, a tamper sensor was attached to the tag. This sensor accuses "ok" status if its terminal were placed in short circuit and "violated" if the terminal were open circuit. This feature allowed the development of a tag model that could be fixed at a truck's dump-cart, seen in Figure 1. If occurs a tag removal attempt, the tamper sensor would detect and send a violation signal to the nearest reader.

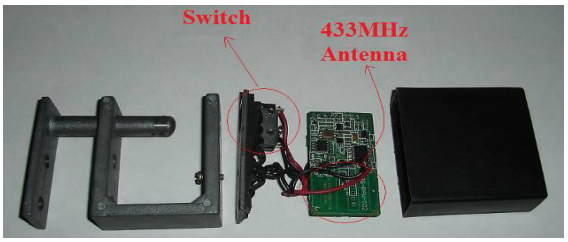


Figure 1: Active tag in a padlock shape with opening detector disassembled.

The main difficulty found in the tag design was towards the great difference between trucks used by the company, which made difficult to develop a single tag that could be fixed in each of them. To solve this problem, the tag was designed be similar to a padlock that can be fixed in any bar up to two inches thick, common in all kinds of truck. Therefore, it was possible to fix a tag in multiple parts of the truck. The violation detection of a tag is made throughout a small switch, that when the lock is closed, this switch is pressed and when the lock is opened, the switch is released. This switch was connected from two wires to the tag's circuit violation terminals sensor.

The adopted rule to install readers was: at each entrance and exit was installed a reader. For example, at the weighting scale platform were installed two readers, one at the entrance and another at the exit. At the unloading warehouses, one reader was put at each entrance/exit gate. All readers were connected by Unshielded Twisted Pair (UTP) cables Cat 5e. Warehouses and weighting scales were connected by multimode optical fiber cables due the long distance between readers' installation point and the server. The optical fiber and the UTP Cat 5a cables create a local area network between all devices and the server which executes managing application. So, each truck's passage through a reader generates a data packet to the server with information containing the tag's identification, electronic seal status, timestamp and the identification of the reader that received the packet.

3 DIRECTIVE ANTENNA

The place where the tag is attached on each truck may vary depending on the truck's dump-cart, and the material of it can vary between wood and steel as well. These variations can influence in the sensitivity of the tags, causing undesired tag readings outside the calibrated reader reading range. These readings happen when the truck is crossing in front of the warehouse entrance, causing unexpected alarms and making the system calls the security team unnecessary. To solve this problem, two resources were used: a directive antenna and the Received Signal Strength Indication (RSSI) feature of the reader (Foina, 2006).

This antenna will amplify the signal received in its directive lobule, sending a stronger signal to the reader in this case. But, if the signal is received outside its directive lobule, the reader will receive a weak signal (Balanis, 1997). The antenna used has an E-Plane and H-Plane beamwidth of 65 degrees at 3dB, circular polarization and 7 dBi of gain.

The RSSI feature of the active readers allows the measurement of the signal strength received by the reader. In the chosen equipment, the RSSI is an index from 0 to 255, where a higher value means a stronger signal.

With the RSSI and the directive antenna, the software can analyse the RSSI to check if the tag is behind or in front of the antenna. The directive antenna was installed behind the reader, facing to the warehouse door, approximately 1 meter away according to the Figure 2.

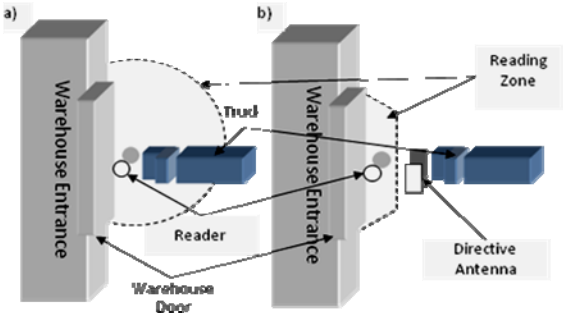


Figure 2: Aerial representation of the reading zone of both approaches, a) using the omnidirectional antenna and b) using the directive antenna.

When the tag is close to the door, but behind the antenna, the RSSI index will be very low because the tag is not in the main lobule of the antenna. When the tag is positioned between the door and the antenna, the RSSI index will be much higher than if measured in the same position with an omnidirectional antenna. This enables the middleware to make the decision of whether or not granting access into the warehouse.

4 MANAGING APPLICATION

The managing application developed, called Hybrid Unified Trekking Environment (Hunter), controls the unloading process information over the monitored vehicle. Every truck is treated by the system as an event queue to be executed. Inside the system, every truck's passage through an entrance/exit is an event. Full weight, unload and empty weight are examples of events to be executed by the truck. Thus, in the system, each truck has a state within a finite state's machine. Through received events by that truck, it registers exactly in which stage of unloading flow the truck is in real-time. In other words, the trucks in movement inside the area are treated as independent threads in the system, which is initiated when its tag passes by the weighting scale to capture the full weight, generating the first event. When these threads are created, a task queue at the data base is created as well, with all the tasks the truck must accomplish along the process, such as: weighting scale entrance for full weight, full weight capture, weighting scale exit, unloading area entrance, etc. While the truck moves and generates the events, these are interpreted by the system and converted into its respective tasks, and thus they are removed from the task queue and inserted in the journal table. In case the tasks orders are not obeyed, an alarm is generated on the system operator's screen.

The system graphical interface shows the number of trucks on each state of the unloading process, seen in Figure 3. The numbers of trucks in the lines between each state are showed as well. So, there are five states presented in the interface: Full weight capture, Unload line, Unloading process, Empty weight line and Empty weight capture. Through the graphical interface, it is possible to see the trucks that are in this state and how long they are there.

The application was totally developed in Java, being a part of it monolithic and a part in form of a web-base application. It is executed by a Tomcat application server and connected to an Oracle 9i data base with features of advanced task queue. As different technologies are used, the managing application was developed with modules, divided in three layers (Chen, 2003).

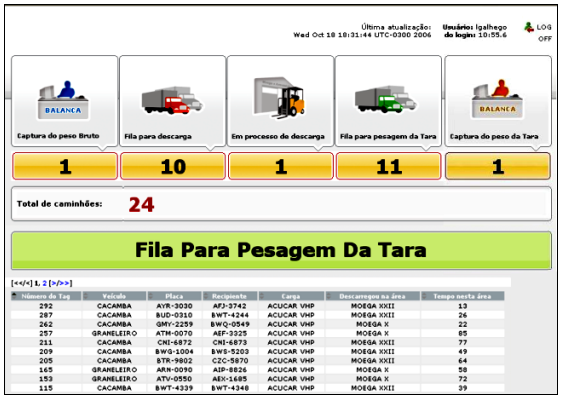


Figure 3: Hunter's screenshot showing the number of trucks in each state.

The first system layer, called Device Management Subsystem (DMS) is the layer responsible for the connection between the physical device, in this case RFID readers, and the system engine for process control, times and alarms, called here as Core. The DMS stays waiting for a package at the serial port or at the Ethernet connections, and when it receives a data sent by any device, it interprets the information and generates a message in Extensible Markup Language (XML) with information about the devices positioning, the name of the place where the tag was read and details about the truck which has read tag attached. to be sent to Core. DMS is a middleware, connecting different devices in the system at the same time, and allow new devices to be connected to the system without any change in the source code of the application.

The Core has a servlet that receives the XMLs from DMS and inserts them into the data base. When this insertion is made, the table where the information is stored has a trigger that initiates all the data analysis process, such as the alarms that can be generated due to this event or updates on truck's state in the unloading process. Connected to Core, there is the interface layer with two other modules, the graphical interface and the legacy integration systems. The first one is responsible for presenting information on the screen, such as alarms, navigation maps and reports. The second module is responsible for sending information to the company's legacy system, updating the warehouse stock, linking timing information and alarms generated to the company's Warehouse Management System (WMS).

All communication between system modules is made through XML messages and all system graphical interfaces are web-based, allowing its access from any computer without needing specific application installation, just a conventional web browser. The integration of the managing application with the company's corporative systems is made through recorded text files containing the XML messages in a shared directory due limitations of the company legacy system.

5 CONCLUSIONS

Despite the difficulties installing a wireless network and an optical fiber network in old warehouses, and placing sensible radio frequency equipment exposed to weather hazards, it was possible to install successfully all equipment and do their calibration. The server applications behaved correctly when were placed an amount of 200 trucks simultaneously circulating around the area and unloading the cargo.

Some unexpected problems happened and they were solved. The switch installed inside the padlock tag changed the inductance of the 433MHz antenna, so the internal organization of the tag was changed to keep the antennas as far as possible from the switch. Sometimes the truck parks in front of the reader, generating many readings of the tag, but sometimes the readings happed before the truck left the place. For this reason, filtering controls were implemented inside the DMS to avoid generation of fake alarms.

The system helped the company management and security. With the additional information generated by the system for the logistic department about the weight, unload time statistics, and the number of trucks in real time in each stage of the unloading process, it was possible to optimize the trucks' line, reducing the wait in line and therefore, reducing the average time of the unload process from 50 minutes to 30 minutes. The installation of RFID padlock tags in the trucks reduced in 60% the load theft in the port area and allowed the security team to find out points of vulnerability of the previous system and to detect the majority of the truck drivers corrupted by the theft group.

This same system showed flexible enough to be used with other RFID technology and applications, like passive tags to control automated manufacture lines, supply chain pallets and forklifts. In this case the application will control if the item pass through all the phases in manufactory line, from the beginning until the expedition. Can control who was the forklift operator and how long he took to move the products from one place to other. So the product will be tracked, supplying information about the storing and movement time, and the forklifts will be tracked as well, supplying information about the efficiency of the forklift operator and generating alarms if he does something wrong.

REFERENCES

Almanza-Ojeda, D. L.; Hernandez-Gutierrez, A.; Ibarra-Manzano, M. A.; 2006. Design and implementation of a vehicular access control using RFID. *In Electronics and Photonics, 2006. MEP 2006. Multiconference on 2006, pp. 223 – 225.*

Balanis, C. A.; 1997. *Antenna Theory: Analysis and Design*. John Wiley & Sons, USA, second edition, pp. 86-88.

Blythe, P.; 1999. RFID for road tolling, road-use pricing and vehicle access control. *In RFID Technology (Ref. No. 1999/123), IEE Colloquium on 25 Oct. 1999, pp. 8/1 – 816.*

Chen, S.; Gulati, S.; Hamid, S.; Huang, X.; Luo, L.; Morisseau-Leroy, N.; Powell, M. D.; Zhan, C.; Zhang, C; 2003. A three-tier System Architecture Design and Development for Hurricane Occurrence Simulation. *In Information Technology: Research and Education, 2003. Proceedings International Conference, pp. 113 – 117, 11-13 Aug. 2003.*

Foster, P.R.; Burberry, R.A.; 1999. Antenna problems in RFID systems. *In RFID Technology (Ref. No. 1999/123), IEE Colloquium on 25 Oct. 1999, pp. 3/1 - 3/5.*

Foina, A. G.; Barbin, S. E.; Ramirez Fernandez, F. J.; 2007. A New Approach for Vehicle Access Control using Active RFID Tags. *In IMOC - International Microwave and Optoelectronics Conference, 2007. Proceedings of the 2007 IMOC.*

Kim, T.; Kim, H.; 2006. Access Control for Middleware in RFID Systems. *In Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference Vol. 2, 20-22 Feb. 2006, pp. 1020 – 1022.*

Ni, L. M.; Liu, Y.; Lau, Y. C.; 2003. LANDMARC: Indoor Location Sensing Using Active RFID. *In IEEE International Conference in Pervasive Computing and Communications, 2003, pp. 407-415.*

Raza, N.; Bradshaw, V.; Hague, M.; 1999. Applications of RFID technology. *In IEE Colloquium on, 25 October, 1999, pp. 1/1 - 1/5.*

Tuttle, J.R.; 1997. Traditional and emerging technologies and applications in the radio frequency identification industry. *In Radio Frequency Integrated Circuits (RFIC) Symposium on, 8-11 Jun. 1997, pp. 5 – 8.*

Zhao, J.; Zhang, Y.; Ye, M.; 2006. Research on the Received Signal Strength Indication Location Algorithm for RFID System. *In Communications and Information Technologies, 2006. ISCIT '06. International Symposium on Oct. 2006, pp. 881 – 885.*

SELF CONSTRUCTING NEURAL NETWORK ROBOT CONTROLLER BASED ON ON-LINE TASK PERFORMANCE FEEDBACK

Andreas Huemer[a], Mario Gongora[b] and David Elizondo[b]
[a]*Institute Of Creative Technologies, De Montfort University, Leicester, U.K.*
[b]*Centre for Computational Intelligence, De Montfort University, Leicester, U.K.*
{ahuemer, mgongora, elizondo}@dmu.ac.uk

Keywords: Spiking neural network, reinforcement learning, robot controller development.

Abstract: A novel methodology to create a powerful controller for robots that minimises the design effort is presented. We show that using the feedback from the robot itself, the system can learn from experience. A method is presented where the interpretation of the sensory feedback is integrated in the creation of the controller, which is achieved by growing a spiking neural network system. The feedback is extracted from a performance measuring function provided at the task definition stage, which takes into consideration the robot actions without the need for external or manual analysis.

1 INTRODUCTION

Machine intelligence and machine learning techniques have been used extensively in the tuning and optimisation of robot controllers capable of enabling the execution of complex tasks. Similarly, given the vast variety of possible conditions present in the real world, machine learning has been the subject of significant research to improve the responses of autonomous robots to a variety of situations. But the use of these techniques in the actual design of the controllers is still in its early stages. This paper presents a novel methodology capable of creating a spiking neural based controller, which is being studied and evaluated as part of our research in autonomous robots. Our method also takes into account the aspect of online learning which is a much more intuitive approach to real world problems, specifically for the study of robotics in non-structured environments.

One type of learning system that has been applied successfully to some control problems is based on neural networks. One approach to have robots that are able to adapt to completely new situations could be to provide a neural network with enough fully connected neurons, and have those connections adapted with known machine learning methods; this, in principle, would provide enough adaptable components in their control system. Alternatively, as has been shown by Elizondo et al. (Elizondo et al., 1995), partially connected neural networks are faster to train and have better generalisation capabilities. Similar effects have been found considering the number of neurons (Gómez et al., 2004), where it has been shown that it is not necessarily better to have more neurons.

In this paper we present a novel method for creating a neural network based robot controller which starts with a minimalistic neural network having a small number of neurons and connections and grow it until it can fulfil effectively the tasks required by the robot. The results presented have been evaluated with a set of experiments where a simulated robot learns to avoid obstacles while wandering around in a room.

We have created a self constructing controller for a robot which consists of a spiking neural network which learns from experience by connecting the neurons, adapting the connections and growing new neurons depending on a feedback process that will correspond to the measurement of a perceived "gratification" value of the robot. The measurement of the "gratification" that the robot perceives can be defined by an evaluation function that rewards the robot depending on the performance of the task, causing that the neural network develops itself. This self construction occurs without the need of any external or human intervention, creating a purely automated learning mechanism; the self construction can be guided

as well with runtime feedback from a trainer (either automated or human operated), representing the validation of an expert.

Reward based systems have been presented, as in (Florian, 2005) where a worm that was fed with positive reward when its mouth was moving towards its food source and negative reward when its mouth was moving in the other direction was simulated using a neural network to control its movements. Depending on the feedback the connections between the neurons were adapted, which finally took the mouth of the worm to the food source. A similar method was used in the experiments of this paper. A reward measurement will be used both for growing new neurons and for adapting the connections between them.

During the creation of the network we have separated the neural connections in two parts: artificial dendrites and axons. These do not only play an important role with the growth mechanism but in the basic decision mechanism for the actions.

At this stage of the research we have set some initial constraints to provide a reliable evaluation of the novel growth methods. For example recurrent connections and Spike Time Dependent Plasticity have been excluded, which would both increase the capabilities of the neural network as for example shown by Gers et al. (Gers et al., 2002) or Izhikevich (Izhikevich, 2006). However, they would also increase the dynamics of the network and hence the effort of evaluating it and the certainty of the evaluation at this initial stage.

The paper is organised in the following way. Section 2 explains the principle of the type of connections used in our neural network. In section 3 the basic learning mechanisms of the spiking neural network are described. The growth mechanism of the controller is discussed in section 4, followed by results of testing the mechanism in section 5 and an analysis of them in section 6. Section 7 contains concluding remarks. At the end some ideas for further work are mentioned, in section 8.

2 ACTION SELECTION

2.1 Neural Task Separation

In the model presented in this paper we use spiking neurons which send Boolean signals via the connections when a certain threshold potential is exceeded (a basic explanation of these can be found in (Vreeken, 2003)). For the experiments that are reported in this paper the threshold is kept constant and is the same in the whole neural network. This has some advantages such as that all new neurons can be created with the same properties.

For applications in robot control, we can use neural networks for classification and for action selection. The classification task is needed to reduce the number of neurons that are responsible for selecting an action. The number of connections between neurons can be reduced as well by merging certain input patterns into classes. Classification is usually done by connecting several input neurons to a neuron that represents the class that all of the connected input neurons belong to. This process is often called representation and can be distributed over several layers. By combining several neurons into a single one at the next level, in the succeeding parts of the network the number of neurons and connections can be reduced as well. This optimisation processes are critical as it has been shown that less neurons and connections result in less computation requirement and better development of the network (Elizondo et al., 1995) (Gómez et al., 2004).

As we need the system to be capable of dealing with classification and action selection mechanisms at the same time, it is useful to separate the connections into two parts. For the model we are presenting here, dendrites connect axons with a postsynaptic neuron and axons connect a presynaptic neuron with a dendrite.

2.2 Neural Task Processing

A presynaptic neuron is activated when its potential reaches a threshold and fires off "spikes" via its axons. The "spike" is an all-or-nothing signal, but its influence on the connected dendrite is weighted. The weights are adjusted by the learning process discussed later. A single axon can be sufficient to activate a dendrite. More issues of the separation of connections into axons and dendrites are discussed in section 4.

The signals travel from a presynaptic to a postsynaptic neuron as explained by the following equations. For all equations it is assumed that all axon weights of one dendrite sum up to 1. If weights are changed, they have to be normalised afterwards, so that the sum is 1 again.

Input of a dendrite:

$$I_d = \sum_p O_a(p) \cdot w_a(p) \tag{1}$$

where I_d is the dendrite's input. $O_a(p)$ is the output of axon p, which is 1, if the presynaptic neuron has fired and 0 otherwise. $w_a(p)$ is the weight of axon p.

Output of a dendrite:

$$O_d = \frac{1}{1 + e^{-b \cdot (I_d - \theta_d)}} \tag{2}$$

where O_d is the dendrite's output and I_d is its input. θ_d is a threshold value for the dendrite. b is an activation constant and defines the abruptness of activation.

The influence of the dendrites on the postsynaptic neuron is again weighted and the weights are again adapted by the learning process. Contrary to the situation in a dendrite, a neuron is only activated and fires when many or all of the excitatory dendrites are active. An excitatory dendrite has got a positive weight, while an inhibitory dendrite has got a negative weight and decreases the probability of a neuron to fire.

Similar to the axons, all excitatory dendrites of one neuron sum up to 1. The inhibitory dendrites of a neuron sum up to -1. Again, normalisation is needed after weight changes.

Input of a neuron:

$$I_j = \sum_q O_d(q) \cdot w_d(q) \tag{3}$$

where I_j is the input of the postsynaptic neuron j, $O_d(q)$ is the output of dendrite q and $w_d(q)$ is the weight of dendrite q.

Change of neuron potential:

$$P_j(t+1) = \delta \cdot P_j(t) + I_j \tag{4}$$

where the new neuron potential $P_j(t+1)$ is calculated from the potential of the last time step t, $P_j(t)$, and the current contribution by the neuron input I_j. δ is a constant between 0 and 1 for recovering to the resting potential (which is 0 in this case) with time. The fact that δ will never bring the potential exactly to the resting potential, is not very important but can be avoided with a total reset when reaching a small range around the resting value.

The postsynaptic neuron is activated when its potential reaches the threshold θ_j and becomes a presynaptic neuron itself for neurons which its own axons are connected to. After firing the neuron resets its potential to its resting state. In contrast to similar neuron models that are for example summarised by Katic (Katic, 2006), a refractory period is not implemented here.

The processes for a neuron are shown in figure 1.

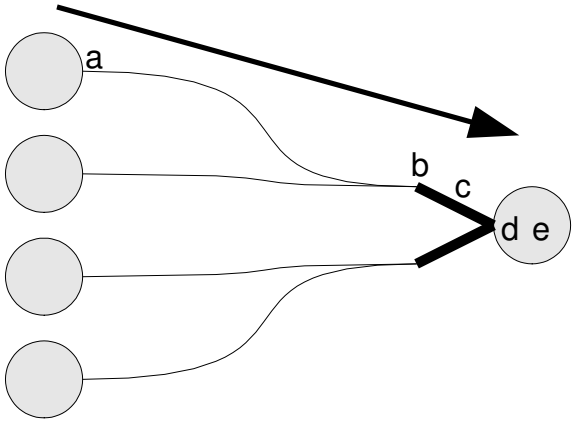


Figure 1: A spike is produced when the presynaptic neuron fires and is sent to a dendrite (a). The dendrite sums up the weighted spikes (b, equation 1) and calculates its output (c, equation 2). The postsynaptic neuron sums up the weighted output of all of its dendrites (d, equation 3) and calculates its new potential (e, equation 4).

3 MACHINE LEARNING AND EXPERIENCE

3.1 Measuring and using Feedback

For complex situations as usually encountered in robotic applications in the real world there is rarely an exact error value which is known and is to be minimised.

As experience based learning is meant to use past events to correct and optimise the behaviour, we need a measurement of error or its equivalent if the former is not directly available. We have chosen as an alternative to an error value one or more reward values that can be fed into the control system to represent the "well-being" of the robot.

These reward values can be modified by positive "good experience" or negative "bad experience" feedback relative to the robot's performance in the task that has been assigned. The positive experience is to be maximised.

In our model we have a single reward value that represents the general "well-being" of the robot. Its range is kept from -1, very bad, to 1, very good. The calculation of the reward can be varied. Usually it combines current measurements like fast movement, crashes or the energy level with residual effects of recent ones to avoid too rapid changes. For example, if the robot crashes into an object, the value for representing its "well-being" will be negative for a short while. A robot that moves away from a obstacle after crashing into it deserves an increase of the reward.

For the methods that are explained here it is assumed to have a meaningful global reward value $\Pi(t)$ at each time step t. This value can be added to a learning rule as an additional factor. Different authors, all of them using different neuron functions and learning functions, have shown that this surprisingly simple method can successfully be used to implement reinforcement learning in a neural network (Daucé and Henry, 2006) (Florian, 2005) (Izhikevich, 2007). They do not need an external module that evaluates and changes the connections of the network after each processing step any more.

An example for adapting axons and dendrites using Activation Dependent Plasticity is shown below. Activation Dependent Plasticity is based on Hebb's ideas of strengthening connections that fire together (Hebb, 1949). As shown by Izhikevich reward can also be integrated into the more sophisticated Spike Time Dependent Plasticity (STDP) learning model (Izhikevich, 2007).

Adaptation of an axon weight:

$$w_a(t+1) = w_a(t) + \eta_a \cdot \Pi(t) \cdot \phi_a \cdot O_d \qquad (5)$$

where $w_a(t)$ and $w_a(t+1)$ are the axon weights before and after the adaptation. η_a is the learning factor for axons and O_d is the recent output of the connected dendrite. ϕ_a shows if the axon was active shortly before the postsynaptic neuron fired. For STDP this value can be the result of a function that takes into consideration the time when spikes were transmitted via the axon. In any case ϕ_a is a value from 0 to 1.

$\Pi(t)$ is the current reward. If it is positive, the strength of the axon will increase. A negative value will decrease the strength of the axon.

Adaptation of a dendrite weight:

$$w_d(t+1) = w_d(t) + \eta_d \cdot \Pi(t) \cdot \phi_d \qquad (6)$$

where $w_d(t)$ and $w_d(t+1)$ are the dendrite weights before and after the adaptation. η_d is the learning factor for dendrites and ϕ_d is the activity value of the dendrite. ϕ_d is the equivalent of ϕ_a, but ϕ_d represents the activity of the dendrite.

With this function, active excitatory dendrites are strengthened and active inhibitory dendrites are weakened, if the current reward $\Pi(t)$ is positive. Otherwise excitatory dendrites are weakened and inhibitory dendrites are strengthened.

3.2 Delayed Feedback

In robotics and maybe other real-time control applications it is very important to consider delayed sensorial and perception issues when dealing with feedback from the environment and ensuing rewards. When weights are adapted and as discussed later also neurons are created on the controller based on the current reward, this may be a problem. Typically, sensor based feedback is received some time after the responsible action has been decided and executed. Depending on the task the robot is performing, the time differences can vary significantly.

There are two components to consider for tackling this issue efficiently:

- Feedback is not fed directly into the neural network but just changes the current reward value, which also contains residual effects of past feedback. This avoids fast changes of the reward value, which would be difficult to assign to a certain neuron activity pattern.
- In spiking neural networks, there is no single event that is responsible for an action, but a continuous flow of spikes. In control terms, this is equivalent to having the integral element of a PID scheme; this acts as an embedded filter that makes that the input pattern, and hence the spiking pattern, does not change rapidly if a certain feedback is received. Figure 2 shows an example situation for this issue.

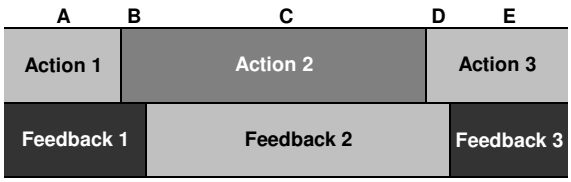


Figure 2: Section A in the figure illustrates any previous action of the robot, for example "turning right". In section B the robot has started a new action like "moving forward" but still receives the feedback that should be assigned to the previous action. Section C shows the time when feedback is correctly assigned to the current action. In section D the next action has already started but the feedback is the reaction to action 2. Section E completely belongs to the next action. Sections B and D, where feedback is not assigned correctly, are very short compared to the other sections.

In many robotics situations it is still difficult to assign the feedback correctly, for example if there is a big time difference between action and feedback, or if there are many concurrent tasks with opposite actions or feedback values at the same time. However, even humans do not always arrive at the correct conclusions and therefore, although is our aim for robots to deal with very complex relations, it is not realistic to expect it to happen with all.

In further work, a method will be introduced that may enable a robot to deal with delayed feedback in a better way, or may even be used to predict feedback. The method will be refined through further experimentation and research.

4 NEURAL CONTROLLER CONSTRUCTION

The neural network to be grown to create a robot control system initially has no links from the input to the output. The developer only defines the input neurons and how they are fed with signals to produce spikes, the output neurons and how their signals are used, and

how the global reward is calculated. An example for how this is done is explained in section 5.

If a non-input neuron has got no predecessors (neurons, which it gets spikes from), it creates a new excitatory dendrite and connects it to any neuron. In the experiments that are discussed later a predecessor is looked for that is positioned above the postsynaptic neuron in a layered network structure. Excitatory dendrites can also look for new presynaptic neurons every now and then and connect them with weak strength (low weights). That way a new connection does not abruptly change an established behaviour.

The method to grow new axons, which are the connections between presynaptic neurons and dendrites, can only be used for the action selection task. To classify different input patterns a method that creates new neurons is presented. Liu, Buller and Joachimczak have already shown that correlations between certain input patterns and a certain reward can be stored by creating new neurons (Liu and Buller, 2005) (Liu et al., 2006).

In the model proposed here, if the current reward is positive, a neuron that was active recently should be active again in similar situations, because, if a certain action was responsible for positive reward, it may be successful again. In section 3 delayed feedback was discussed. To avoid wrong correlations between feedback and neuron activity, a neuron will only create a connection to a new neuron in the following way, if it was active for some time already:

- All axons with enough influence on a neuron that was active before receiving positive feedback are redirected to a new neuron. The influence depends on the axon weights and the recent activity of the presynaptic neurons.
- The redirected connections are no longer just axons to one dendrite but are all connected to their own dendrite at the new neuron. This stores the combination of input signals.
- The old axons need not be removed completely, but most of their strength will be moved to a new axon that is connected to the new neuron.
- The process, which is illustrated in figure 3, is repeated for all dendrites of a neuron.

For a negative reward the process of creating a new neuron is similar, but the new neuron is not connected by a new axon but by a new inhibitory dendrite. In the future a similar input pattern will then inhibit the neuron that was active before receiving negative feedback. Bad actions will be suppressed that way.

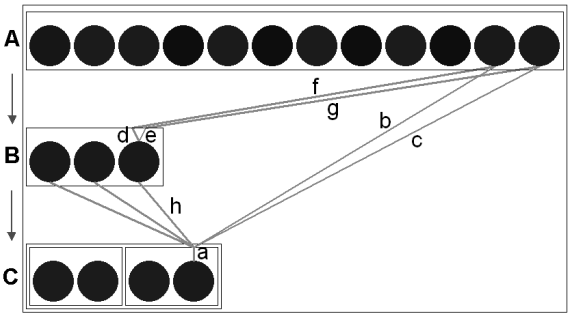


Figure 3: The excitatory dendrite a is connected to two neurons of the input layer A via the axons b and c. Both were active when there was a significant positive reward. A new neuron was created in the hidden layer B that connects the same input neurons by two dendrites (d, e) and one axon for both dendrites (f, g). Then the new neuron was connected to dendrite a (axon h) of the neuron in the output layer C.

5 EXPERIMENTAL SETUP

Our novel methodology for autonomously constructing a spiking neural network based controller from a basic initial definition structure was tested with a simulation of a Pioneer Peoplebot which moves using differential steering, as depicted in figure 4. The initial neural structure consists of 12 input neurons (2 for each sensor), 4 output neurons (2 for each motor), and no connections as indicated by layers A and C in figure 3.

The input neurons are fed by values from 6 sonar sensors as shown in figure 4, each sensor feeds the input of 2 neurons. The sonar sensors are arranged so that 4 scan the front of the robot and 2 scan the rear as shown in the figure. The distance value is processed so that one input neuron fires more frequently as the measured distance increases and the other neuron connected to the same sensor fires more frequently as the distance decreases.

For the actuator control, the output connections are configured so that the more frequently one of the output neurons connected to each motor fires, the faster this motor will try to drive forward. The more frequently the other output neuron connected to the same motor fires, the faster that motor will try to turn backward. The final speed that each motor will drive is calculated by the difference between both neurons.

With the sensor and actuator configuration described, the experiment was setup for the robot to learn to wander around randomly in the simulated office shown in figure 5 while avoiding obstacles.

In each control cycle the global reward value is updated along with the processing of the whole simulated system and movement of the robot. The orig-

inal Peoplebot's bumpers are included in the simulation and are used to detect collisions with obstacles, and are used to penalise significantly the reward values when such a collision occurs. The reward is increased continuously as the robot travels farther during its wandering behaviour. Backward movement should only be acceptable when recovering from a collision, therefore it will only be used to increase the robot's reward value in that case, while it is used to decrease this value for all other cases. The straighter the robot goes the more positive reward it will receive. So in the long run straight movement will be prefered compared to moving in circles.

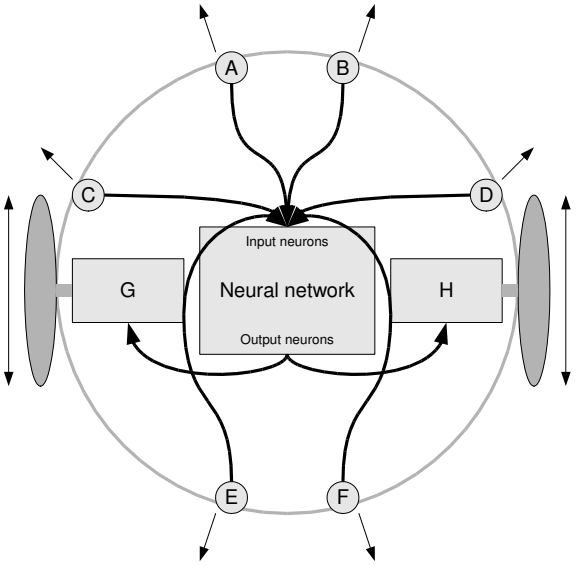


Figure 4: The robot interface includes sonar sensors A to F and motors G and H.

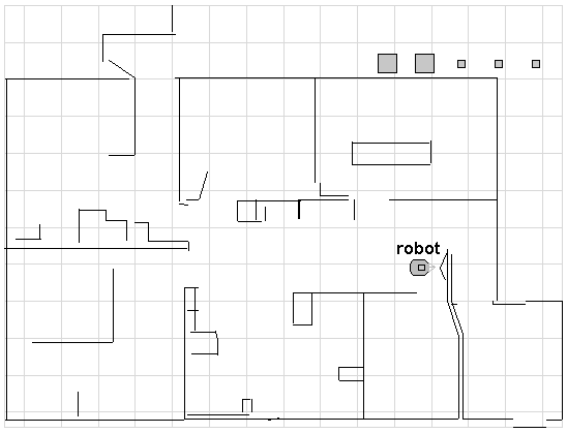


Figure 5: A simulated Peoplebot is situated in this simulated office provided by MobileRobots/ActivMedia.

Once the controller starts to create connections and new neurons, they are organised in layers as indicated in figure 3. One layer contains the input neurons and output neurons are located at the layer in the opposite end. New layers can be created in-between these two to accommodate new neurons. For these experiments the network is evaluated as a strict feed forward network, which means that there are no connections to neurons from the same layer (i.e. no local inhibition) or to neurons from a previous layer (i.e. no recurrent connections).

For all experiments Activation Dependent Plasticity was used. That means actions are selected based on co-activation of certain neurons without considering the exact spike times. This is suitable for these experiments where the robot needs to exhibit a purely reactive behaviour; therefore constraints are accepted in terms of having no competing actions that need executing in parallel and without planning tasks where a sequence or time synchronous actions need to be executed. Similarly, although advantages of spike time dependent processes have for example been investigated by Izhikevich (Izhikevich, 2006) or van Leeuwen (van Leeuwen, 2004), the learning and growing mechanisms are not based on such times to make evaluation easier.

From the explanation of the process to create new neurons presented in section 4, an additional issue had to be considered to avoid the creation of large sequences of neurons when a particular high reward value is received from the feedback system. When a new neuron that stores an input pattern that seems to be responsible for a certain reward value is created, itself will again generate a reason to produce another neuron because its output can also be assigned to a certain reward. To avoid this, the age of the connections is considered so that young axons, even if they seem to be responsible for a lot of reward, will not lead to a new neuron. The effect of this can be observed in Figure 6 where a growing process with and without this consideration is shown. Without considering the age of the connections the neural network is growing so fast that the time needed for all calculations of one time step increases enormously. Ignoring young connections saves an extreme amount of new neurons and also connections.

6 ANALYSIS OF RESULTS

Various experiments were run and consistent results were obtained where the robot was able to learn autonomously to wander around while turning away from obstacles. Figure 7 shows an example run in

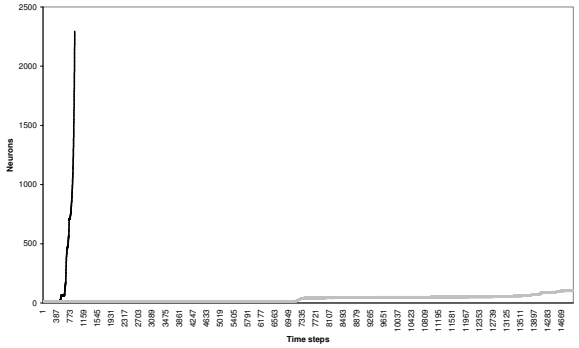


Figure 6: The black line shows the number of neurons without considering the age of the connections for creating new neurons. For the grey line connections younger than 6000 time steps were ignored at the growing process.

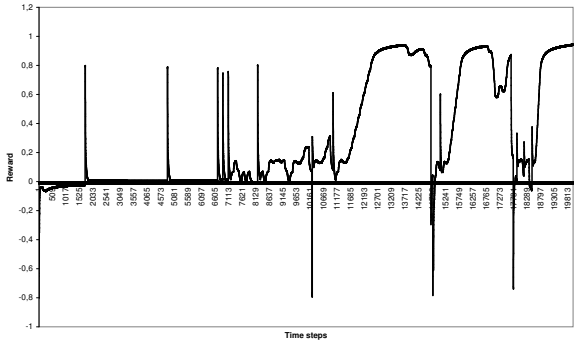


Figure 7: At the beginning the robot did not perceive very much reward. After some random movements the robot learned how to increase positive reward. Smaller reward at later stages shows that the robot slowed down near obstacles. The negative amplitudes show that not all obstacles could be avoided.

which the robot perceived more reward when its experience increased.

The trend of reward seen in figure 7, where the feedback varies significantly from high to negative might look obvious, but it is critical for the robot to be capable of continuous adaptation. A monotonic increase in the suitability of the system, as is achieved with other machine learning approaches, would mean that once the controller learns to perform a task, it cannot re-adapt to any alteration. This supports further the suitability and potential of our novel feedback guided methodology for autonomously creating robot controllers.

Table 1 shows some results of a test sample of 50 simulation runs, each run starting with the initial network definition and without connections, the system executed 20000 time steps, where one time step is over when all neurons have been updated once. The speed values are measured in internal simulation units. In all cases, the same number of inhibitory axons as inhibitory dendrites were created, because as explained in section 4 a new inhibitory axon is always created with a new dendrite.

Table 1: The table shows results of 50 simulation runs.

	Min.	Max.	Avg.
Total reward	−164.96	7412.83	3160.72
Avg. reward	−0.01	0.37	0.16
Max. speed	395.00	1303.00	970.38
Avg. speed	4.09	388.24	190.48
Crashes	0.00	16.00	3.36
Neurons	16.00	32.00	20.62
Exc. axons	14.00	126.00	34.46
Exc. dendrites	4.00	96.00	18.98
Inh. axons	4.00	6.00	4.12

As the system responds autonomously to the feedback received in the form of reward, it is possible to add or remove neurons to the input or output layers at any time, associated either to existing or new sensors and actuators. The controller will continue to receive the feedback and continue to adapt autonomously. This provides a very powerful potential for online adaptation to both new situations and new configurations of the robot's hardware. Even in non-explicit situations, such as standard wear and tear of the system or degradation and failure of a particular component (sensor or actuator), as long as the task is still possible to be achieved, the controller will adapt to it.

There is some potential for improving the methods for the robot to learn to recover if it crashes into an object. The different parameters of the neural network have to be adjusted and tested to render the strengths and weaknesses of the proposed robot control system more precisely.

7 CONCLUSIONS

We have shown that a robot controller can be created autonomously using our novel methodology. A neural network can be grown based on the reward measured by a feedback function which analyses in real time the performance of a task.

We have defined a novel methodology where the design of a robot controller is defined in a completely new way: as an intuitive process where all that is required is to identify the inputs, the outputs and the mechanism to quantify a reward perception from feedback that depends on the performance of the system carrying out a task.

In addition, since the complete process is integrated in a single and robust stage capable of learning

from experience in a continuous way when running, this methodology has the potential to be an adaptable system where we can add or remove any sensors or actuators, and the controller can adapt autonomously and online to the new situation.

8 FURTHER WORK

The different parameters that define the speed of adapting connection weights and the way of creating new neurons and connections have to be investigated further to evaluate our novel methodology for creating controllers for concurrent tasks. These investigations will lead us to find an elaborate but still very basic "artificial brain" model that enables a system to achieve a sophisticated level compared to other artificial intelligence models by learning from experience efficiently.

When the basic methods are investigated in detail, some extensions can be added like Spike Time Dependent Plasticity or a feedback prediction mechanism. Initial ideas for both enhancements were discussed in this paper. Those improvements would help the controlled systems to deal with more complex situations, especially when timing considerations are important.

As mentioned in section 3 assigning delayed feedback more efficiently or even predicting feedback will be an interesting research issue for future work. The idea is that a neuron that receives positive or negative reward very often when it is active will probably receive the same reward also in the future. Predicting reward could actually be one reason for producing reward. This earlier reward may now be correlated to the activity of another neuron. That neuron could again produce reward when predicting it. By the recursive process reward could potentially be predicted progressively earlier.

REFERENCES

Daucé, E. and Henry, F. (2006). Hebbian learning in large recurrent neural networks. Technical report, Movement and Perception Lab, Marseille.

Elizondo, D., Fiesler, E., and Korczak, J. (1995). Non-ontogenetic sparse neural networks. In *International Conference on Neural Networks 1995, IEEE*, volume 26, pages 290–295.

Florian, R. V. (2005). A reinforcement learning algorithm for spiking neural networks. In *Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 299–306.

Gers, F. A., Schraudolph, N. N., and Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, 3:115–143.

Gómez, G., Lungarella, M., Hotz, P. E., Matsushita, K., and Pfeifer, R. (2004). Simulating development in a real robot: On the concurrent increase of sensory, motor, and neural complexity. In *Proceedings of the Fourth International Workshop on Epigenetic Robotics*, pages 119–122.

Hebb, D. O. (1949). *The Organization of Behaviour: A Neuropsychological Approach*. John Wiley & Sons, New York.

Izhikevich, E. M. (2006). Polychronization: Computation with spikes. *Neural Computation*, 18:245–282.

Izhikevich, E. M. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral Cortex*, 10:1093–1102.

Katic, D. (2006). Leaky-Integrate-and-Fire und Spike Response Modell. Technical report, Institut fï¿½r Technische Informatik, Universitï¿½t Karlsruhe.

Liu, J. and Buller, A. (2005). Self-development of motor abilities resulting from the growth of a neural network reinforced by pleasure and tension. In *Proceedings of the 4th International Conference on Development and Learning 2005*, pages 121–125.

Liu, J., Buller, A., and Joachimczak, M. (2006). Self-motivated learning agent: Skill-development in a growing network mediated by pleasure and tensions. *Transactions of the Institute of Systems, Control and Information Engineers*, 19(5):169–176.

van Leeuwen, M. (2004). Spike timing dependent structural plasticity in a single model neuron. Master's thesis, Intelligent Systems Group, Institute for Information and Computing Sciences, Utrecht University.

Vreeken, J. (2003). Spiking neural networks, an introduction. Technical report, Intelligent Systems Group, Institute for Information and Computing Sciences, Utrecht University.

AUTOMATIC PARAMETERIZATION FOR EXPEDITIOUS MODELLING OF VIRTUAL URBAN ENVIRONMENTS

A New Hybrid Metaheuristic

Filipe Cruz, António Coelho and Luis Paulo Reis
Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias 4200-465, Portugal
filipe.cruz@fe.up.pt, acoelho@fe.up.pt, lpreis@fe.up.pt

Keywords: Metaheuristics, optimization algorithms, optimization systems, expeditious modelling.

Abstract: Expeditious modelling of virtual urban environments consists of generating realistic 3d models from limited information. It has several practical applications but typically suffers from a lack of accuracy in the parameter values that feed the modeller. By gathering small amounts of information about certain key urban areas, it becomes possible to feed a system that automatically compares and adjusts the input parameter values to find optimal solutions of parameter combinations that resemble the real life model. These correctly parameterized rules can then be reapplied to generate virtual models of real areas with similar characteristics to the referenced area. Based on several nature inspired metaheuristic algorithms such as genetic algorithms, simulated annealing and harmony search, this paper presents a new hybrid metaheuristic algorithm capable of optimizing functions with both discrete and continuous parameters and offer competitive results in a highly neglected field of application.

1 INTRODUCTION

There has been a growing need for expeditious modelling systems for urban environments in recent years. Applications include virtual city tours, georeferenced services, urban planning.

Creating urban models that are accurate representations of the real world can be extremely hard in terms of information gathering and invested man power. Close resemblances to the real world is often an acceptable compromise for some applications. It often occurs that it is quite impossible to gather information from the entire city you are trying to model but it is somewhat easy to extract detailed information from certain key-defining small areas from within the city. Generating expeditious modelling rules that will match such information is an important requirement. The artificial intelligence field of optimum search includes meta heuristic algorithms that can determine optimum parameter values.

The work described in this paper focuses on solving this problem by creating a new hybrid meta heuristic algorithm, competitive and capable of handling both discrete and continuous parameters. Adapting ideas from several other nature based meta heuristic algorithms such as genetic algorithms, simulated annealing and harmony search.

This paper is divided into 5 sections: the first serving as the introduction, the second section referring to related work, the third describing the developed system, the fourth presenting and discussing the results achieved, and the fifth presenting the conclusions and future work.

2 RELATED WORK

2.1 Expeditious Modelling of Virtual Urban Environments

The modelling of virtual urban environments has many different applications including virtual city tours (Schilling and Coors, 2003), georeferenced services (Ito et al., 2005), cultural heritage preservation (Hildebrand et al., 2000) (Zach et al., 2001) and urban planning. Often there is a need for realistic or semi-realistic models of cities, however, modelling accu-

rate realistic models grows problematic considering increasing needs in size and complexity.

Using principles of l-system (Lindenmayer, 1968) (Prusinkiewicz, 1986), it is possible to define sets of production rules to model all kinds of urban environment elements (Parish and Muller, 2001) based on limited georeferenced data (Muller et al., 2006). It's often difficult to acquire reliable data of an entire city we are trying to model, however it is relatively easy to gather information from certain key-defining sections of a city and assign the data to a georeferenced database. The production rules with the data collected will enable the creation of models resembling the area of gathered information.

2.2 Meta Heuristic Algorithms for Solving Optimization Problems

The field of artificial intelligence branches many areas (Russel and Norvig, 2003), one of the most relevant ones is the field of optimum search. Meta heuristic algorithms play an important role in this field. Different meta heuristic variants and hybrid versions refer to evolutionary computation or nature inspired behaviors: genetic algorithms (Holland, 1992) (Goldberg, 2002), simulated annealing (Kirkpatrick et al., 1983) (Aarts et al., 2005) and harmony search (Lee and Geem, 2005) (Mahdavi et al., 2007).

The common principle is to find the best combination of parameters of a given vector function such that a related objective function is maximized or minimized. This process is iterated through educated random guesses following the heuristic logic of the algorithm. There is no best heuristic, performance varies according to the constraints of problem. A big number of proposals for hybrid or customized adaptations is found in recent literature (Deep and Thakur, 2007) (Arumugama et al., 2005) (Kumar et al., 2007) (Lee and Geem, 2005) (Mishra et al., 2005).

3 IMPLEMENTATION

3.1 Problem Statement

An l-systems based expeditious modeler for the generation of realistic urban environments becomes more valuable with an optimum parameterization system.

The system must handle the input of boolean, integer and real based parameters. The system must allow a configuration easily adaptable to the problem. When solving linear constrained problems the system must out-perform simple random based algorithms such as hill climber and random search. When solving problems with multiple local maximums the system is required to perform above par of classic meta heuristic algorithms such as simulated annealing, genetic algorithm and harmony search.

3.2 Hybrid Optimizer Meta Heuristic Algorithm

The hybrid meta heuristic is inspired by basic principles of real based genetic algorithms and concepts from simulated annealing and harmony search.

There are two families of populations resident in memory at all times, the original parent family and the top list family. Their dimensions can be configured by XML. Each iteration step of the meta heuristic algorithm consists on creating a new original parent family generation. The top list family maintains in memory the best solutions ever found so far, sorted by quality.

Each solution stores values for all parameters being calibrated. Each parameter has information regarding its type (integer, boolean, real) and scope (minimum and maximum values). The type and scope for each parameter are pre-configured by XML. The values for the first generation of the original parent family are calculated randomly within its scope boundaries. The first generation of the top list family is obtained by sorting the first generation of original parent family. These values can also be loaded from disk to test scenarios in same starting terms.

Each following original parent generation is obtained by cross-breeding the original parent family with a chosen member of the top list family. Ensuring an elitist selection behavior inspired by genetic engineering. Several threshold variables further influence the selection of the new solution to ensure a wider search space scope not limited to the first generation. These variables incorporate monte carlo methods (Metropolis and Ulam, 1949) using probability thresholds inspired by simulated annealing and harmony search. Some are, or can be, affected by what is referred to as *globalentropy*, an internal value increasing by each passing iteration as described in (1). There are a total of five threshold parameters which must be calibrated considering the problem.

Random New Struct Threshold (*trns*), affects the probability of choosing a completely random new solution. The higher this value the more probable it becomes to occur a total random creation of a new solution as seen in formula (3).

$$globalentropy = iterationstep/maxsteps \quad (1)$$

$$trns = thresholdRandomNewStruct \quad (2)$$

$$randSol = rand() * globalentropy < trns \quad (3)$$

Random New Type Threshold (*trnt*), affects the probability of choosing a completely random new value for each of the solution parameter types. The higher this value is the more probable it is to occur a totally random new value for the current parameter type of the solution as seen in formula (5).

$$trnt = thresholdRandomNewType \quad (4)$$

$$randType = rand() * globalentropy < trnt \quad (5)$$

Toplist Dispersion Threshold (*ttld*), affects the probability of choosing lower ranking *toplist* parents to cross the solution with. The higher this value the wider the scope of choice as seen in formula (8).

$$ttld = thresholdToplistDispersion \quad (6)$$

$$ttld > 1.0 \; : \; ttld = 1 - globalentropy \quad (7)$$

$$victim = (rand() * ttld * maxFamilySize) \quad (8)$$

Typevalue Dispersion Threshold (*ttvd*), affects the parental gene influence for each value of the parameter types of the solution as seen in formula (13).

$$tlv = topslistParentValue \quad (9)$$

$$orv = originalParentValue \quad (10)$$

$$ttvd = thresholdTypevalueDispersion \quad (11)$$

$$ttvd > 1.0 \; : \; ttvd = 1 - globalentropy \quad (12)$$

$$newvalue = (ttvd * orv) + ((1 - ttvd) * tlv) \quad (13)$$

Typevalue Entropy Threshold (*ttve*), affects the probability of scope jitter for each value of the parameter types of the solutionas can be seen in formula (17).

$$ttve = thresholdTypevalueEntropy \quad (14)$$

$$ttve > 1.0 \; : \; ttve = 1 - globalentropy \quad (15)$$

$$scope = \|(OParentValue - TLParentValue)\| \quad (16)$$

$$igl = (1.0F - globalentropy) \quad (17)$$

$$ttvss = igl * igl * igl \quad (18)$$

$$range = MaxParamValue - MinParamValue \quad (19)$$

$$scope < ttvss * range \; : \; maxscope = range * igl \quad (20)$$

$$scope > ttvss * range \; : \; maxscope = scope * ttve \quad (21)$$

$$maxscope = scope * rand() \quad (22)$$

$$newvalue = newvalue + scope - (maxscope/2) \quad (23)$$

4 RESULTS

A test-case was prepared involving the parameterization of a set of production rules which would model several buildings with certain height values missing. The known information from all of the buildings included georeferenced location and the values of the buildings perimeter, area and *bottomzvalue*. The unknown information from some of the buildings comprised solely the *topzvalue*.

The production rule used to estimate the unknown *topzvalue* from the buildings is described mathematically in formula (27).

$$av = (cra - 1) * fca * area \quad (24)$$

$$ab = (crb - 1) * fcb * bottomzvalue \quad (25)$$

$$ap = (crp - 1) * fcp * perimeter \quad (26)$$

$$topzvalue = avgz + disp * (av + ab + ap) \quad (27)$$

The formula implies a relation between the building's area (24), perimeter (26) and *bottomzvalue* (25) with the building's height to estimate a realistic *topzvalue* for input to the expedite modeler.

Our formula has a total of 8 unknown fields to be parametrized: *avgz* [100.0 .. 120.0], the average height for all the buildings. *disp* [0.0 .. 1.0], the dispersion rate from the average height. *cra* [0 .. 3], area correlation. *crb* [0 .. 3], *bottomzvalue* correlation. *crp* [0 .. 3], perimeter correlation. *fra* [0.0 .. 1.0], the area value correlation factor. *frb* [0.0 .. 1.0], the *bottomzvalue* correlation factor. *frp* [0.0 .. 1.0], the factor of the perimeter correlation.

Different configurations of the meta heuristic algorithm were tested with a fixed *toplist* family size of 20. The different tested configurations include the behavior of some classic algorithms: random search, hill climber and simulated annealing. A few additional configurations were also tested for comparative performance results. Each test iterated 50 generations with a family size of 8 and were labeled as follows: *h1r8*, hybrid random search. *h1n8*, hybrid new configuration. *h2n8*, hybrid second new configuration. *h3n8*, hybrid third new configuration. *hhc8*, hybrid hill climber. *hsa8*, hybrid simulated annealing.

The calibration parameters of each configuration tested can be consulted in Table 1.

Table 1: Threshold parameters of the different configurations.

config	trns	trnt	ttld	ttvd	ttve
h1r8	1.0	1.0	0.0	0.5	0.5
h1n8	0.01	0.01	0.1	1.1	1.1
h2n8	0.001	0.001	0.4	0.15	1.1
h3n8	0.001	0.001	0.1	0.1	1.1
hhc8	0.01	0.01	0.0	1.0	0.1
hsa8	0.01	0.01	0.0	1.0	1.1

The quality function for the test case is calculated as a weighted sum of the height from the involved buildings. All simulations were performed three times to present some insight on how deeply the performance of the meta heuristics algorithm stochastic nature is

affected. Table 2 displays the progressive results obtained from our test case. It shows the quality value of the best solution found at 2%, 40%, 80% and at the end of the iteration process and allow us to speculate on how each configuration depend on the initial state and perform comparatively to known random search, hillclimber and simulated annealing algorithms.

Table 2: Progressive solution quality results from the different configurations.

config	2%	40%	80%	final
h1r8-1	163.95	54.254	44.854	44.854
h1r8-2	373.90	98.338	14.557	14.557
h1n8-1	272.53	9.319	9.319	9.319
h1n8-2	29.479	10.910	8.972	8.972
h2n8-1	438.53	24.047	14.425	14.425
h2n8-2	1038.61	1.758	1.758	1.758
h3n8-1	22.164	6.418	0.175	0.175
h3n8-2	288.56	3.934	0.798	0.798
hhc8-1	117.35	47.728	47.728	32.145
hhc8-2	1069.77	427.75	232.77	104.08
hsa8-1	1207.92	75.354	40.354	40.354
hsa8-2	315.95	137.05	42.689	9.804

5 CONCLUSIONS AND FUTURE WORK

An automatic parameterization system for expeditious modelling of virtual urban environments has been developed with a successful field application. Our test case, despite its relatively low complexity and linear constraints, demonstrates the potential of our new hybrid meta heuristic algorithm in finding optimum parameters for rule sets of expeditious modelling competitively to common optimum search algorithms. Further test results are required to statistically compare the performance of the new hybrid meta heuristic algorithm with other meta heuristic algorithms and parameter optimization problems.

An envisioned improvement to the system involves applying principles of nested partition and linear regression to strengthen performance.

REFERENCES

Aarts, E., Korst, J., and Michiels, W. (2005). Simulated annealing. pages 187–210. Springer.

Arumugama, M., Raoa, M., and Palaniappanb, R. (2005). New hybrid genetic operators for real coded genetic algorithm to compute optimal control of a class of hybrid systems. pages 38–52.

Deep, K. and Thakur, M. (2007). A new mutation operator for real coded genetic algorithms.

Goldberg, D. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Addison-Wesley, Reading, MA.

Hildebrand, A., Dahne, P., Seibert, F., Christou, I., Demiris, A., Diorinos, M., Ioannidis, N., Almeida, L., Diogo, A., and Weidenhausen, J. (2000). Augmented reality based system for personalized tours in cultural heritage sites.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.

Ito, H., Teh, S., Nakanishi, H., and Tagawa, T. (2005). Design and implementation of 3d interface for digital city. volume 88, pages 60–68.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimizing by simulated annealing. volume 220.

Kumar, A., Sharma, D., and Kalyanmoy, D. (2007). A hybrid multi-objective optimization procedure using pcx based nsga-ii and sequential quadratic programming.

Lee, K. and Geem, Z. (2005). A new meta-heuristic algorithm for continuous engineering optimization harmony search theory and practice. volume 194, pages 3902–3933.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development, parts i and ii. volume 18, pages 280–315.

Mahdavi, M., Fesanghary, M., and Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. pages 1567–1579.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. volume 44, pages 335–341.

Mishra, N., Prakash, M., Tiwari, R., Shankar, F., and Chan, T. (2005). Hybrid tabusimulated annealing based approach to solve multi-constraint product mix decision problem.

Muller, P., Wonka, P., Haegler, S., Ulmer, A., and Gool, L. (2006). Procedural modeling of buildings.

Parish, Y. and Muller, P. (2001). Procedural modeling of buildings.

Prusinkiewicz, P. (1986). Graphical applications of l-systems. pages 247–253.

Russel, S. and Norvig, P. (2003). *Artificial Intelligence: A modern approach*. Prentice-Hall, 2 edition.

Schilling, A. and Coors, V. (2003). Generation of vrml city models for focus based tour animations. pages 39–48.

Zach, C., Klaus, A., Bauer, J., Konrad, K., and Grabner, M. (2001). Modeling and visualizing the cultural heritage data set of graz. pages 219–226.

WISA
A Modular and Hybrid Expert System for Machine and Plant Diagnosis

Mario Thron, Thomas Bangemann and Nico Suchold
Institut für Automation und Kommunikation e.V. Magdeburg, Steinfeldstr 3, D-39179 Barleben, Germany
Mario.Thron@ifak.eu, Thomas.Bangemann@ifak.eu, Nico.Suchold@ifak.eu

Keywords: Industrial diagnosis, expert system, Fuzzy Logic, Bayesian Network, logic description, modularity.

Abstract: Expert systems are well known tools for diagnosis purposes in medicine and industry. One problem is the hard efford, to create the knowledge base. This article describes an expert system for industrial diagnosis and shows an efficient approach for the creation of the rule base, which is based on the reusage of knowledge modules. These knowledge modules are representants for assets like devices, machines and plants. The article encourages manufacturers of such assets to provide diagnosis knowledge bases by using a proposed multi-paradigm rule definition language called HLD (Hybrid Logic Description).
Rule based knowledge may be expressed by using various methodologies, which differ in expressiveness but also in runtime performance. The HLD allowes rules to be defined as propositional logic with or without the use of certainty factors, as Fuzzy Logic or as probabilistic rules as in Bayesian Networks. The most effective rule type may be chosen to describe causal dependencies between symptoms and failures. An evaluation prototype implementation has been developed in the research project WISA, which includes a sotftware tool chain for handling HLD files.

1 INTRODUCTION

A fundamental goal of industrial companies is to optimize the use of their production equipment, which consists of simple assets like devices (sensors, actuators, etc.), assembled machines and even more complex assets like production lines and plants. The term 'optimization' relates here to maximization of the asset performance and to minimization of production down-times as effects of failures.

This article is dedicated to the maximization of the equipment availability. Industrial companies prefer the use of reliable equipment over managing failures. *Failure Mode and Effects Analysis* (*FMEA*) or *Failure Mode, Effects and Criticality Analysis* (*FMECA*) may be used for engineering of robust production systems. But the production equipment may fail even in the case of a careful design. Thus different maintenance strategies have been developed in order to increase the availability of the assets:

- *Predictive Maintenance.* Assets are continuously monitored to get information on the abrasion state. The remaining asset life time is predicted in order to schedule necessary maintenance activities like repairing or asset exchanges.
- *Preventive Maintenance.* Maintenance activities are scheduled according to the machine run time and absolute machine life time. Time based abrasion models are used.
- *Reactive Maintenance.* The equipment is repaired in cases of failures according.

Traditionally a mixture of preventive and reactive strategies is used. Predictive and preventive maintenance may only be complementary elements to the reactive maintenance, since they left a remaining risk. This article emphases the optimization of reactive maintenance by

- decreasing the diagnosis time to find the failure causes and the repair information,
- integrate diagnosis results into the IT supported planning workflows of the companies.

An approved approach to increase diagnosis efficiency is the use of expert systems as shown in figure1. It contains a knowledge acquisition component, which is the interface for the expert. He may input new information about the respective knowledge domain into the knowledge base. Expert systems guide the user with questions to input only most

relevant symptom information for determine failure causes.

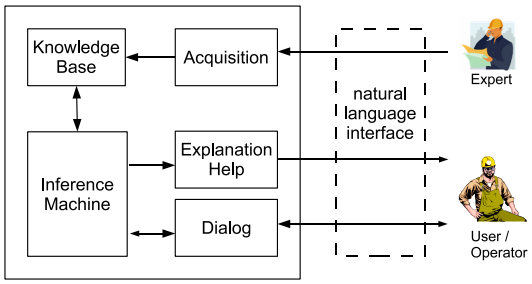


Figure 1: Structure of an expert system.

According to Moore's Law and its derivatives computer performance and memory availability doubles roughly every two years. Thus some handicaps of expert systems, regarding space and time complexity of applied algorithms, are reduced by time.

The WISA project focuses one of the remaining problems, which is the reduction of necessary effort of the experts to establish and maintain the knowledge base. This problem avoids the broad assignment of knowledge based systems in the industry.

The WISA project is funded by the German Ministry of Economy and Employment. It is registered under reg.-no. IW060215.

2 WISA SYSTEM DESCRIPTION

2.1 Standards based Knowledge Integration

Key factors of decreasing the effort for the creation and maintenance of knowledge bases are to reuse formalized knowledge and to spread the work among different experts. Another one is the formalization of the knowledge by the creators of the equipment, since they already own the necessary expertise.

Thus there is a need for a language, which makes it possible to describe knowledge bases. The complexity of this language should be small and reflect the needs of the application context (diagnosis) only. In any case it should be a standardized language to become relevant for the industry.

There are different approaches to establish a standard. The WISA project aims to be part of the following approach: *problem definition* → *finding solution ideas* → *proof of concept* → *publication* → foundation of a standardization consortium → redefinition of the proposals → reference implementation → finalization of the standard. The project leads up to the publications, which are necessary to involve candidates for a standardization consortium.

The subject for standardization is a language for description of industrial equipment. It is called Hybrid Logic Description (HLD). Design goals coming from the application domain of intelligent diagnosis where:

- handling of logic and causal dependencies between failure symptoms and their causes,
- handling of uncertain assertions about these dependencies,
- handling of vague information about the symptoms and their causes.

The HLD design includes propositional logic in combination with certainty factors, Fuzzy Logic and Bayesian networks.

2.2 Language Features of the HLD

2.2.1 Logical Description

A natural kind of causal reasoning is propositional logic. A simple sentence for diagnosis purposes could be

IF symptom1 AND symptom2 THEN cause1

The knowledge base of a propositional logic system would consist of a bunch of such rules. Variables contain truth values therein. Operators like NOT, AND, OR, IMPLICATION, BICONDITIONAL may be used in the rule sets.

Diagnosis for a broad variety of devices may be done by using propositional logic. Even good cars or TVs ship with a manual containing chapter for first aid diagnosis in a kind of propositional logic descriptions. A look into these documents shows that there is seldom a need for operators of first-order or higher-order logic like ALL- or EXISTENCE operators. Thus the WISA project focused on propositional logic for handling binary variables.

Certainty Factors are extensions to propositional logic. They were developed in the 1970s for medical diagnosis expert systems, see for example (Russell and Norvig, 2003) or (Norvig, 2004). The principal idea is that the same observable symptomatic may have different causes but with different certainty. Thus the conclusion of a rule is weighted by using a *certainty factor* (*CF*).

Certainty factors are born by empirical development. The value range of a *CF* is part of the interval $[-1,+1]$, where selected meanings for the variable value are:

```
CF = +1     => true
CF =  0     => unknown
CF = -1     => false
```

Disjunctive and conjunctive combinations of conditions are allowed in logic systems with certainty factors. A variable may be the conclusion in different rules. Thus there exists a combination calculus for merging different conclusion results.

One problem with certainty factors is that the memory space, which is necessary for the computation of the conclusion, grows with complexity $O(c^n)$ because of a necessary breadth-first search. Therein c is an approximation of the average count of conditions in the rules, whereas n is the number of rules, which are involved in the recursive computation of the conclusion. Another problem is the sloppy handling of certainties for maximization of efficiency. Bayesian Networks are additionally included into the HLD, which avoid these problems but expect input of probabilities from the expert, which are difficult to estimate.

2.2.2 Fuzzy Logic

Propositional logic deals perfectly with truth values. But in industrial plants there are symptoms and failures with continuous value domains, like temperatures, pressures, speed and other physical and chemical phenomenon.

Fuzzy Logic is a better way to deal with this kind of information than handling truth variables, which represent special value ranges. Continuous value variables are handled as linguistic variables in Fuzzy Logic. Attributes (with a representation as fuzzy set) are spread over ranges of the full variable value space. Attributes are characterized by membership functions, which map a continuous space $[0..1]$ over the related value range. This allows a more accurate reasoning in Fuzzy Logic than in propositional logic with discretized values.

An interesting point is that the notations of rules in Fuzzy Logic are very similar to the notations in propositional logic. Thus experts have to rearrange their thinking not too much when using both.

But the inference processes in Fuzzy Logic is expensive. It is done in a *fuzzyfication - inference - defuzzyfication* approach. While fuzzyfication is a straight forward computation of the membership value, inference and defuzzyfication are time consuming operations. Thus Fuzzy Logic should only be used in diagnosis scenarios where really necessary.

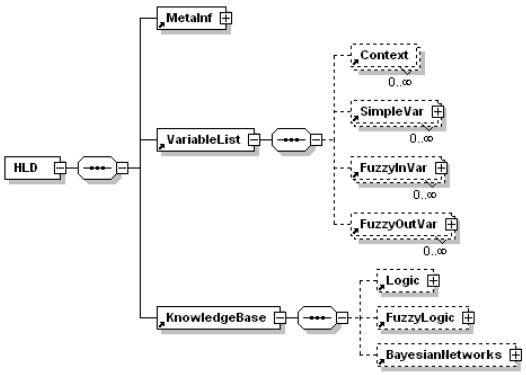


Figure 2: High level elements of the HLD specification.

2.2.3 Bayesian Networks

Bayesian Networks deal with event probabilities. They consist of directed acyclic graphs. The nodes consist of random variables arcs notify dependencies. An event is defined in a manner, that a random variable fits a specific state. According to (Jensen, 1996) there is then the possibility to declare rules in the form:

IF event A has taken place, THEN B has taken place with certainty x.

In a diagnosis applications event B may be an observed symptom, while A may be a cause. The probability of B is computable by using the Bayes' rule:

$$P(B|A,C) = \frac{P(A|B,C)P(B|C)}{P(A|C)}. \qquad (1)$$

The variable C is a context variable therein. Noteworthy is that the Bayesian Network theory stands completely on a solid mathematical fundament, see (Pearl, 2000) for example. There is no approximation needed like in the Certainty Factor theory. Pearl showed, that Bayesian Networks admit d-separation. The usage of this circumstance leads to very efficient computations of the conditional probabilities in Bayesian Networks.

2.3 Language Formalization

XML has been chosen as base format for the HLD. Some requirements where:

- integrated UNICODE support,
- automatic support for a validation of instance documents,
- broad acceptance in industry.

XML is used for example as message transport format for engineering and commissioning tools (e.g. by the FDT Group and OPC Foundation), for maintenance

and application integration purposes (e.g. by the MIMOSA Alliance) and for formal device descriptions (e.g. by PROFIBUS International).

Figure 2 shows the base levels of the XML-Schema based definition of the HLD. Meta information (MetaInf) are used to describe an equipment type verbally. The other elements contain the formal description of the algorithmic HLD elements as introduced in the previous sections.

2.4 Integration Concept

A machine or plant is mostly an assembly of equipment of different vendors. Thus the diagnosis descriptions have to integrate and extend descriptions of aggregated equipments. Figure 3 gives an abstract overview of assembled equipment element as considered in HLD descriptions.

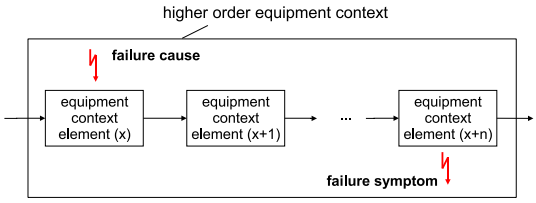


Figure 3: High level elements of the HLD specification.

An equipment is described by a single HLD file. If it contains other types of equipment then these are referenced as contexts (see also figure 2). It is possible to define rules or Bayesian Networks, which contain variables of the aggregated equipment contexts since they are referenced by context paths to variables.

2.5 Wisas - A HLD Tool Chain

It is necessary to show, that the approach leads to an efficient diagnosis, in order to gain attention and acceptance by industrial companies. But it is also necessary to show how various tools may downsize the effort for creation and maintenance of the knowledge bases. A tool chain is built in the WISA project, which is called WISAS and which consists of following components:

- *Editor*. Involved industrial partners showed that it is necessary to leave the vocabulary of AI technologies but to use the vocabulary of maintainers. This language transformation is done by a graphical editor component.
- *Validator*. This component is necessary to support creation of correct HLD files.
- *Interpreter*. This is the reasoning component for interpretation of HLD files.
- *Documentation generator*. It is assumed that industrial companies want to provide written documents for diagnosis. This component is dedicated to reach that goal.
- *Package management*. There is a need to deliver a bunch of HLD files for description of higher order equipment. There may be complex dependencies, which are handled by a package build tool and a HLD repository maintenance tool, which may use Internet connections.
- *Planning system integration*. The last but yet unfinished part of the project is to define adapters to ERP and CMMS systems.

3 CONCLUSIONS

The initial problem to reduce the effort for creation of knowledge bases for intelligent industrial diagnosis systems has been solved by a modularization of the knowledge bases. It has been shown that the industrial relevance will only be reached if there will be a standardization process. Therefore the project WISA proposes a knowledge description language for the special purpose of industrial diagnosis. A prototypical tool chain has additionally been developed in order to find more acceptance at industrial companies.

There have been critical discussions about the approach especially if XML is the right base technology. Some advantages as the possibilities to validate and transform XML files has been higher rated than the disadvantage of poor parsing performance compared to special language parsers.

There are some open questions regarding time and space complexity. These questions will be answered until the end of the project in August 2008. Implementation of future HLD interpreters and editors will benefit from using on-line data, which could be used for automatic triggering a diagnosis process but also for learning procedures for the Bayesian Networks.

REFERENCES

Jensen, F. V. (1996). *An introduction to Bayesian networks*. UCL Press Limited, London, 1st edition edition.

Norvig, P. (2004). *Paradigms of artificial intelligence programming: case studies in Common Lisp*.

Pearl, J. (2000). *CAUSALITY Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 1st edition edition.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition.

AN EFFICIENT INFORMATION EXCHANGE STRATEGY IN A DISTRIBUTED COMPUTING SYSTEM
Application to the CARP

Kamel Belkhelladi[1,2], Pierre Chauvet[1,2] and Arnaud Schaal[2]
[1]*LISA, Université d'Angers, 62, avenue Notre Dame du Lac, 49000 Angers, France*
kamel.belkhelladi@uco.fr, pierre.chauvet@uco.fr
[2]*CREAM, Université Catholique de l'Ouest, 44-46, rue Rabelais, 49008 Angers, France*
arnaud.schaal@stjodijon.com

Keywords: Capacitated Arc Routing Problem, Information exchange strategy, Distributed computing, Agents.

Abstract: Distributed computation models have been widely used to enhance the performance of traditional evolutionary algorithms, and have been implemented on parallel computers to speed up the computation. In this paper, we introduce a multi-agent model conceived as a conceptual and practical framework for distributed genetic algorithms used both to reduce execution time and get closer to optimal solutions. Instead of using expensive parallel computing facilities, our distributed model is implemented on easily available networked personal computers (PCs). In order to show that the parallel co-evolution of different sub-populations may lead to an efficient search strategy, we design a new information exchange strategy based on different dynamic migration window methods and a selective migration model. To evaluate the proposed approach, different kinds of experiments have been conducted on an extended set of Capacitated Arc Routing Problem(CARP). Obtained results are useful for optimization practitioners and show the efficiency of our approach.

1 INTRODUCTION

Genetic Algorithms, a special class of Evolutionary Algorithms (EAs), are proven to be successful in a wild range of applications, especially in cases of optimization. In order to solve more difficult problems, two inherent features of EAs, *premature convergence* and *computation time* must be improved. To overcome the two problems, the idea of parallelizing EAs has been proposed by different researchers, and it has been proven to be a promising method (Cantù-Paz, 2000). Conceptually, the parallelism is to divide a big population in a sequential EA into multiple smaller sub-populations that are distributed to separate processors and can then be evaluated simultaneously. According to the sub-population size, the parallel EAs are categorized into two types: *coarse-grain* and *fine-grain*, and they are usually implemented on distributed memory and massively parallel computers, respectively. All though parallel computers can speed up EAs, they are not easily available and it is very expensive to upgrade their processing power and memory. A promising alternative without expensive hardware facilities is to construct the parallel evolutionary computation (EC) framework on a set of networked personal computers.

To make the parallel EAs more realistic, we propose in this paper a mobile agent-based methodology to support parallelism on networked computers in a flexible and adaptive way. In addition, we design a new information exchange strategy. The idea is that an EA periodically selects some promising individuals from each sub-population and sends them to other sub-populations, according to certain criteria. To evaluate the proposed approach, different kinds of experiments have been conducted on an extended set of Capacitated Arc Routing Problem (CARP) and the obtained results show the promise and efficiency of our approach.

2 THE CARP

The Capacitated Arc Routing Problem (CARP) is a vehicle routing problem raised by applications like urban waste collection, snow plowing, sweeping, gritting, etc. It is defined on an undirected network G=(V,E), with a set V of n nodes and a set E of m edges. A fleet of identical vehicles with capacity W is based at a depot node s. Each edge e can be traversed

any number of times, each time with a cost c_e, and has a non-negative demand q_e. All costs and demands are integers. The τ edges with non-zero demands are called required edges or tasks and require service by a vehicle. The goal is to determine a set of vehicle trips of minimum total cost, such that each trip starts and ends at the depot, each required edge is serviced by one single trip, and the total demand handled by any vehicle does not exceed *W*.

Since the CARP is *NP-hard*, large scale instances must be solved in practice using heuristics. Among fast constructive methods, one can cite Path-Scanning (Golden et al., 1983) and Ulusoy's splitting technique (Ulusoy, 1985). Available meta-heuristics are very recent and include tabu search methods (Belenguer and Benavent, 2003; Hertz et al., 2000), guided local search (Beullens et al., 2003) and genetic algorithm (Lacomme et al., 2001). All these heuristics algorithms can be evaluated through lower bounds (Amberg and Voß, 2002).

3 AGENT-BASED APPROACH

The DGA-MAS[1] developed in this work uses several components of the genetic algorithm algorithm proposed by (Lacomme et al., 2001) for the CARP. The common parts are described below:

Solution Encoding: The network is coded as a symmetric digraph, in which each edge is replaced by two opposite arcs. A chromosome is an ordered list of the τ tasks, in which each task may appear as one of two directions. The implicit shortest paths are assumed between successive tasks. The chromosome does not include trip delimiters and can be viewed as a giant trip for an incapacitated vehicle. A procedure *Split* optimally partitions (subject to the sequence) the giant trip into feasible trips. The fitness function of the genetic algorithm (GA) is the total cost of the resulting CARP solution.

Initialization: The global population *P* of chromosomes is initialized with the solutions of the two CARP heuristics cited in introduction (PS and UH)(Golden et al., 1983; Ulusoy, 1985), completed by random permutations. The distributed genetic algorithm (DGA) forbids clones (identical chromosomes).

Selection and Crossover: At each iteration, two parents are selected by a biased roulette wheel (Goldberg, 1989). Three crossovers were defined for the giant trip representation, LOX (Linear Order Crossover), a modified version of the classical order crossover OX and X1 (crossing in a point).

Mutation: Several mutation operators have been proposed for the CARP, such as inversion, insertion, displacement (MOVE), and reciprocal exchange mutation (SWAP). Reciprocal exchange mutation selects two positions at random and swaps the tasks on these positions.

Several researchers are trying to compare the performance of using *coarse-grain* and *fine-grain* models to parallelize genetic algorithms. Some prefer *fine-grain* models but others favor *coarse-grain* ones (Cantù-Paz, 2000). As our goal is to develop a distributed genetic framework without using a powerful connection machine, we choose to implement a *coarse-grain* model on a set of networked PCs. By using the concept of *coarse-grained* parallelization, the population is divided into a few large subpopulations. These sub-populations evolve independently and concurrently on different processors. After a predefined period of time, some selected individuals are exchanged via a migration process.

Our distributed genetic algorithms with multiple sub-populations uses the master-slave (Luque et al., 2005) multi-agent model for the CARP. The interconnection topology is logically a star, with the master in the center. Our new migration process is based on different dynamic migration window methods (Kim, 2002) and the selective migration model (Eldos, 2006).

In the master-slave multi-agent model, the master agent maintains the lists of partial results that are sent from slaves. At first, the master agent generates the population and divides them into sub-populations. It sends them to slave agents. The slaves execute a conventional genetic algorithm on their sub-population: fitness evaluation, selection, crossover, mutation and periodically return their best partial results to the master agent. Pseudo-code of the conventional genetic algorithm is shown in section 3.1. The subroutine *Migration* is an extension to a general *coarse-grained* model. The master stores the partial results in lists. Then, the master agent searches the lists and selects two slaves that have bad partial results and sends the migration window size to selected slaves. The selected slaves exchange individuals according to the migration window methods. However, in the proposed model, individuals are screened and examined at both the source and the destination to qualify for migration. The source gives or denies a visa based on local qualification criteria and the destination grants or denies a residency based on local qualification criteria. The simplest form of qualification criteria is through the individual's relative fitness. Every individual in a very sub-population is ranked locally;

[1]Distributed Genetic Algorithm using Multi-agent System

an individual qualifies for a visa at the source sub-population if its rank is within a range, typically the middle class. On the destination sub-population, an immigrant is accepted as a new member of the population if its rank is better than a threshold set by that receiving sub-population.

3.1 The Algorithm

The Pseudo-code of the algorithm in each slave Agent is described as below:

```
initialize the population, P
while generation < max_generation begin
    evaluate P
    select P1′ from P using roulette
        wheel selection
    select parents randomly from P1′
    apply genetic operators to create
        the rest of the new population, P2′
    merge P1′ and P2′ to P′
    replace P with P′
    if an interval of K generations is reached
        Migration
    end
    generation = generation + 1
end
subroutine Migration
begin
    Pick a random number (1 - pop_size)
    to nominate
        an immigrant X
    If the rank of X is within limits then
        Send X to the Master
        Mark X as "dead" to be killed
    end
    If an immigrant Y is received then
        If the rank of Y exceeds threshold
            Add to the population
            Update Fitness and Ranking
            Pick a random number (1 - pop_size)
                to nominate a victim V
            Discard the victim V
        end
    end
end
```

4 EXPERIMENTS AND RESULTS

Following our computational experiments and the agent's methodology, we conducted two series of experiments to compare the corresponding performance for genetic computation with and without individual's exchange. The first series examines whether the parallel implementation can improve search quality. The second series of experiments evaluates the performance of using Multi-agent and the individuals exchange strategies in order to exploit the computational power of multiple machines.

Both the pure Parallel Genetic Algorithm (PGA) and the Distributed Genetic Algorithm (DGA-MAS), including the multi-agent strategies, are implemented on a network of PCs, connected with a 100Mbit/sec Ethernet. In the experiments, we arranged eight networked computers running Suse Linux operating system as a distributed computing environment to support our Multi-agent system.

One computer played the role of "master" and ran the multi-agent platform. Once the platform was activated, the master agent instantiated other agents (slave agents) and sent each one towards a machine in the network. Each machine in this framework had a slave agent to take care of the computation for each sub-population. Also, the master agent allowed activation of the communication phase described in Section 3 for exchanging individuals among sub-populations.

In the experiments, we use three crossover types (LOX, OX, X1) and two mutation types (MOVE, SWAP). Seven values for crossover rate are used ranging from 0.2 to 0.7 in increments of 0.1. Also, seven mutation rates are allowed varying from 0.1 to 0.4 in increments of 0.05. Our results are obtained with small sub-populations of 30 solutions. Clones (identical solutions) are forbidden in each sub-population, to have a better dispersion of solutions and to diminish the risk of premature convergence. The number of generations is fixed to 5000 for all slave agents. The migration interval is 500 generations. Whenever migration occurs, the dynamic migration window size varies at random from 1 to θ. The θ value is generated at random within 20% of the sub-population size. The threshold value is fixed to the mean fitness of the sub-population.

These tests are done on a standard set of undirected instances in which all edges are required. Table 1 contain 23 instances from (DeArmon, 1981) with 7 to 27 nodes and 11 to 55 edges. All these files can be obtained at http://www.uv.es/∼belengue/carp.html.

In the Table 1, *Pb* gives the instance number and *N*,*M* the numbers of nodes and edges. *LBB* is a lower bound from (Belenguer and Benavent, 2003). *TS* is the result of Carpet (Hertz et al., 2000) with the parameter setting yielding the best results on average (the same setting for all instances). *Best* gives the best solution published, generally obtained by Carpet with various parameter settings. *GA* is the solution of GA from (Lacomme et al., 2001). Our results are shown in the pure parallel genetic algorithm (*PGA*) and the distributed genetic algorithm including the Multi-agent strategy *(DGA)* columns.

Table 1: Results of the DGA-MAS on DeArmon's instances vs best known results.

Pb	N	M	LBB	Best	TS	GA	PGA	DGA
1	12	22	316	316	316	316	316	316
2	12	26	339	339	339	339	339	339
3	12	22	275	275	275	275	275	275
4	11	19	287	287	287	287	287	287
5	13	26	377	377	377	377	377	377
6	12	22	298	298	298	298	298	298
7	12	22	325	325	325	325	325	325
8	27	46	344	348	352	350	355	344
9	27	51	303	311	317	303	323	305
10	12	25	275	275	275	275	275	275
11	22	45	395	395	395	395	403	395
12	13	23	448	458	458	458	462	452
13	10	28	536	544	544	540	544	540
14	7	21	100	100	100	100	100	100
15	7	21	58	58	58	58	58	58
16	8	28	127	127	127	127	129	127
17	8	28	91	91	91	91	91	91
18	9	36	164	164	164	164	164	164
19	11	11	55	55	55	55	58	55
20	11	22	121	121	121	121	123	121
21	11	33	156	156	156	156	158	157
22	11	44	200	200	200	200	203	200
23	11	55	233	233	233	235	237	233

5 DISCUSSION

We can observe that *DGA-MAS* obtained solutions as good as our *PGA*. Furthermore, within 48% of the solved problems, *DGA-MAS* outperformed our *PGA* implementation of evolutionary system by a mean advantage of 0.98% in term of solution cost. *DGA-MAS* is very efficient: on all instances, it is at least as good as Carpet. On the 23 DeArmon's instances, it outperforms Carpet 4 times, improves 4 best known solutions with one to optimality, and reaches *LBB* 20 times. The average deviation to *LBB* is roughly divided by 3 compared to Carpet and becomes 0.19%. Moreover, the mean speed up factor for the execution times between *DGA-MAS* and *PGA* is approximately 1.7, in spite of overhead communication time spent in *DGA-MAS*. Hence, the mean execution time spent to find a solution using *PGA* and *DGA-MAS* are respectively 40.3 and 24.5 seconds.

6 CONCLUSIONS

In this paper, we have proposed *DGA-MAS*, an efficient distributed genetic algorithm which combines the two advantages of parallelism: the computational power and co-evolution. We implemented an information exchange strategy based on the dynamic migration window methods that control the size and the frequency of migration and the selective migration model for the choice of individuals to migrate. Results confirm the positive impact of using MAS[2] strategy in regard to the pure parallel GA. They also point out that such strategies are, at a minimum, as good as the best known methods.

In future work, we will improve the performance of our information exchange system by the addition of intelligent modules, thus allowing the study of the exchange semantics of the requests for improving information. We will test also our framework on other *NP-complete* problems such as planning and scheduling problems.

REFERENCES

Amberg, A. and Voß, S. (2002). A hierarchical relaxations lower bound for the capacitated arc routing problem. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3*, pages 83–84, Washington, DC, USA. IEEE Computer Society.

Belenguer, J. and Benavent, E. (2003). A cutting plane algorithm for the capacitated arc routing problem. *Comput. Oper. Res.*, 30(5):705–728.

Beullens, P., Muyldermans, L., Cattrysse, D., and Oudheusden, D. (2003). A guided local search heuristic for the capacitated arc routing problem. *European Journal of Operational Research*, 127(3):629–643.

Cantù-Paz, E. (2000). *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

DeArmon, J. (1981). *A Comparison of Heuristics for the Capacitated Chinese Postman Problem*. Master's Thesis. The University of Maryland at College Park, MD, USA.

Eldos, T. (2006). A new migration model for distributed genetic algorithms. In *The International Conference On Scientific Computing (CSC)*, pages 128–134.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Golden, B., DeArmon, J., and Baker, E. (1983). Computational experiments with algorithms for a class of routing problems. *Computers and Operations Research*, 10(1):47–59.

Hertz, A., Laporte, G., and Mittaz, M. (2000). A tabu search heuristic for the capacitated arc routing problem. *Oper. Res.*, 48(1):129–135.

[2]Multi-Agent System

Kim, J. (2002). Distributed genetic algorithm with multiple populations using multi-agent. In *ISHPC'02: Proceedings of the 4th International Symposium on High Performance Computing*, pages 329–334, London, UK. Springer-Verlag.

Lacomme, P., Prins, C., and Ramdane-Chérif, W. (2001). A genetic algorithm for the capacitated arc routing problem and its extensions. In *Proceedings of the EvoWorkshops on Applications of Evolutionary Computing*, pages 473–483, London, UK. Springer-Verlag.

Luque, G., Alba, E., and Dorronsoro, B. (2005). *Parallel Genetic Algorithms , E. Alba (ed.), Parallel Metaheuristics: A New Class of Algorithms*. John Wiley.

Ulusoy, G. (1985). The fleet size and mix problem for capacitated arc routing. *European Journal of Operational Research*, 22(3):329–337.

A REAL TIME EXPERT SYSTEM FOR FAULTS IDENTIFICATION IN ROTARY RAILCAR DUMPERS

Osevaldo S. Farias, Jorge H. M. Santos, João V. F. Neto, Sofiane Labidi, Thiago Drumond
Department of Electrical Engineering, Federal University of Maranhão
Av. dos Portugueses, s/n, São Luís, Brazil
{osevaldo, jjhsantos}@yahoo.com.br, jviana@dee.ufma.br, labidi@uol.com.br, thiago_dr@yahoo.com.br

José Pinheiro de Moura, Simone C. F. Neves
Companhia Vale do Rio Doce, Retorno do Itaqui, SN, São Luís, Ma, Brazil
pinheiro.moura@cvrd.com.br, simone.neves@cvrd.com.br

Keywords: Expert System, Knowledge Base, Industrial Process, Rotary Railcar Dumper.

Abstract: This paper describes the development of a real-time Expert System applied to the ore extraction Industrial branch, specifically used to assist the decision making and fault identification on rotary railcar dumpers of the operational productive system located at Ponta da Madeira Dock Terminal, built and operated by Companhia Vale do Rio Doce (now referred just as VALE) in São Luis-MA. The Expert System is built on JESS (Java Expert System Shell) platform and provides support to engineers and operators during the ore unloading as soon as supplying on-line information about faults triggered by device sensors of the rotary railcar. The system's conception involves the application of CommonKADS methodology, knowledge engineering and artificial intelligence techniques at the symbolic level for representing and organizing the knowledge domain in which the system is applied.

1 INTRODUCTION

In a great deal of industrial production mechanisms approaches able to turn automatic a wide range of processes have being used. Such applications demand from high control pattern to decision taking, where Artificial Intelligence (AI) presents wide applicability of those approaches implementing their concepts under the form of Expert Systems (Su et al., 2005). Applications with this architecture allow the machine to be structured into a model apt to behave in the most similar way a human specialist uses its reasoning when facing a decision taken problem (Feigenbaum, 1992).

The VALE production system comprehends several mining complexes, among which is notorious the Ponta da Madeira Dock Terminal (PMDT). In this complex, processes such as Unloading, Storing and Minerals Shipping are conducted, supervised by the Operational Control Center (OCC). This article discusses the development of a real time Expert System applied to decision making when facing faults occurred in the VV311-K01 rotary railcar dumper system used to unload minerals at the VALE's PMDT. Besides, we apply some information technologies such as: the JESS, the JAVA language and also XML (eXtensible Markup Language) aiming the real time running of the Expert System.

This paper is organized as follows: Section 2 describes the particularities and the operation of the rotary railcar dumper system. In Section 3 are detailed the Expert System Development steps, such as knowledge acquisition, representation and system's implementation. Section 5 summarizes some remarks and the conclusion.

2 THE ROTARY RAILCAR DUMPER SYSTEM

The minerals unloading mechanism initiates at the rotary railcar dumper with the arrival of the locomotive pulling behind it 102 to 104 rail-wagons that will be positioned in the dumper.
To attain the rotation a positioner car fixes the rail-wagons in the rotary and this, consequently, unloads

the material by performing a 160° rotation, in the conveyor belts (Fonseca Neto et al., 2003). The Operational Process is monitored by the Supervisory Control and Data Acquisition (SCADA). This supervision is also conducted by means of the programmable logic controllers (PLCs) which receive all the information from the dumper hardware through input cards.

Thus, the rail-wagons dumper's hardware is one important middleware for the communication between the Expert System and the VV311-K01 hydraulic and mechanical components at the operation time.

3 THE EXPERT SYSTEM DEVELOPMENT

In order to develop the Expert System we highlighted its stages based on JESS and CommonKADS methodologies.

The JESS architecture involves cognition components defined as: Inference Engine, Agenda and Execution Engine. All these structures catch assertions or domain facts and also create new assertions. The inference JESS engine (based on the Rete algorithm) is constituted by the Pattern-Matching mechanism that decides which rules will be activated. The Agenda programs the order in which the activated rules will be fired, and the Execution Engine is in charge of the triggering shot (Friedman-Hill, 2003). In JESS the reasoning formalism presents rules composed by *if...then* patterns, represented by the LHS (Left-Hand Side) and RHS (Right-Hand Side), respectively.

CommonKADS is a methodology for building knowledge based systems (Labidi, 1997). Products arisen from Expert Systems development that use this methodology are the result of the performed phases modelling activities, and characterize the input artefacts for the successive refinements undergone in the next steps of the CommonKADS life cycling.

The steps of the system with actions such as Acquisition and Knowledge representation are summarized– also including the Analysis phase– Rules representation– attending the Design phase – and the System's codification – satisfying the phase Implementation.

3.1 Acquisition and Knowledge Representation

All the knowledge acquisition was done by means of interviews with the expert through information kept in the operational reports. The knowledge representation was built based upon production rules that map the knowledge of the VV311-K01 operation expert.

There were observed the main concepts related with the dumper's positioner car along activities in the operational productive system, aiming at getting knowledge elements description to elaborate the organizational model that complements the CommonKADS.

The domain facts deal with the equipment situation and the potential causes that promote the main system stopping or the reduction of its productivity. Thus, by correlating problems and opportunities that can be solved or enhanced by the Expert System from which there were extracted the identified slots for building the VV311-K01 templates, it was elaborated the organizational model presented in Table 1.

Table 1: Organization Model.

SLOT	OPPORTUNITIES	PROBLEMS
Situation	Spin	Vibration
		Broken Rollers
		Lack of voltage
	Positioner car	Short-circuit
		Broken fixing screws
		Broken Counter-bolts

The slot called 'Situation' is one of the units for representing the knowledge in the JESS inference engine. The causes that lead the dumper to reach certain circumstances are pointers for guiding what must be done as to specify derivations that constitute a method for the positioner car problem resolution, and the strategies to attain this solution. These efforts are described through the knowledge model shown in Table 2.

Table 2: Knowledge Model.

SLOT	INFERENCE LEVEL	TASK LEVEL
Cause	Vibration	Motor basement snap
		Resonance
		Bend axle
	Short-circuit	Terminal out of order
		Low isolation
		Falling's wire material

According to Labidi (1997), the inference and task levels are layers that describe the expert Knowledge; thus, the model in Table 2 constitutes a set of knowledge instances on the VV311-K01 positioner car component. In Table 2, in order to better characterize the system's knowledge mechanism in agreement with the CommonKADS methodology, the activities organized in the inference model presented in Figures 1, were decomposed.

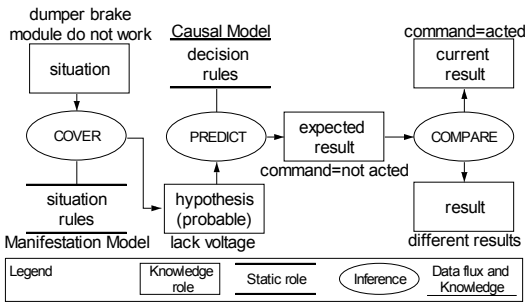


Figure 1: Inference Model.

The knowledge's roles, described in Figure 1, are functional names that capture the elements participant in the reasoning process. Inference actions assume as inputs static roles, represented by the manifestation and causal models. Within the causal model, the rules relate the positioner car fault modes taking into account their attribute's values, while in the manifestation model are reunited the production rules that express their responsibilities through the attributes' values, which satisfy some given conditions. The inference concepts represent reasoning axioms that will be mapped by the JESS inference engine used in the Expert System.

3.2 System Implementation

The Expert System decision module performs a scanning on the faults detected by the sensors present in the VV311-K01 instrumentation.

Through the checking of these faults addresses, which are generated by the PLCs and mapped into faults tagnames stored in the relational database, the Expert System rules are activated or not in the working memory of the JESS inference engine.

Tagnames (e.g. `AFF_CEP_F01@VV311K01`) are part of historical registry from PIMS (Plant Information Management System). In the Expert System they play the role of input data for the rules that deduce the situation the VV311-K01 components pass through, that is why they form the LHS rules pattern in the JESS language syntax, as can be seeing in the excerpt bellow:

```
(defrule rule198
(test(eq TRUE(actedTag
"AFF_CEP_F01@VV311K01")))
=>
(store RESULT198 "Loose-Wire-
Connection")
(assert (decision(decCausa Loose-Wire-
Connection))))
```

The 'actedTag' function returns true for 1 and false for 0, according to the result read in the XML file generated by the stored procedure into the PIMS oracle database server (see Figure 2). The function 'test' is a conditional JESS instruction responsible for determining that this pattern will only be unified if the result of the 'actedTag' function returns true. The part of XML document is shown as the following structure:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <tags>
 - <tagname id="ASC_B11@VV311K01">
    <value>0</value>
   </tagname>
 - <tagname id="AIN_ALI_001@VV311K01">
    <value>1</value>
   </tagname>
```

Aiming at present the system's global solution, Figure 2 describes the Expert System architecture and the nodes distribution that form its structure in order to summarize the Expert System solution.

Once the architecture is formulated, in the following parcel of the system's decision rule, it is possible to observe that the cause 'Loose-Wire-Connection' deduces that the ultrasound on VV311-K01 must be made.

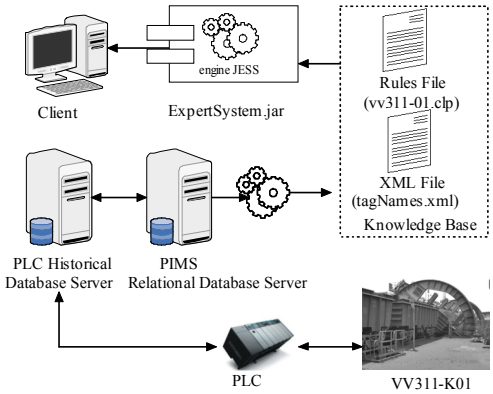


Figure 2: Expert System Architecture.

The RHS pattern of the rule's parcel below stores the result through the command store, and next inserts in the 'decision' template, the action to be done.

```
(defrule rule204
(decision (decCause Loose-Wire-
Connection))
 =>
(store RESULT204 "Do ultrasound on
VV311-K01")
(assert (decision
        (decDecision  Do-ultrasound-on-
VV311-K01))))
```

At the end of the performed deductions, in the shell JESS working memory is shown a window with the recommendations that were stored by the command store, based in the causes that led to the inference process pointed out by the system, along the responses given by the user. Figure 3 shows the decision taking delivered by the system.

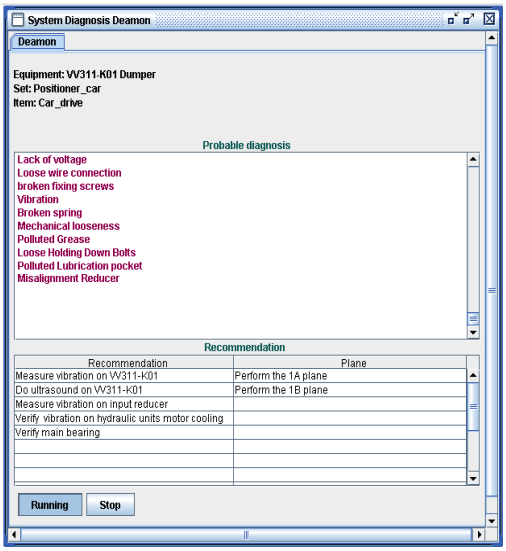


Figure 3: Expert System Recommendations.

The recommendations viewed in Figure 5, represent the rules that were activated in the working memory of the JESS engine and were triggered (shot) only because they got the unification of patterns present in its structure, deduced by the Rete algorithm. The control structure used for the Expert System rules chaining was based on backward chaining (Friedman-Hill, 2003).

4 CONCLUSIONS

The system developed in this work presented the conception and automation of the strategic knowledge required by VALE's mineral unloading system activities. The building of the Expert System in JESS, turned available the use of existing and well succeeded methods for the developing of systems based on knowledge, like the CommonKADS, and the direct handling of JAVA technology objects.

The system's performance while processing information in the JESS inference engine was considered satisfactory once the search frequency of such information elements was tested at 5 s. interval for the deduction of 250 rules, in order to update the expert system the events triggered by the PIMS server. The speed and proper timing obtained in terms of the updating processes of the expert system rules base was due to the use of the XML technology as to feed the system's knowledge base.

Finally, this system furnished enhancement and relative readiness to the knowledge processing, as a guide for the decision taking of the VALE's rail-wagon unloading system experts.

ACKNOWLEDGEMENTS

The authors express their thanks to FAPEMA/CAPES, Intelligent Systems Laboratory/UFMA, Automation and Control Laboratory/UFMA and VALE.

REFERENCES

Feigenbaum, A.E. Expert Systems: principles and practice. In: Encyclopedia of computer science and engineering, 1992.

Friedman-Hill, E. J. Jess in Action: Rule-based systems in java. USA: Manning Press, 2003.

Su, K.W. Hwang, S.L. Chou Y.F. Applying knowledge to the usable fault diagnosis assistance system: A case study of motorcycle maintenance in Taiwan. Elsevier Science Ltd, 2005.

Fonseca Neto, J. V. da., Moura, J.P. de. Um Sistema Especialista para Identificação de Falhas e Tomada de Decisão em Correias Transportadoras de Minério, VI SBAI, 2003.

Labidi, S. CommonKADS Extension for Supporting Multi-Expertise. In: The 17th International Conference of the British Computer Society on Expert Systems, (ES'97), 1997, Cambridge. Proceedings of the 17th, International Conference of the British Computer Society on Expert Systems, (ES'97), 1997.

NONLINEAR SYSTEM IDENTIFICATION USING DISCRETE-TIME NEURAL NETWORKS WITH STABLE LEARNING ALGORITHM

Talel Korkobi, Mohamed Djemel
Institute of Problem Solving, XYZ University, Intelligent Control, design & Optimization of complex Systems
National Engineering School of Sfax - ENIS, B.P. W, 3038 Sfax, Tunisia
korkobi_talel@yahoo.fr, mohamed.Djemel@enis.rnu.tn

Mohamed Chtourou
Intelligent Control, design & Optimization of complex Systems
National Engineering School of Sfax - ENIS, B.P. W, 3038 Sfax, Tunisia
mohamed.chtourou@enis.rnu.tn

Keywords: Stability, neural networks, identification, backpropagation algorithm, constrained learning rate, Lyapunov approach.

Abstract: This paper presents a stable neural sytem identification for nonlinear systems. An input output discrete time representation is considered. No a priori knowledge about the nonlinearities of the system is assumed. The proposed learning rule is the backpropagation algorithm under the condition that the learning rate belongs to a specified range defining the stability domain. Satisfying such condition, unstable phenomenon during the learning process is avoided. A Lyapunov analysis is made in order to extract the new updating formulation which contain a set of inequality constraints. In the constrained learning rate algorithm, the learning rate is updated at each iteration by an equation derived using the stability conditions. As a case study, identification of two discrete time systems is considered to demonstrate the effectiveness of the proposed algorithm. Simulation results concerning the considered systems are presented.

1 INTRODUCTION

The area of system identification has received significant attention over the past decades and now it is a fairly mature field with many powerful methods available at the disposal of control engineers. Online system identification methods to date are based on recursive methods such as least squares, for most systems that are expressed as linear in the parameters (LIP).
To overcome this LIP assumption, neural networks (NNs) are now employed for system identification since these networks learn complex mappings from a set of examples. Due to NN approximation properties as well as the inherent adaptation features of these networks, NN present a potentially appealing alternative to modeling of nonlinear systems.
Moreover, from a practical perspective, the massive parallelism and fast adaptability of NN implementations provide additional incentives for further investigation.
Several approaches have been presented for system identification without using NN and using NN (Narendra and Parthasarathy, 1990) (Boskovic and Narendra 1995). Most of the developments are done in continuous time due to the simplicity of deriving adaptation schemes. To the contrary, very few results are available for the system identification in discrete time using NNs. However, most of the schemes for system identification using NN have been demonstrated through empirical studies, or convergence of the output error is shown under ideal conditions (Ching-Hang Lee and al, 2002).
Others (Sadegh, 1993) have shown the stability of the overall system or convergence of the output error using linearity in the parameters assumption. Both recurrent and dynamic NN, in which the NN has its own dynamics, have been used for system identification.

Most identification schemes using either multilayer feedforward or recurrent NN include identifier structures which do not guarantee the boundedness of the identification error of the system under nonideal conditions even in the open-loop configuration.

Recent results show that neural network technique seems to be very effective to identify a broad category of complex nonlinear systems when complete model information cannot be obtained. Lyapunov approach can be used directly to obtain stable training algorithms for continuous-time neural networks (Ge, Hang, Lee, Zhang, 2001), (Kosmatopoulos, Polycarpou, Christodoulou, Ioannou, 1995) (Yu, Poznyak, Li, 2001). The stability of neural networks can be found in (Feng, Michel, 1999)and (Suykens, Vandewalle, De Moor, 1997). The stability of learning algorithms was discussed in (Jin, Gupta, 1999) and (Polycarpou, Ioannou 1992).

It is well known that normal identification algorithms are stable for ideal plants (Ioannou, Sun, 2004). In the presence of disturbances or unmodeled dynamics, these adaptive procedures can go to instability easily. The lack of robustness in parameters identification was demonstrated in (E. Barn, 1992) and became a hot issue in 1980s. Several robust modification techniques were proposed in (Ioannou, Sun, 2004). The weight adjusting algorithms of neural networks is a type of parameters identification, the normal gradient algorithm is stable when neural network model can match the nonlinear plant exactly (Polycarpou, Ioannou 1992). Generally, some modifications to the normal gradient algorithm or backpropagation should be applied, such that the learning process is stable. For example, in (L. Jin, M.M. Gupta, 1999) some hard restrictions were added in the learning law, in (Polycarpou, Ioannou 1992) the dynamic backpropagation was modified with stability constraints.

The paper is organized as follows. Section II describes the adopted identification scheme. In section III and through a stability analysis a constrained learning rate algorithm is proposed to provide stable adaptive updating process. two simple simulation examples give the effectiveness of the suggested algorithm in section VI.

2 PRELIMINARIES

The main concern of this section is to introduce the feedfarward neural network adopted architechture and some concepts of backpropagation training algorithm. Consider the following discrete-time input-output nonlinear system

$$y(k+1)=f\left[y(k).....y(k-n+1),u(k)....u(k-m+1)\right] \quad (1)$$

The neural model for the plant can be expressed as

$$\hat{y}(k+1)=F\left[Y(k),W,V\right] \quad (2)$$

Where $Y(k)=(y(k),y(k-1),....y(k-n+1),u(k)$

$$,u(k-1),.....,u(k-m+1))$$

and W and V is the weight parameter vector for the neural model.

A typical multilayer feedfarward neural network is shown in *Figure 1*, where I_i is the ith neuron input, Oj is the jth neuron output i, j and k indicate neurons, wij is the weight between neuron i and neuron j. For the ith neuron, the nonlinear active function is defined as

$$f(x)=\frac{1}{1+e^{-x}} \quad (3)$$

The output $y^m(k)$ of the considered NN is

$$\left.\begin{aligned} I_j&=\sum_{i=1}^{n+m+1}x_i w_{ij}\ ;\ O_j=f(I_j)\ ;j=1,\ldots,m \\ I'&=\sum_{j=1}^{N}O_j V_j\ ;\ y^m(k)=f(I')\ ;\ k=1,...,N \end{aligned}\right\} \quad (4)$$

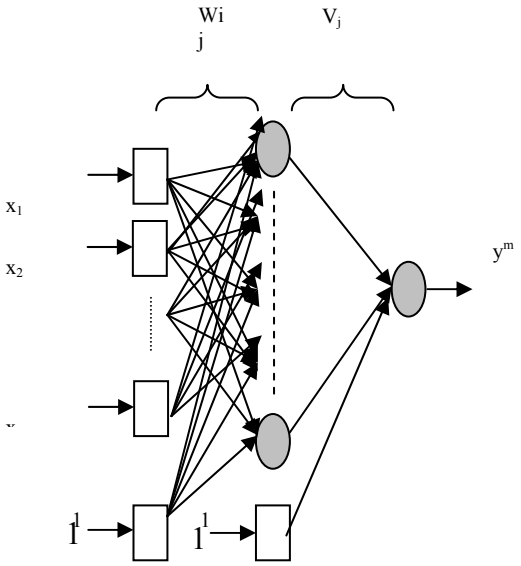


Figure 1: Feedforward neural model.

Training the neural model consists on the adjustment the weight parameters so that the neural model emulates the nonlinear plant dynamics. Input–output examples are obtained from the operation history of the plant.

Using the gradient decent, the weight connecting neuron *i* to neuron *j* is updated as

$$\begin{cases} W_{ij}(k+1) = W_{ij}(k) - \varepsilon \cdot \dfrac{\partial J(k)}{\partial W_{ij}(k)} \\ V_j(k+1) = V_j(k) - \varepsilon \cdot \dfrac{\partial J(k)}{\partial V_j(k)} \end{cases} \quad (5)$$

Where $J(k) = \frac{1}{2}\left[y(k+1) - y^m(k+1)\right]^2$

ε is the learning rate. The partial derivatives are calculated with respect to the vector of weights *W*.

$$\begin{cases} \dfrac{\partial J(k)}{\partial V_j(k)} = f'(I_j)(y(k+1) - y^m(k+1))O_j \\ \dfrac{\partial J(k)}{\partial w_{ij}(k)} = f'(I_j)\left[\sum_{j=1}^{L} f'(I_j)(y(k+1) - y^m(k+1))V_j\right]x_i \end{cases} \quad (6)$$

Backpropagation algorithm has become the most popular one for training of the multilayer perceptron . Generally, some modifications to the normal gradient algorithm or backpropagation should be applied, such that the learning process is stable. For example, in (B. Egardt, 1979) some hard restrictions were added in the learning law, in (J.A.K. Suykens, J. Vandewalle, B. De Moor, 1997) the dynamic backpropagation was modified with stability constraints.

3 STABILITY ANALYSIS

In the literature, the Lyapunov synthesis (Z.P. Jiang, Y. Wang, 2001), (W. Yu, X. Li, 2001) consists on the selection of a positive function candidate *V* which lead to the computation of an adaptation law ensuring it's decrescence **,** i.e $\dot{V} \le 0$ for continuous systems and $\Delta V(k) = V(k+1) - V(k) \le 0$ for discrete time systems. Under these assumptions the function *V* is called Lyapunov function and garantee the stability of the system. Our objective is the determination of a stabilizing adaptation law ensuring the stability of the identification scheme presented below and the boundness of the output signals. The following assumptions are made for system (1)

Assumption 1. The unknown nonlinear function *f(·)* is continuous and differentiable.

Assumption 2. System output y(k) can be measured and its initial values are assumed to remain in a compact set Ω_0 .

3.1 Theorem

The stability of the identification scheme is guaranteed for a learning rate verifying the following inequality :

$$0 \le \varepsilon \le \frac{2\left[tr\left(\dfrac{\partial J}{\partial W(k)} W^T(k) + \dfrac{\partial J}{\partial V(k)} V^T(k)\right)\right]}{\sum_{i,j}\left(\dfrac{\partial J}{\partial W_{ij}(k)}\right)^2 + \sum_{ij}\left(\dfrac{\partial J}{\partial V_j(k)}\right)^2} \quad (7)$$

Where *W, V* are respectivly the vector weight between the inputs and the hidden layer and the vector weight between the hidden layer and the outputs layer. *i* denote the ith input and *j* the jth hidden neuron.

3.2 Proof

Considering the Lyapunov function:

$$V_L(k) = tr\left(\tilde{W}^T(k)\tilde{W}(k)\right) + \tilde{V}^T(k)\tilde{V}(k) \quad (8)$$

Where

tr(.) denotes the matrix trace operation.

$\tilde{V}(k) = V(k) - V^*$

$\tilde{W}(k) = W(k) - W^*$

W^* denotes the optimal vector weight between the inputs and the hidden layer .

V^* denotes the optimal vector weight between the hidden layer and the outputs.

The computation of the $\Delta V_L(k)$ expression leads to :

$$\begin{aligned} \Delta V_L(k) = V(k+1) - V(k) = &\left[tr\left(\tilde{W}^T(k+1)\tilde{W}(k+1)\right) + \tilde{V}^T(k+1)\tilde{V}(k+1)\right] \\ &- \left[tr\left(\tilde{W}^T(k)\tilde{W}(k)\right) + \tilde{V}^T(k)\tilde{V}(k)\right] \end{aligned} \quad (9)$$

The adopted adaptation law is the gradient algorithm. We have:

$$\begin{cases} \tilde{W}(k+1) = W(k) - \varepsilon \dfrac{\partial J}{\partial W(k)} - W^* \\ \tilde{V}(k+1) = V(k) - \varepsilon \dfrac{\partial J}{\partial V(k)} - V^* \end{cases}$$

Where the partial derivatives are expressed as

$$\begin{cases} \dfrac{\partial J}{\partial W(k)} = \dfrac{\partial J}{\partial y(k+1)} \dfrac{\partial y(k+1)}{\partial W(k)} \\ \dfrac{\partial J}{\partial V(k)} = \dfrac{\partial J}{\partial y(k+1)} \dfrac{\partial y(k+1)}{\partial V(k)} \end{cases} \quad (10)$$

Our field of interest covers the black box systems. The partial derivatives denoting the system dynamic are approximated as follow:

$$\varepsilon_s=\frac{2\left[tr\left(\left\{\left[\left[y^m(k+1)(1-y^m(k+1)).e(k).\sum_{j=1}^{N}V_{j1}.O_j.(1-O_j)\right]x_i\right\}_{j\in[1\cdots m]}^{i\in[1\cdots n]}\cdot W^T(k)+\left\{\left[y^m(k+1)(1-y^m(k+1)).e(k).O_j\right]\right\}_{j\in[1\cdots m]}\cdot V^T\right.\right.}{\sum_{i,j}\left(\left[y^m(k+1)(1-y^m(k+1)).e(k).\sum_{j=1}^{N}V_{j1}.O_j.(1-O_j)\right]x_i\right)^2+\sum_j\left(\left[y^m(k+1)(1-y^m(k+1)).e(k).O_j\right]\right)^2}$$

$$\begin{cases}\dfrac{\partial y(k+1)}{\partial W(k)}\approx\dfrac{\partial y^m(k+1)}{\partial W(k)}\\[2ex]\dfrac{\partial y(k+1)}{\partial V(k)}\approx\dfrac{\partial y^m(k+1)}{\partial V(k)}\end{cases}\qquad(11)$$

The approximated partial derivatives are given through:

$$\begin{cases}\left[\dfrac{\partial J}{\partial W_{ij}(k)}\right]=\left[y^m(k+1)(1-y^m(k+1))e(k).\sum_{j=1}^{N}V(j,l).O_j.(1-O_j)\right]x_i\\[2ex]\left[\dfrac{\partial J}{\partial V_j(k)}\right]=\left[y^m(k+1)(1-y^m(k+1)).e(k).O_j\right]\end{cases}\qquad(12)$$

Adopting the variables A and B defined by:

$$A=tr\left(\tilde{W}^T(k+1)\tilde{W}(k+1)\right)-tr\left(\tilde{W}^T(k)\tilde{W}(k)\right)$$

$$B=\tilde{V}^T(k+1)\tilde{V}(k+1)-\tilde{V}^T(k)\tilde{V}(k)$$

The $\Delta V(k)$ expression i s calculated as :

$$\begin{aligned}\Delta V(k)&=A+B\\&=tr\left(\varepsilon^2\left[\frac{\partial J}{\partial W^T(k)}\frac{\partial J}{\partial W(k)}\right]-2\varepsilon\frac{\partial J}{\partial W(k)}\tilde{W}(k)\right)\\&\quad+\left(\varepsilon^2\left[\frac{\partial J}{\partial V^T(k)}\frac{\partial J}{\partial V(k)}\right]-2\varepsilon\frac{\partial J}{\partial V(k)}\tilde{V}(k)\right)\\&=\left(\varepsilon^2\sum_{i,j}\left[\frac{\partial J}{\partial W_{ij}(k)}\right]^2-2\varepsilon tr\left(\frac{\partial J}{\partial W(k)}\tilde{W}^T(k)\right)\right)\\&\quad+\left(\varepsilon^2\sum_{j}\left[\frac{\partial J}{\partial V_j(k)}\right]^2-2\varepsilon\frac{\partial J}{\partial V(k)}\tilde{V}^T(k)\right)\\&=\varepsilon^2\left(\sum_{i,j}\left[\frac{\partial J}{\partial W_{ij}(k)}\right]^2+\sum_{j}\left[\frac{\partial J}{\partial V_j(k)}\right]^2\right)\\&\quad-2\varepsilon\left(\frac{\partial J}{\partial V(k)}\tilde{V}^T(k)+tr\left(\frac{\partial J}{\partial W(k)}\tilde{W}^T(k)\right)\right)\\&\le\varepsilon^2\left(\sum_{i,j}\left[\frac{\partial J}{\partial W_{ij}(k)}\right]^2+\sum_{j}\left[\frac{\partial J}{\partial V_j(k)}\right]^2\right)\\&\quad-2\varepsilon\left(\frac{\partial J}{\partial V(k)}V^T(k)+tr\left(\frac{\partial J}{\partial W(k)}W^T(k)\right)\right)\\&\le\alpha\cdot\varepsilon^2-2\cdot\beta\cdot\varepsilon\end{aligned}$$

Where

$$\alpha=\sum_{i,j}\left[\frac{\partial J}{\partial W_{ij}(k)}\right]^2+\sum_{j}\left[\frac{\partial J}{\partial V_j(k)}\right]^2$$

$$\beta=\frac{\partial J}{\partial V(k)}V^T(k)+tr\left(\frac{\partial J}{\partial W(k)}W^T(k)\right)$$

The stability condition $\Delta V(k)\le 0$ is satisfied only if :

$$\alpha\cdot\varepsilon^2-2\cdot\beta\cdot\varepsilon\le 0\qquad(13)$$

Solving this ε second degree equation lead to the establishment of the condition (7) :

$\Delta V(k)\le 0$ if ε satisfies the following condition :

$$0\le\varepsilon\le\varepsilon_s$$

where

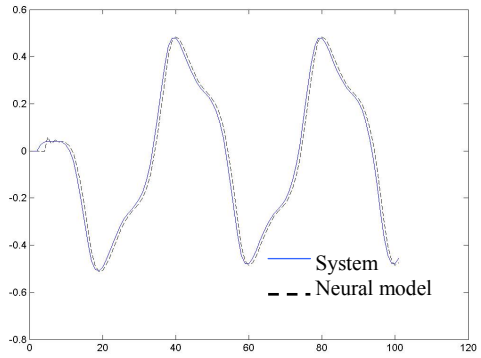


Figure 2: Evolution of the system output and the neural model output ($\varepsilon\in$ *stability domain*).

4 SIMULATION RESULTS

In this section two discrete time systems are considered to demonstrate the effectivness of the result discussed below.

4.1 First Order System

The considered system is a famous one in the litterature of adaptive neural control and identification. The discrete input-output equation is defined by:

$$y(k+1)=\frac{y(k)}{1+y(k)^2}+u(k)^3 \qquad (15)$$

For the neural model , a three-layer NN was selected with two inputs, three hidden and one output nodes. Sigmoidal activation functions were employed in all the nodes.
The weights are initialized to small random values. The learning rate is evaluated at each iteration through (14). It is also recognized that the training performs very well when the learning rate is small.
As input signal, a sinusoidal one is chosen which the expresion is defined by:

$$u(k)=0.5\cdot\cos\left(0.05k\pi+\frac{\pi}{5}\right) \qquad (16)$$

The simulations are realized in the two cases during 120 iterations. Two learnning rates values are fixed in and out of the learning rate range presented in (7).Simulation results are given through the following figures :

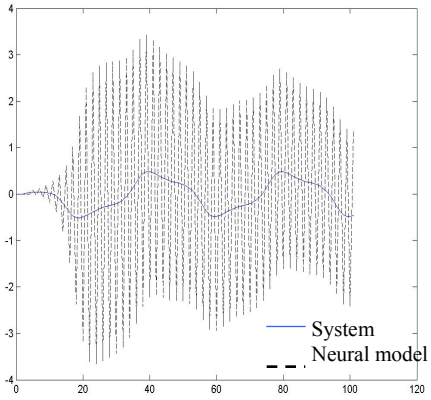


Figure 3 : Evolution of the system output and the neural model output ($\varepsilon \notin$ *stability domain*).

4.1.1 Comments

Fig 2 and Fig 3 show that if the learning rate belongs to the range defined in (7), the stability of the identification scheme is garanteed. It is shown through this simulation that the identification objectives are satisfied. Out this variation domain of the learning rate, the identification is instable and the identification objectives are unreachable.

4.2 Second Order System

The second example concerns a discrete time system given by:

$$y(k)=50\tanh[\varphi(k-1)]+0{,}5u(k-1)$$

where (17)

$$\varphi(k)=2.10^{9}\left(\frac{(24+y(k))}{3}y(k-1)-8\frac{u^2(k)}{1+u^2(k)}y(k-1)\right)$$

The process dynamic is interesting. In fact it has the behaviour of a first order law pass filter for inputs signal amplitude about 0.1, the behaviour of a linear second order system in the case of small amplitudes ($0{,}1 < |u| < 0{,}5$) and the behaviour of a non linear second order system in the case of great inputs amplitudes ($0{,}5 < |u| < 5$) (Ching-Hang Lee and al, 2002).
For the neural model , a three-layer NN was selected with three inputs, three hidden and one output nodes. Sigmoidal activation functions were employed in all the nodes.
The weights are initialized to small random values. The learning rate parameter is computed instantaneously. As input signal, a sinusoidal one is chosen which the expresion is defined by:

$$u(k)=0.5\cdot\cos\left(0.005k\pi+\frac{\pi}{3}\right) \qquad (18)$$

The simulations are realized in the two cases. Two learnning rates values are fixed in and out of the learning rate range presented in (7).
Simulation results are given through the following figures:

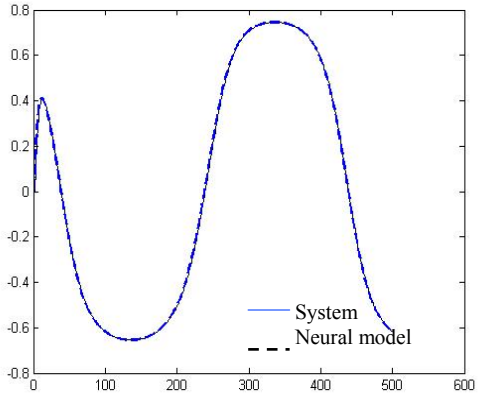


Figure 4: Evolution of the system output and the neural model output ($\varepsilon \in$ *stability domain*).

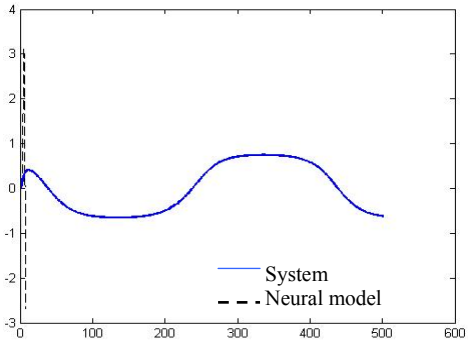


Figure 5: Evolution of the system output and the neural model output ($\varepsilon \notin$ *stability domain*).

4.2.1 Comments

Here we made a comparative study between an arbitrary choice of a learning rate out side of the stability domain and a constrained choice verifying the stability condition and guarantying tracking capability. The simulation results schow that a learning rate in the stability domain ensure the stability of the identification scheme.

5 CONCLUSIONS

To avoid unstable phenomenon during the learning process, constrained learning rate algorithm is proposed. A stable adaptive updating processes is guaranteed. A Lyapunov analysis is made in order to extract the new updating formulations which under inequality constraint. In the constrained learning rate algorithm, the learning rate is updated at each iterative instant by an equation derived using the stability conditions. The applicability of the approach presented is illustrated through two simulation examples.

REFERENCES

B. Egardt, 1979. Stability of adaptive controllers. in: *Lecture Notes in Control and Information Sciences 20*, Springer-Verlag, Berlin.

Z. Feng, A.N. Michel, 1999. Robustness analysis of a class of discrete-time systems with applications to neural networks, in: *Proceedings of American Control Conference*, San Deigo, pp. 3479–3483.

S.S. Ge, C.C. Hang, T.H. Lee, T. Zhang, 2001: Stable Adaptive Neural Network Control, *Kluwer Academic*, Boston.

P. A. Ioannou, J. Sun, 2004. Robust Adaptive Control, *Information Sciences 158*, 31–147, Prentice-Hall, Upper Saddle River.

L. Jin, M.M. Gupta, 1999. Stable dynamic backpropagation learning in recurrent neural networks, *IEEE Trans. Neural Networks 10 (6) , 1321–1334*.

Z. P. Jiang, Y. Wang, 2001. Input-to-state stability for discrete-time nonlinear systems, *Automatica 37 (2) , 857–869.*

E. B. Kosmatopoulos, M.M. Polycarpou, M.A. Christodoulou, P.A. Ioannou, 1995. High-order neural network structures for identification of dynamical systems, IEEE Trans. Neural Networks 6 (2), 431–442.

M. M. Polycarpou, P.A. Ioannou 1992. Learning and convergence analysis of neural-type structured networks, *IEEE Trans. Neural Networks 3 (1) ,39–50*.

J. A. K. Suykens, J. Vandewalle, B. De Moor, 1997. NLq theory: checking and imposing stability of recurrent neural networks for nonlinear modelling, *IEEE Trans. Signal Process (special issue on neural networks for signal processing) 45 (11) , 2682–2691.*

W. Yu, X. Li, 2001. Some stability properties of dynamic neural networks, *IEEE Trans. Circuits Syst., Part I 48 (1) , 256–259*.

W. Yu, A.S. Poznyak, X. Li, 2001. Multilayer dynamic neural networks for nonlinear system on-line identification, *Int. J. Control 74 (18), 1858–1864*.

E. Barn, 1992. Optimisation for training neural nets, *IEEE Trans. Neural Networks 3 (2) , 232–240*.

Ching-Hang Lee and al, 2002. Control of Nonlinear Dynamic Systems Via adaptive PID Control Scheme with Daturation Bound, *International Journal of Fuzzy Systems, Vol. 4 No. 4 ,922-927.*

K. S. Narendra and K. Parthasarathy, 1990. Identification and control of dynamical systems using neural networks*, IEEE Transaction Neural Networks, vol.1, pp.4–27, Mar.*

J. D. Boskovic and K.S.Narendra 1995. Comparison of linear nonlinear and neural-network based adaptive controllers for a class of fed-batch fermentation process*, Automatica, vol. 31, no. 6, pp. 537-547.*

A MARINE FAULTS TOLERANT CONTROL SYSTEM BASED ON INTELLIGENT MULTI-AGENTS

Tianhao Tang and Gang Yao
Department of Electrical & Control Engineering, Shanghai Maritime University
1550 Pudong Road, Shanghai, 200135, P. R. China
thtang_smu@yahoo.com, shakesteel@hotmail.com

Keywords: Multi-agent system, Faults tolerant control, Communication, Coalition, FIPA Agent Platform.

Abstract: This paper presents a hybrid intelligent multi-agent method for marine faults tolerant control (FTC). A new FTC schema, implemented by different kinds of agent, is discussed as well as the structure and functions of those agents, which have been encapsulated with intelligent algorithms to carry out different aspects in FTC. These agents could, having a purpose of trying to earn payoff as much as possible in a mission, communicate and form a coalition via negotiation when they find cooperation would bring them more benefits. Simulation experiments and its results are shown at last to demonstrate the efficiency of the proposed system.

1 INTRODUCTION

The development of modern marine vehicles is evolving rapidly towards the direction of large scale and complexity, as well as the trend of unmanned intervention, to satisfy the increasing requirements of international marine transportation trade. These features, hereby, call for more and more safer mechanism to guarantee the reliability of ship manoeuvring. For this reason, that applying fault tolerant control (FTC) theory, especially combined with distributed artificial intelligece (DAI), into marine control system attracted much attentions in recent years by researchers and engineers.

FTC could diagnosis component failures in a control system and maintain the system performance at a possibly low but acceptable level. Accordingly, it is possible to improve the system efficiency and to guarantee the operation safety in the control process (Edgar, 2000).

However, the control of marine vehicle, being a typical large scale and complicated system, is a nonlinear, undetermined, time-variable, and open process. The complexity of the FD and FTC system is growing with the increasing complexity of control plants. To keep the FD and FTC system effective, it is essential to encapsulate different tasks and to define strict interfaces between plant components and between components of the monitoring and diagnosis system, although it is quite difficult. To guarantee flexibility -- changing needs in case of an industrial application, the monitoring and FTC system has to be configurable and expandable without the need of modifying any line of code (Luder, 2001). The diagnostic knowledge about an industrial process is available on different parties (process specialists, component manufacturers, etc.). A modern FD and FTC system should be able to integrate the diagnostic knowledge from all available sources, even if different diagnostic mechanisms are applied. To achieve an overall diagnosis of a control process, several diagnostic tasks have to be performed in parallel. This requires new strategies to handle diagnostic conflicts that might occur between different diagnostic results.

Multi-agent system (MAS), about which rapid progress has been made, is an important research branch in DAI parallelized with distributed problem solving (DPS). Possessing modularity, adaptability and other attractive characteristics, MAS drew much attention in recent years and is adopted by many researches in control systems.

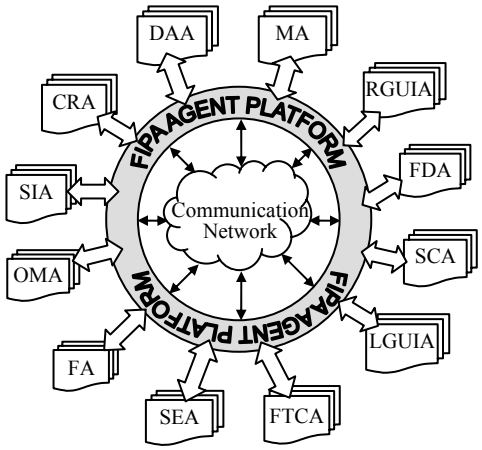
This paper presents a hybrid intelligent mutli-agent method for mrine fault tolerant control. The architecture of the MAS, as well as the structure of an agent, and the control diagram are designed in section 2. Then, the algorithms encapsulated in the information processing mudule of agents are briefly mentioned in section 3. In section 4, simulation experiments, applying the proposed method in

marine FTC system, is carried out. And the conclusion of this paper is made in section 5.

2 SYSTEM ARCHITECTURE

2.1 MAS Perspective

The framework of hybrid intelligent multi-agent method, with hierarchical and federal organized intelligent agents that are responsible for different tasks, is presented in figure 1.



MA-Management Agent; RGUIA/LGUIA-Remote/Local Graphic User Interface Agent; FDA-Fault Diagnosis Agent; SCA-State Control Agent; FTCA-Fault Tolerant Control Agent; SEA-State Estimate Agent; FA-Facilitate Agent; OMA–Ontology Mediate Agent; SIA-System Identification Agent; CRA-Conflict Resolution Agent; DAA-Database Access Agent

Figure 1: Architecture of MAS organization.

As shown in the figure, many agents with different capabilities are connected together by accessing FIPA Agent Platform and through communication networks to form a MAS. In this system, each individual has a special function that can work autonomously and independently. Although being fully autonomous, like human beings, on the other hand, agents in the figure trend to seek cooperation to fulfil more difficult task if they believe that better payoff will gain by form a coalition or the job assigned to some agents is impossible to achieve with their own capability. To facilitate a coalition, *superadditive environment* (Nicholas, 2007) is assumed in this paper.

At agent level, all agents in proposed system have the same hybrid architecture, where the agents are capable of reactive and deliberative behaviours. The proposed agent architecture is based on horizontal layering where all layers are connected to the perception and actuation of the agents with the environment. In the reactive layer, an agent could deal with urgent situations according to the rule in its rules library while it will do some inference, in normal case, in its information processing module (IPM), which has been encapsulated with intelligent algorithms and upon which different type of agents are determined, according to the current mental state, knowledge and goal in deliberation layer.

Details about the MAS and agent architecture could refer to (Yao, 2007).

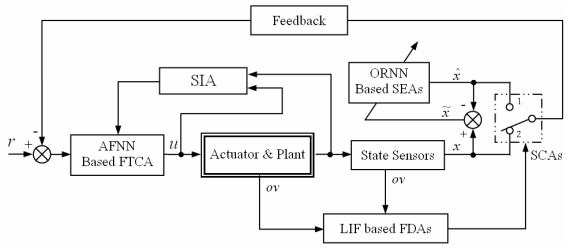
In figure 1, twelve species of agent, which are determined by different algorithms in an agent's IPM, are involved: MAs are mainly used in FIPA Platform to manage the life cycle of agents; FAs provide yellow page services; DAAs are responsible for interoperation with database; RGUIAs and LGUIAs are graphic user interface extendedly and locally respectively; FDAs are in charge of diagnosis faults for the whole system; FTCAs are going to do fault tolerant control in case that the system actuator lost efficiency. SEAs are supposed to replace failed sensors by estimating state signals and feedback them to keep the system stable; SIAs collaborate with the FTCA to do fault tolerant control; SCAs answer for switching feedback signals from failed sensors to SEAs; OMAs maintain term mapping tables between different domain ontologies; CRAs resolve conflicts in diagnosis results reported by different FDA. When faults exist in a control system or a user instruction is issued, all the above species of agent will act autonomously and could communicate to form coalitions to fulfil the task with lease cost.

2.2 Control System Perspective

The architecture of MAS was discussed in last section. In order to illuminate how those agents implement fault tolerant control in a control system, this section describes MAS based FTC system architecture in a state feedback control system perspective, as shown in figure 2.

In the figure, r is system reference; u is the output of an adaptive fuzzy neural network (AFNN) based FTCA, which acts as system controller; OV is omen variables collected by fault diagnosis agents; $\tilde{x}$ is the difference between x (the detected value from system sensor) and $\hat{x}$ (the output value of an output recurrent neural network (ORNN) based SEA), which is use by IPM of SEA to train the state

estimation neural network on line; SCAs are used to cut off failed sensors.



AFNN-Adaptive Fuzzy Neural Network
ORNN-Output Recurrent Neural Network
LIF- Layered Information Fusion

Figure 2: Architecture of MAS based FTC system.

In this FTC system there are three distinct parts that differ from traditional control systems: fault diagnosis agents to diagnose system failure, state estimation agents to estimate state approximately when the sensor failed, and system self-adaptive controller, realized by FTCA, for FTC under actuator faults:

(1) In the LIF based FDA, two kinds of faults, actuator faults and sensor faults will be diagnosed. If actuator faults happened, FTCA will take some responding actions according to the fault state. While the sensor faults have been found, the SEA will play a part in the sensors.

(2) A SEA based on an output recurrent neural network is proposed to provide the system states estimation to replace the fault sensors. Usually, SCA controls the switch to be on the position 2. In this case, the system state feedback signals will be from sensors. If the sensor faults occur, SCA will switch to the position 1. In this case, the signals from SEAs will be used as the system feedback.

(3) The FTCA, system controller, will give different control strategy according to the system state and fault cases to keep the system performance. For example, when a fault from the actuator was detected, the FTCA based on AFNN could adjust the control signal to overcome the influence of the fault according to the system response on line.

3 INTELLIGENT ALGORITHMS OF AGENTS

As mentioned above, algorithms encapsulated in an agent's IPM determines its functions. In the IPM of a FDA, multi-layer information fusion technology is adopted for fault diagnosis, which separated fault diagnosis into two parts: local diagnosis fusion implemented by multi-sensors fuzzy inference and global diagnosis fusion implemented by a three-layer fuzzy neural network. In SEA, a new output recurrent neural network is designed to construct the system state estimator. For FTCA, a self-adaptive fuzzy neural network is proposed as its information process method. Detail about these solutions could refer to (Yao, 2006)

4 SIMULATION EXPERIMENTS

4.1 Experiment Platform

In this simulation, the MAS framework is coded in JADE platform, a software development framework for agent application developed by TILAB. The algorithms mentioned in section 3 are coded in Matlab 6.5.

To implement calling Matlab methods from JADE, JMatLink, a small toolkit to connect Java with Matlab, is used to call for the functions in an m-file.

The main user interface of MAS compiled in JADE is shown in figure 3.

4.2 Working Flow of MAS

A prominent advantage of MAS is that agents could discover an optimized way to fulfil tasks by negotiation, coordination and cooperation via sending messages. Accordingly, the communication and cooperation between agents are the most two important research topics regarding MAS.

The communication among agents, as well as their knowledge and mental state, is based upon domain ontology. But different ontologies regarding one domain may exist sometimes in a system. These ontologies contain different terms, which engender great obstacles for agent communication, to express same or similar concepts. Dealing with this problem, a method called term substitution based on intelligent ontology mapping is proposed and implemented by ontology mediation agent (OMA).

OMA maintain glossary mapping tables between domain ontologies. When an agent receives an ACL message containing a few baffling words, the agent forwards this message to OMA for interpretation. If OMA could find terms in an ontology upon which this agent based corresponding to those baffling ones, it will substitute them and send the message back. And then, the agent will understand the message meaning.

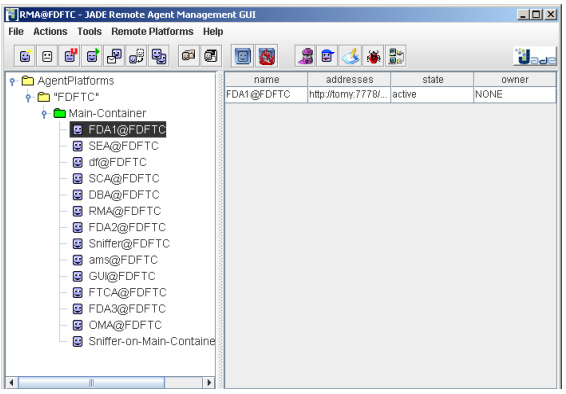


Figure 3: MAS interface in JADE.

Cooperation will bring agents more efficiency and benefits, while coalition is an important solution to implement it. As mentioned in section 2.1, superadditive environment is assumed to facilitate the coalition formation. Under this circumstance, agents in MAS are inclined to form a grand coalition, because they believe that they will earn payoff in this coalition at least as much as those they will get if they work alone. New task and payoff allocation algorithms are also put forward. Details about these solutions regarding with the agent communication and coalition will be specified in future papers. And some research results about these solutions have been applied in agents work flow in this simulation experiment, as illustrated in figure 4.

4.3 Marine FTC System Framework

The method proposed above had been used in marine automatic steering system for actuator fault tolerant control. The structure of ship FTC system is shown in figure 5.

In order to represent the rudder faults, a coefficient called as loss of efficient (*LOE*) is introduced to simulate the faults. When the rudder is normal, define *LOE*=0. And define *LOE*=1 if the rudder is whole failure. So the coefficient *LOE* expresses the fault degree of the rudder. In the simulation system, the faults of the rudder servo system could be set by fault factor L, which is defined as L=1-*LOE*.

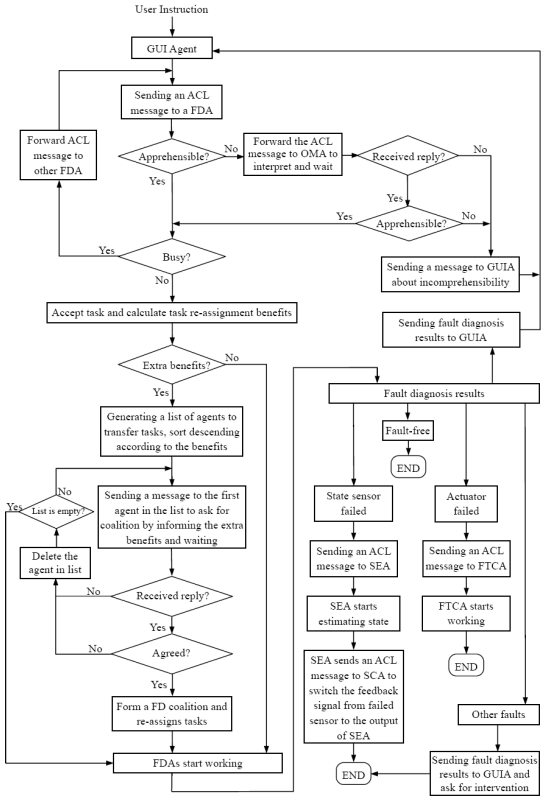


Figure 4: Flow chart of MAS based FTC process.

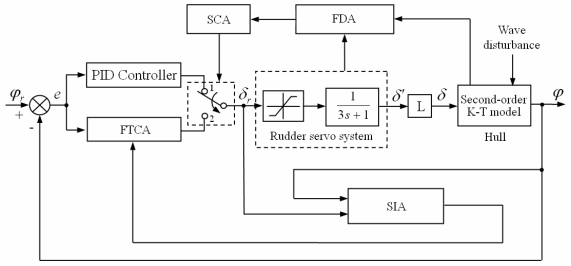


Figure 5: Marine FTC System.

Marine vehicles will definitely be disturbed by ocean wave which is the uppermost factor to cause ship rolling. Usually, wave motions on a ship can be analytically computed using strip theory. The ocean wave spectrum model adopted in the simulation is shown in figure 6.

4.4 Simulation Results

During the simulation, Sniffer, a monitoring tool in JADE, is used to sniffer the communication of agents. The monitoring result is shown in figure 7.

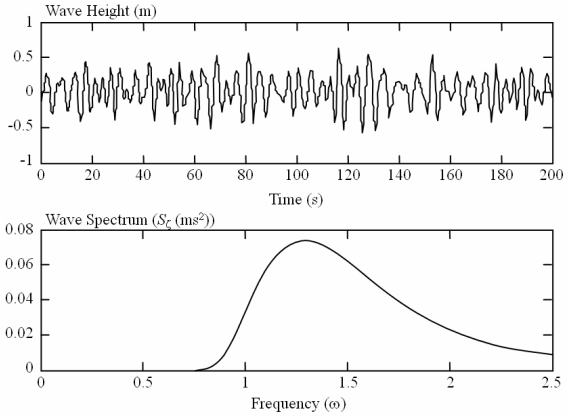


Figure 6: Ocean wave spectrum.

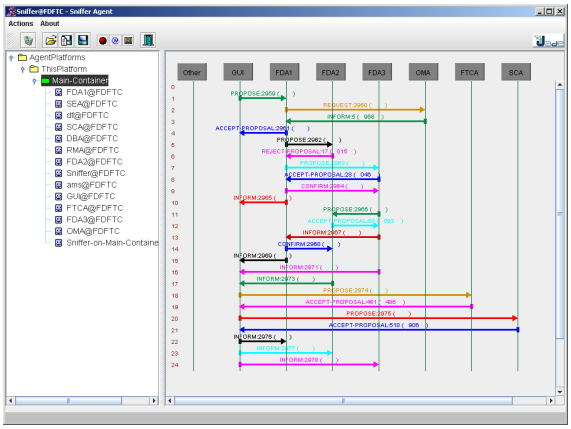


Figure 7: The communication of cooperated agents.

Some rudder faults with different degrees are used to test the performance of the proposed system. In these cases, the effect of FTC is simulated with and without wave disturbances.

To judge how well the FTC method performs, a PID controller is used to perform the same tasks under the same conditions. Some simulation results are shown in figure 8 ~ figure 11.

In order to make a detailed comparison between the two kinds of controllers with rudder faults, the rise times of the ship steering system are recorded to evaluate the performances, where the rise time is defined as the time taken for the ship to rise from five to ninety five percent of the demanded yaw angle. The rise times of the two controllers in different fault conditions without and with disturbances have been compiled in Table 1 and Table 2 shown below.

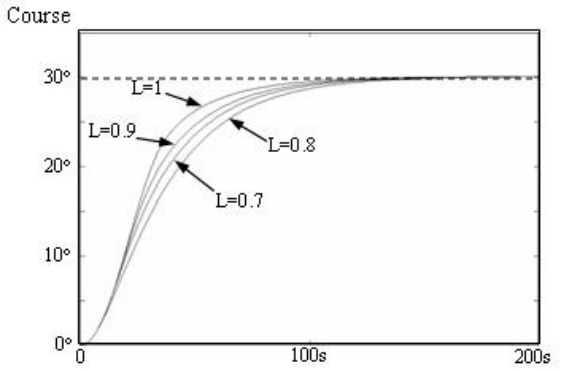


Figure 8: Ship courses with PID controller (without disturbance).

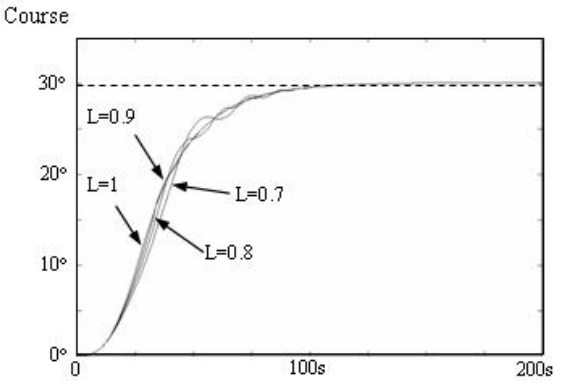


Figure 9: Ship courses with FTCA (without disturbance).

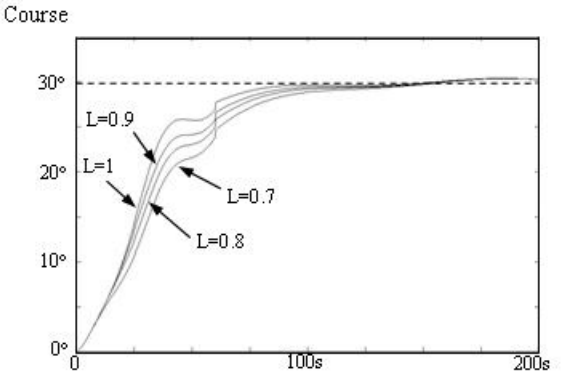


Figure 10: Ship courses with PID controller (with disturbance).

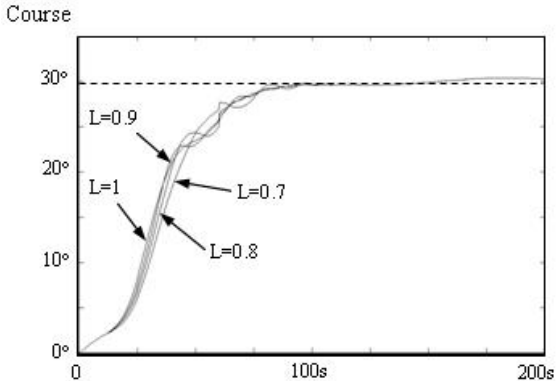


Figure 11: Ship courses with FTCA (with disturbance).

Table 1: Rise Times without Disturbances (s).

	LOE =0	*LOE* =10%	*LOE* =20%	*LOE* =30%
PID	70	81	87.5	96.5
FTCA	70	70	70.5	71

Table 2: Rise Times with Disturbances (s).

	LOE =0	*LOE* =10%	*LOE* =20%	*LOE* =30%
PID	75	84	90	98.5
FTCA	75	75.5	77	79.5

From the figure 7, it could be seen that different types of agents in MAS followed the working flow shown in figure 4 to work cooperatively. When a FDA found there exists baffling terms in an ACL message, it will ask for help to OMAs automatically. If an agent finds that extra value will be gained if tasks are re-assigned, it will send proposing message to others about coalition by informing this value. When the coalition is formed, tasks will be done and coalition value will be divided according to each agent's contribution.

From the showings of figure 8, figure 10 and two tables, it is easy to find that the PID control system doesn't have the fault tolerant capability; when the rudder servo system lost some efficiency, the system performance is greatly reduced with increased rise time and stable time.

Figure 9, figure 11 and also those two tables show that in the MAS based marine fault tolerant control system, the system's performance can be distinctively improved in the condition of actuator failure; it is nearly close to the normal performance. The rising time and stable time are not influenced much. The target of actuator fault tolerant control is fulfilled.

From the simulation experiment above, it is easy to find that the MAS based marine fault tolerant control system can adapt to different failure modes when the rudder servo system is partially failed. It successfully realized fault tolerant control for actuator failure by the multi-agents organization shown in figure 1. The agents could communicate to seek coalition autonomously. Adopting this MAS based method will need no object's mathematic model and could realize cooperation between heterogeneous intelligent algorithms.

5 CONCLUSIONS

This paper describes a concept of building a hybrid intelligent fault tolerant control system for marine vehicles based on the application of MAS, and also proposed a new fault tolerant control schema integrates several algorithms implemented within the MAS method, which allows the flexibility, the extendibility, and a cost-effective development of the system. Details about the overall architecture, algorithm encapsulated in IPM, and coding tools are discussed. And at last, some simulation experiment results are given to demonstrate the efficiency of the presented system.

ACKNOWLEDGEMENTS

This work was supported by Natural Science Fund of China (NSFC 60434020, 60572051); the Education Key Project (07ZZ102) and the Education Development Project (08YZ109) from Shanghai Municipal Education Commission.

REFERENCES

Edgar, T. F., Dixon, D. A, 2000, Computational needs of the chemical industry, University of Texas Press.

Luder, A., et al., 2001, Industrial requirement and overall specification. Prepared within the PABADIS IST research project no. IST-1999-60016. Available at www.pabadis.org.

Nicholas R. Jennings, 2007, Forming fuzzy coalitions in cooperative superadditive games, IEEE International Conference on e-Business Engineering (ICEBE'07) pp 221-228.

Yao Gang, Tianhao Tang, 2007, A hybrid intelligent multi-agent method for monitoring and faults diagnosis, 4th International Conference on Informatics in Control, Automation and Robotics, France.

Guy Lebret, Gang, Yao, 2006, A gain-scheduling and intelligence fusion method for fault-tolerant control, 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Beijing, China.

AUTHOR INDEX

AUTHOR INDEX (CONT.)

Proceedings of ICINCO 2008
Fifth International Conference on Informatics in Control, Automation and Robotics
ISBN: 978-989-8111-30-2
http://www.icinco.org