8th INTERNATIONAL INDUSTRIAL SIMULATION CONFERENCE 2010

ISC'2010

EDITED BY

Gábor Lencse

and

László Muka

JUNE 7-9, 2010

BUDAPEST, HUNGARY

A Publication of EUROSIS-ETI

8th Industrial Simulation Conference 2010

BUDAPEST, HUNGARY

JUNE 7-9, 2010

Organised by

ETI- The European Technology Institute

Sponsored by

EUROSIS - The European Simulation Society

Co-Sponsored by

AEKI

CREAX

Ghent University

UPV

Hosted by

Ramada Plaza Hotel

Budapest, Hungary

EXECUTIVE EDITOR

PHILIPPE GERIL
(BELGIUM)

EDITORS

General Conference Chair
Gábor Lencse, Szechenyi Istvan University, Gyor, Hungary

General Program Chair
László Muka, Elassys Consulting Ltd, Budapest, Hungary

Publications Chair
Yan Luo, NIST, Gaithersburg, USA

INTERNATIONAL PROGRAMME COMMITTEE

Discrete Simulation Methodology, Languages and Tools
Matthias Becker, University Hannover, Hannover, Germany
Helge Hagenauer, Universitaet Salzburg, Salzburg, Austria
Sophie Hennequin, ENIM, Metz Cedex, France
Mauro Iacono, SUN II, Universita di Napoli, Naples, Italy
Bjorn Johansson, Chalmers University of Technology, Gothenburg, Sweden
Sarka Kvetonova, Brno University of Technology, Brno, Czech Republic
Jan Lemeire, VUB, Brussels, Belgium
Goreti Marreiros, Polytechnic of Porto, Porto, Portugal
Stefano Marrone, Seconda Università di Napoli. Naples, Italy
Maria do Rosário Moreira, University of Porto, Porto, Portugal
Aziz Naamane, ISIS-EPUM, Marseille, France
Paulo Novais, Universidade do Minho, Braga, Portugal.
Werner Pohlmann, Universitaet Salzburg, Salzburg, Austria.
Jiri Safarik, University of West Bohemia, Plzen, Czech Republic
Werner Sandmann, TU-Clausthal, Clausthal, Germany.
Antonella Zanzi, Institute for the Protection and Security of the Citizen, Joint Research Centre - European Commission, Ispra , Italy

Simulation in Manufacturing
Pascal Berruet, Universite Bretagne Sud, Lorient, France
Peter Byrne, Dublin City University, Dublin, Ireland
Ana Camacho, UNED, Madrid, Spain
Luis Camarinha-Matos, Universidade Nova de Lisboa, Lisbon, Portugal
Eduardo Castellano, IKERLAN Technological Research Centre, Mondragón-Arrasate, Spain
Remy Dupas, Universite Bordeaux 1, Bordeaux, France.
Alexander Felfernig, University of Klagensfurt, Klagensfurt, Austria
Habtom Mebrahtu, APU, Chelmsford, United Kingdom
Carlo Meloni, Politecnico di Bari - DEE, Bari, Italy
Pascal Meyer, Forschungszentrum Karlsruhe, Karlsruhe, Germany
Marcos A.Rodrigues, Sheffield Hallam University, Sheffield, United Kingdom
Marina Valles, UPV, Valencia, Spain .
Markus Vorderwinkler, PROFACTOR GmbH, Steyr, Austria

Simulation in Steel Manufacturing
Brian Hollocks, Bournemouth University, Bournemouth, United Kingdom

Simulation in Automotive Systems
Naoufel Cheikhrouhou, EPFL, Lausanne, Switzerland
Julien Richert, Daimler AG, GR/PAA, Sindelfingen, Böblingen, Germany
Rob Walker, Anglia Ruskin University, Chelmsford, United Kingdom

INTERNATIONAL PROGRAMME COMMITTEE

Simulation in Robotics
Leopoldo Arnesto Angel, UPV, Valencia, Spain
A. Chatzinikolaou, Athens, Greece
Andrzej Dzielinski, Warsaw University of Technology, Warsaw, Poland
Markus Koch, Orga Systems GmbH, Paderborn, Germany
George L.Kovacs, Hungarian Academy of Sciences, Budapest, Hungary
Alexandru Marin, Polytechnical Inst.of Bucharest, Bucharest, Romania.
Martin Mellado, UPV, Valencia, Spain
Bogdan Raducanu, Computer Vision Centre, UAB, Barcelona, Spain
Krzysztof Skrzypczyk, Silesian University of Technology, Gliwice, Poland
Sergiu-Dan Stan, Technical University of Cluj-Napoca, Cluj-Napoca, Romania

Simulation in Electronics, Computer and Telecommunications
Christos Bouras, University of Patras, Patras, Greece
Boguslaw Butrylo, Bialystok Technical University, Bialystok, Poland
Tom Dhaene, Ghent University, INTEC, Ghent, Belgium
Domenico Giunta, ESA-ESTEC, Noordwijk, The Netherlands
Silvia Mirri, University of Bologna, Bologna, Italy
Maurizio Palesi,Universita di Catania, Catania, Italy
Marco Roccetti, University of Bologna, Bologna, Italy
Fernando Boronat Segui, UPV, Valencia, Spain
Renate Sitte, Griffith University, Gold Coast, Australia

Simulation of Complex Multiprocessor Systems
Orhan Gemikonakli, Middlesex University, London, United Kingdom

Simulation in Computer Science
Paul Chung, Loughborough University, Loughborough, United Kingdom .
Lipika Deka, Tezpur University and IIT Guwahati, India

Simulation in Logistics, Transport and Harbour Simulation
El-Houssaine Aghezzaf, Ghent University, Ghent, Belgium
Christian Almeder, University of Vienna, Vienna, Austria
Isabel Garcia Guttierez, Univ. Carlos III de Madrid, Madrid, Spain
Martin Grunow, Technical University of Denmark, Lyngby, Denmark
Thomas Hanne, University of Applied Sciences Northwestern Switzerland, Olten, Switzerland
Marie-Ange Manier, UTBM, Belfort, France
Roberto Montemanni, IDSIA, Manno-Lugano, Switzerland
Jaap Ottjes, TU Delft, Delft, The Netherlands.
Selwyn Piramuthu, University of Florida, Gainesville, USA
Roberto Razzoli, University of Genova, Genova, Italy
Rosaldo Rossetti, University of Porto, Porto, Portugal
Rik van Landeghem, Ghent University, Ghent, Belgium
Hans Veeke, TU Delft, Delft, The Netherlands
Pengjun Zheng, University of Southampton, Southampton, United Kingdom

Hospital Logistics Simulation
Track Chair: Giorgio Romanin-Jacur, University of Padova, Vicenza, Italy
Antonio Abelha, Universidade do Minho, Braga, Portugal .
Joel Colloc, Universite du Havre, Le Havre, France .
Jose Machado, University of Minho, Braga, Portugal .
Peter Summons, University of Newcastle, Australia

INTERNATIONAL PROGRAMME COMMITTEE

Complex Systems Modelling
Frantisek Capkovic, Slovak Academy of Sciences, Bratislava, Slovak Republic
Gabriella Dellino, University of Siena, Siena, Italy
Werner Dubitzky, University of Ulster, Coleraine, United Kingdom
Krzysztof Fujarewicz, Silesian Technical University, Gliwice, Poland
László Gulyás, AITIA International Inc., Budapest, Hungary
Alexandre Nketsa, LAAS, Toulouse, France
Miguel Rocha, University do Minho, Braga, Portugal
Alfonso Urquia, UNED, Madrid, Spain

Simulation in Aerospace
Reza Azadegan, Urmia University, Urmia, Iran
Wolfgang Kuehn, University of Wuppertal, Wuppertal, Germany
Martin Spieck, DLR, Goettingen, Germany

Marine Simulation
Cyril Ray, Ecole Navale, Brest, France

Simulation in Industrial Design and Product Design
Sergio Baragetti, University of Bergamo, Dalmine, Italy
Chiara Catalano, IMATI-CNR, Genoa, Italy
Yan Luo, NIST, Gaithersburg, USA
Jorge Perez Mateos, AOS, Madrid, Spain.
Catarina Rizzi, University of Bergamo, Bergamo, Italy

Simulation in Engineering Processes
Chrissanti Angeli, Technological Institute of Piraeus, Athens, Greece
Blazej Balasz, Technical University of Koszalin, Koszalin, Poland
Alejandra Gomez Padilla, University of Guadalajara, Mexico
Jan Studzinski, Polish Academy of Sciences, Warsaw, Poland
Joao Tavares, University of Porto, Porto, Portugal
Yiannis Ventikos, Oxford University, Oxford, United Kingdom
Henk Versteeg, Loughborough University, Loughborough, United Kingdom

Civil and Building Engineering
Alistair Borthwick, Oxford University, Oxford, United Kingdom
Ashu Jain, IIT Kanpur, India
Graham Saunders, Loughborough University, Loughborough, United Kingdom

Simulation in Energy and Power Systems
Jean-Frederic Charpentier, Ecole Navale, Brest, France
Manuel Cordeiro, CETAV/UTAD, Vila Real, Portugal
Janos-Sebestyen Janosy, KFKI Atomic Energy Research Institute, Budapest, Hungary
Gustaf Olsson, Lund University, Lund, Sweden

Simulation in Multibody Systems
Track Chair: Bernd Schaefer, DLR, Wessling, Germany
Ignacio García-Fernández, University of Valencia, Valencia, Spain
Jose Diaz Lopez, Dynasim AB, Lund, Sweden .

Simulation in Chemical, Petroleum and Mining Engineering
Diganta Das, Loughborough University, Loughborough, United Kingdom
Vahid Nassehi, Loughborough University, Loughborough, United Kingdom
Mohamad R. Riazi, Kuwait University, Kuwait

Simulation in Military and Defense
Roberto de Beauclair Seixas, IMPA, Rio de Janeiro, Brazil
Carlos Palau, UPV, Valencia, Spain
Matthias Reuter, CUTEC Gmbh, TU-Clausthal, Clausthal, Germany

INTERNATIONAL PROGRAMME COMMITTEE

Virtual Reality and Graphical Simulation in Industrial Simulation
Track Chair: Guodong Shao, NIST, Gaithersburg, USA
Emilio Camahort, Universidad Politecnica de Valencia, Valencia, Spain
Anders Hast, University of Gavle, Gavle, Sweden
Fabrizio Lamberti, Politecnico di Torino, Turin, Italy
Sudhir Mudur, Concordia University, Montreal, Canada

Verification, Validation and Accreditation
Mokhtar Beldjehem, École Polytechnique de Montréal, Canada
Agustin Yague, Technical University of Madrid, Madrid, Spain

Simulation and Training
Manuel Alfonseca, Universidad Autonoma de Madrid, Madrid, Spain
Wenji Mao, Chinese Academy of Sciences, Beijing, P.R. China
Eshan Rajabally, Loughborough University, Loughborough, United Kingdom
Gerhard Schreck, Fraunhofer IPK, Berlin, Germany

Simulation in Lean Manufacturing
El-Houssaine Aghezzaf, Ghent University, Ghent, Belgium
Hendrik Van Landeghem, Ghent University, Ghent, Belgium

INDUSTRIAL SIMULATION 2010

© 2010 EUROSIS-ETI

Responsibility for the accuracy of all statements in each peer-referenced paper rests solely with the author(s). Statements are not necessarily representative of nor endorsed by the European Simulation Society. Permission is granted to photocopy portions of the publication for personal use and for the use of students providing credit is given to the conference and publication. Permission does not extend to other types of reproduction, nor to copying for incorporation into commercial advertising nor for any other profit-making purpose. Other publications are encouraged to include 300- to 500-word abstracts or excerpts from any paper contained in this book, provided credits are given to the author and the conference.

All author contact information provided in this Proceedings falls under the European Privacy Law and may not be used in any form, written or electronic, without the written permission of the author and the publisher. Infringements of any of the above rights will be liable to prosecution under Belgian civil or criminal law.

All articles published in these Proceedings have been peer reviewed

EUROSIS-ETI Publications are Thomson Reuters and INSPEC referenced

Selected papers of this conference are published in scientific journals.

For permission to publish a complete paper write EUROSIS, c/o Philippe Geril, ETI Executive Director, Greenbridge NV, Wetenschapspark 1, Plassendale 1, B-8400 Ostend, Belgium.

EUROSIS is a Division of ETI Bvba, The European Technology Institute, Torhoutsesteenweg 162, Box 4, B-8400 Ostend, Belgium

Printed and bound in Belgium by Reproduct NV, Ghent, Belgium
Cover Design by Grafisch Bedrijf Lammaing, Ostend, Belgium

EUROSIS-ETI Publication
ISBN: 978-90-77381-5-57
EAN: **978-90-77381-5-57**

PREFACE

Dear participants,

It is our pleasure to welcome you to the 8th Industrial Simulation Conference, ISC'2010, organised by EUROSIS.

Nowadays, as a result of the global crisis, we have entered into a period of increased uncertainty. In this period, however, the modelling and simulation community has a great opportunity and also responsibility to apply methods and tools of modelling and simulation to cope with this uncertainty. Something, we will try to address partially at this year's ISC.

The host city of the conference, Budapest, capital of Hungary is one of the major cities of Central Europe. The Romans already built the town of Aquincum here, and you can still see their aqueducts and amphitheatres just north of Óbuda. Budapest itself emerged from the union of three towns namely the hilly Buda, the flat Pest and the little Óbuda. The river Danube splits the city in two – thus dividing Buda and Pest.

Budapest is also an inspiring city; a city of innovators and original thinkers. The origin of this inspiring effect can probably be traced somewhere in the history and the situation of the city.

The Gömböc (pronounced [gømbøts]), invented by Hungarian scientists, may be one of the interesting inspirations. By definition, Gömböc is a convex three-dimensional homogeneous body which, when resting on a flat surface, has just one stable and one unstable point of equilibrium. Gömböc is also said to be a metaphor for Hungary: it would return to equilibrium position after being placed upside down, no matter how it is pushed or turned around. This metaphor may also help the participants of the conference to find the right paradigms.

For ISC'2010 itself, we would like to thank the keynote and invited speakers who contribute to the effectiveness of the conference. We thank the authors of all submitted papers for their contributions and the paper reviewers for reviewing papers and providing valuable feedback to the authors.

We would like to express special thanks to Philippe Geril from EUROSIS, for managing the compound processes of preparation of the conference and to the numerous contributors and organisers for their tireless efforts.

Thank you for your participation in ISC' 2010 and welcome to Budapest, Hungary. We hope you benefit professionally from the conference.

Gábor Lencse and László Muka
ISC'2010 General Conference and Program Chairs

CONTENTS

Preface ..... **XI**
Scientific Programme **1**
Author Listing **265**

SIMULATION METHODOLOGY AND AI

Contact Status Simulation and Fretting Fatigue Threshold Determination of Spline SHC under Alternating Loadings
Longjiang Shen, Armin Lohrengel and Günter Schäfer **5**

Design of Novel Self-Organization Mechanisms through Simulation
Marton Legeny and Borbala Katalin Benko **10**

SIMULATION TOOLS

Waveform Relaxation for the Simulation of Complex Hybrid Systems in SCICOS
Damien Chapon, Ramine Nikoukhah and François Delebecque **17**

Graphical Object Oriented Simulation with SIMIO
Patrick Kirchhof and Nicolas G. Meseth **22**

FORTIS – An Integrated Simulator for Distributed Time Series Forecasting
Janetta Culita and Dan Stefanoiu **27**

ViennaIPD - An Input Control Language for Scientific Computing
Josef Weinbub, Karl Rupp and Siegfried Selberherr **34**

Simulation of Real-time Robot Control Systems using Player/Stage Software
Vitalijs Komasilovs and Egils Stalidzans **39**

MANAGEMENT SIMULATION

We can save our World if Worlds collide
William Conley **45**

The Large Asteroid Problem from an Industrial Management Perspective
William Conley **50**

CONTENTS

BUSINESS PROCESS MODELLING

Storyboards in Business Process Modelling
Veronika Juchova, Svatopluk Stolfa, David Ježek and Ivo Vondrak **57**

Barriers to the Application of Simulation for Tactical decision Making
PJ Byrne, Paul Young, John Geraghty, Cathal Heavey and Barry Kennedy..... **62**

AIDATrace - Interoperation Platform for Active Monitoring in Healthcare Environments
José Machado, Miguel Miranda, Pedro Gonçalves, António Abelha, José Neves and Alberto Marques **67**

MANUFACTURING PROCESS SIMULATION

A Comparison of Modified Just-in-Time Production system with CONWIP
Ali Ardalan and Rafael Diaz **75**

BPM Method Extension for Automatic Process Simulation
Štěpán Kuchař and Jan Kožusznik **80**

MANUFACTURING OPTIMIZATION

A LEGO Factory for teaching Simulation Based Production Optimization
Anna Syberfeldt **89**

Automatic Analysis of Manufacturing Simulation Output
Habtom Mebrahtu and Rob Walker..... **95**

A Design Method for Parts Picking Zones in a Manufacturing Environment
Klaas Peerlinck, Tim Govaert and Hendrik Van Landeghem **102**

VIRTUAL MANUFACTURING

A comprehensive 3D Simulation System for the Virtual Production
Jürgen Roßmann, Roland Wischnewski and Oliver Stern..... **109**

Specified CAD (Computer Aided Design) Tool for Rapid Manufacturing Technology
Estitxu Martínez de Alegría Uriarte, Iñaki García, Aurelio Díaz de Arcaya, Gonzalo Lázaro and Luis Usatorre **117**

Micro Pattern Shaping Simulation
Su-Jin Kim, Jong-Min Lee, Duy Le and Tae-Jin Je **122**

COMPLEX ENGINEERING SYSTEMS SIMULATION

Building of a Motion feedback Roller Coaster Simulator: A case Study
Miguel A. Expósito-Sánchez, Ignacio García-Fernández,
Miguel A. Gamón-Giménez, Marta Pla-Castells and Rafael J. Martínez-Durá..**131**

Modelling and Simulation of Several Interacting Cellular Automata
Marta Pla-Castells, Ignacio García-Fernández and Rafael J. Martínez-Durá ...**139**

Modular Technology: A Methodology for the Creation of Virtual Environments in Terrain Driving Simulators
Carlota Tovar, Ginés Jesús Jimena and Jose María Cabanellas.....**147**

Simulation of an HEV equipped with a variable displacement ICE
Valerian Croitorescu, Andrei Maciac, Marius Bataus, Mircea Oprean and Cristian Andreescu.....**154**

TRAFFIC SIMULATION

Hierarchical Architecture for Urban Traffic Management – Simulation
Monica Voinescu, Andreea Udrea, Simona Caramihai and Calin Munteanu **161**

Simulation of Traffic Operations for Sea-crossing Bridges
Guiyun Liu and Pengjun Zheng.....**167**

A deterministic Annealing Algorithm for Simultaneous Routing of Loaded and Empty Containers
Kris Braekers, Gerrit K. Janssens and An Caris **172**

INVENTORY LOGISTICS AND SUPPLY CHAIN SIMULATION

Requirements for Electronic Tags from a Business Perspective
Izabella Lokshina and Evan Schiele **179**

Robustness of the Performance of an Unreliable Product Inventory System against various Demand Patterns
Roongrat Pisuchpen and Gerrit K. Janssens **184**

Simulation of a Basic Integrated Inventory Problem with Quality Improvement and Lead Time Reduction
Sofie Van Volsem and Johannes Verdiesen **190**

Green Supply Chain Event Management
Matthias Klumpp, Sascha Bioly, Alexandra Saur and Stephan Zelewski **195**

CONTENTS

TELECOM NETWORKS SIMULATION

Achievable Cell throughput in 3G Systems
Peter Fazekas.....**203**

Models and Algorithms for Self Similar IP Network Traffic Simulation and Performance Analysis
Dimitar Radev and Izabella Lokshina.....**208**

Performance Prediction of Conservative Parallel Discrete Event Simulation
Gábor Lencse and András Varga.....**214**

FLUID CHEMICAL SIMULATION

Fault Detection in Fluid Systems: An Interactive Realtime Approach
C. Angeli and A. Chatzinikolaou.....**223**

Topological Features and Parameters of Biochemical Network Structures
Tatjana Rubina and Egils Stalidzans **228**

SIMULATION IN ENERGY

Spatial Power Distribution Simulation for Nuclear Reactor
Janos Sebestyen Janosy, Andras Kereszturi, Gabor Hazi,
Jozsef Pales and Endre Vegh.....**239**

Model Based Real Time Automated Temperature Control System for Risk Minimization in Honey Bee Wintering Building
Aleksejs Zacepins, Jurijs Meitalovs and Egils Stalidzans **245**

Adjustment of Hydrostatic Transmissions through Virtual Instrumentation Technique
Teodor Costinel Popescu, Dragos Daniel Ion Guta and Alexandru Marin **248**

LATE PAPER

A Fuzzy Q-Learning Approach to Simulate Intelligent Traffic Control
Juan C.Pacheco, Rosaldo J.F.Rossetti and Cesar H.Rodriguez **257**

SCIENTIFIC PROGRAMME

SIMULATION METHODOLOGY AND AI

CONTACT STATUS SIMULATION AND FRETTING FATIGUE THRESHOLD DETERMINATION OF SPLINE SHC UNDER ALTERNATING LOADINGS

Longjiang Shen
Armin Lohrengel
Günter Schäfer
Institute of Mechanical Engineering, Clausthal University of Technology
D38678 Clausthal-Zellerfeld, Germany
E-mail: shen@imw.tu-clausthal.de

KEYWORDS

Spline shaft-hub connections, Contact status simulation, Stick-slip, Fretting fatigue, Critical loading

ABSTRACT

It is generally observed that cracks initiate at the stick-slip (partial slip) zone between fretting contact interface and lead to fretting fatigue. In order to avoid unexpected fracture caused by fretting fatigue, it is worth to trace the evolution of stick-slip zone under alternating loading and find critical loadings of fretting fatigue. This paper will investigate the change of stick-slip boundary in teeth flank of spline shaft-hub connections (SHC) under different combinations of bending moment and fluctuating torque via FE simulation. The normalized parameters σ_{bn1} and τ_{tn1} are used to quantify the magnitude of bending moment and torque in order to eliminate the effects of specific spline SHC parameters in evaluation of different load effects. Results showed that the stick and stick-slip regions spread toward teeth root and the stick region disappeared under loading combination of smaller σ_{bn1}/τ_{tn1} ratio with the increase of τ_{tn1} ; The settled stick region disappeared when σ_{bn1}/τ_{tn1} =0.3-0.35, and it is more wear predominant when σ_{bn1}/τ_{tn1} ≥0.35.

1. INTRODUCTION

Fretting is defined as a contact damage process arising from surface micro-slip associated with small scale oscillatory motion of clamped structural members. Numerous studies, especially experimental studies have been done to investigate this problem (DA. Hills and A Mugadu 2002, DA. Hills and D. Novell 2006, M. Ciavarella and G. Demelio 2001, S. Fouvry, K. Kubiak, 2009). It remains a debate whether fretting fatigue is a surface damage phenomenon or a multi-axial fatigue phenomenon (P. M. WAVISH et al. 2009). Some of studies consider that it is a combination of surface damage associated with micro-slip and fatigue associated with multi-axial stress field, a number of recent efforts have been made to characterize and quantify the interaction between surface damage and multi-axial fatigue (J. J. Madge et al.2007, J. Ding et al. 2008, J.J. Madge et al. 2008).

It is generally observed in fretting contact interface that gross slip gives high energy dissipation and high wear volumes whereas partial slip produces less energy dissipation and smaller wear volumes. However, the former leads to infinite or longer fatigue life (namely fretting wear in the absence of the bulk stress under the contact interface), the later leads to shorter fatigue life and comes to fretting fatigue with the presence of cyclic bulk stresses superimposed on one or both of the contacting components. (O. Vingsbo and S. Soderberg 1988, S. Fouvry et al. 2004). The relationship between fatigue life and wear rate as a function of slip amplitude set up by Vingsbo and Soderberg clearly reveals this phenomenon, as shown in Fig. 1 (O. Vingsbo and S. Soderberg 1988). L. Gallego et al. also presented a similar curve which shows the relationship between tangential force and tangential displacement, three different regimes are distinguished by function of normal load and tangential displacement: the stick regime, corresponding to very limit surface damage; the mixed stick-slip regime, corresponding to cracks; and the gross slip regimes corresponding to wear (L. Gallego et al. 2006). In most cases, both wear and fatigue coexist within the same contact interface, but one of which will be the dominant failure mechanism depending on the loads. Wear

introduced by frictional slip may accelerate crack initiation and produce fretting fatigue combined with cyclic bulk stresses or worn away micro cracks below the surface before propagation, namely, the accelerating effect and the arresting effect of frictional slip. So, it is worth to ascertain a threshold between fretting wear and fretting fatigue in order to avoid unexpected fracture caused by fretting fatigue. As an implication of this conception, J. Ding introduced a frictional work threshold between fretting wear and fretting fatigue in his mSWT fretting fatigue prediction model (J. Ding et al. 2007); Oldendorf, U. developed a stress concentration factor by fitting experimentally obtained damage-slip and damage-surface pressure curves (U. Oldendorf 2003). As well as fretting in railway axles and dovetail roots of compressor blades, fretting damage of spline shaft-hub connections (SHC) arouse more and more attentions in the last decades and becomes a major concern for spline SHC in many applications such as gas turbine engines, gearbox of wind turbine and gas compressor et al. This paper is aimed to indirectly ascertain the load threshold between fretting wear and fretting fatigue in spline SHC by tracing the evolution of contact status under alternating combination of torque and bending moment.

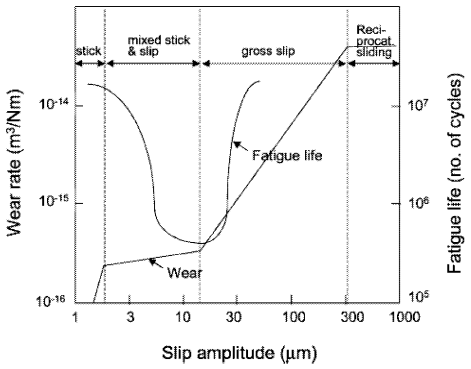


Fig.1.Typical relationship between fatigue life and wear rate as a function of slip amplitude, after Vingsbo and Soderberg (O. Vingsbo and S. Soderberg 1988).

2. FINITE ELEMENT OF SPLINE SHC

2.1. Finite element model and loads

All numerical simulations described in this paper were done with FE software ANSYS version 12.0, performed as transient dynamic analysis. The FE model and boundary conditions are showed in Fig.2. To avoid vague because of too much meshes, Fig.2 (a) shows only the mesh topologies of the whole spline SHC model, but Fig.2 (b) clearly shows the local enlargement of one mesh refined teeth pair. This mesh refinement approach permits a fine FE meshing on one tooth pair to be comparatively large reduced over the adjacent regions but without too many bad shaped elements in the adjacent regions. Eight-node linear brick elements SOLID45 are used throughout the whole model. Because no plastic deformation is predicted to occur for the range of load cases analyzed here, elastic material models are used in the analysis (E-Modul: 212000MPa, Poisson's ratio: 0.29). In the FE model, torque (T) and bending moment (B) are converted to forces and applied at the cylindrical surface of the hub. The torque fluctuates between 80%*T0 and T0 during the cycle loading. The bending moment +B and -B, corresponding to the maximum and minimum total torque values respectively, is applied at a constant magnitude and rotates around the axis of the shaft. This is implemented via two components, B1 and B2, which are 90^0 out of phase with one another (J. Ding et al. 2007). Referring to Fig. 2, with B as the total bending moment, t as the time increment and f as the frequency of rotation, B1 and B2 are defined respectively by:

$$B_1 = -B\sin(2\pi ft) \quad (1)$$

$$B_2 = B\cos(2\pi ft) \quad (2)$$

Fig 3. shows the load sequence of the FE model, which consists of 9 steps，at 1^{st} step, torque and bending moment are applied by ramping up linearly from zero to their respective begin values in cycle ,and the rest 8 steps simulates one cycle loading. For the contact between flanks of shaft and hub, CONTA174 element is used to represent contact and sliding between 3-D "target" surfaces (flank surface of hub) and a deformable surface defined by this element. Contact occurs when the contact element surface penetrates the target segment elements, represented by TARGE170 covered on the target surface. In the FE model the basic Coulomb friction model with isotropic friction is employed. Frictional contact conditions are determined via the augmented Lagrange method. According to (S. Medina and A.V.Olver 2002) and (C.H.H. Ratsimba et al. 2004), a median coefficient of friction (COF) value-0.3 is used in this study. Of course, COF can be expected to vary with stroke and load, but it remains an open problem in fretting contact.

The spline SHC used in simulation is DIN5480 45x2x21 according to the standard DIN5480 (German institute of standard. 2006). The non-dimensionalised parameters σ_{bn1}, τ_{tn1} and σ_{bn1}/τ_{tn1} are used to quantify the magnitude of

bending moment, torque and the ratio between them, so that the effects of specific spline SHC parameters are eliminated in evaluation the effects of different loads. σ_{bn1} and τ_{tn1} are maximal normal stress under bending moment and maximal shear stress under torque within equivalent cylinder shaft respectively, as shown in Fig.4, they are calculated according to the standard DIN5466 (German institute of standard. 2006).

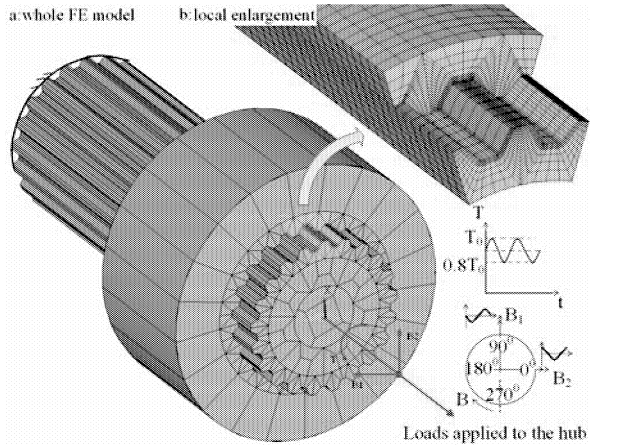


Fig 2.FE model of the spline SHC (DIN5480 45x2x40x21)

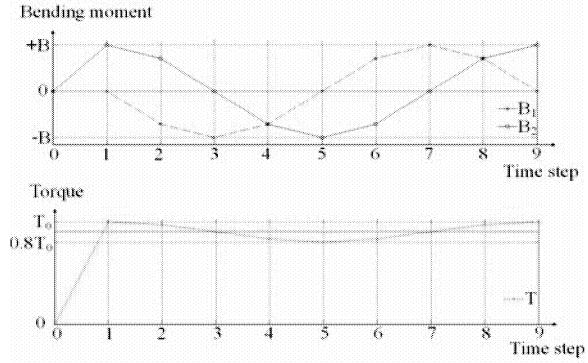


Fig.3.Schematic diagram of loading sequence applied to the FE model

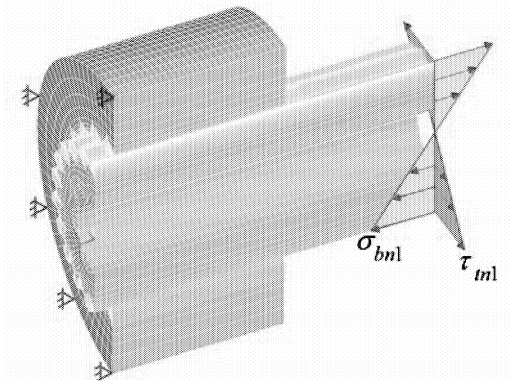


Fig.4. Non-dimensionlise of torque and bending moment

2.2. Post processing of FE analysis results

For the convenience of post processing and results comparison, all the nodal results of contact elements on surface of one mesh refined spline flank are read out from ANSYS and post processed in software package MATLAB version 2009b. Variation of instantaneous contact statuses are mapped to two-dimensional coordinate system, shown in Fig.5, the X and Y coordinates are normalized by dividing lengths along axial direction and involute profile respectively.

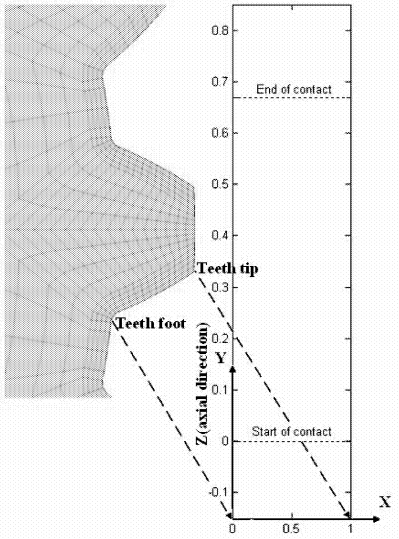


Fig.5. Map of FE analysis results from flank surface to two-dimensional coordinate system

3. NUMERICAL SIMULATION RESULTS

The values and their corresponding contact status defined in ANSYS are as followings: 0 = Open and not near contact, 1 = Open but near contact, 2 = Closed and sliding, 3 = Closed and sticking. All the nodal results of contact status on surface of one mesh refined spline flank are read out from ANSYS and post processed in software package MATLAB. The boundary of stick region and stick-slip region will be found out in post process. Different from the definition of transient contact status, here the stick region and stick-slip region defined like this: for a certain node, when the minimal value of contact status≥3 in a cycle, then this node is always in stick status, collection of all this kind of nodes composed the macro stick region; when the maximal value of contact status≥3 & minimal value of contact status≥2 in a cycle, then this node is defined as stick-slip node, collection of all this kind of nodes composed the macro stick-slip region. According to the mapping concept introduced by Vingsbo etc. (O. Vingsbo and S. Soderberg 1988) and Gallego, L. et al. (L. Gallego et al. 2006), as shown in Fig.1,

cracks will more possibly to occur in the macro stick-slip region defined above. So, when there's no settled boundary of stick-slip region and stick region under certain loading situations, namely, micro cracks below the surface will be worn away by frictional sliding before propagation, the arresting effect of frictional sliding will be predominant in this case. Fig.6 shows the changes of stick and stick-slip regions under alternating loadings. The simulation results showed that the risk of fretting fatigue decrease with the increase of σ_{bn1}/τ_{tn1} ratio, when $\sigma_{bn1}/\tau_{tn1} \geq 0.4$, it is more wear predominant in the spline flank; And with the increase of τ_{tn1}, namely the torque, the stick and stick-slip regions spread toward the direction of teeth root, the stick regions disappear at smaller σ_{bn1}/τ_{tn1} ratio loading state; the axial positions of stick and stick-slip regions' boundary remain close to the start of contact (see Fig 5.). Further simulations focus on the σ_{bn1}/τ_{tn1} ratio interval between 0.2 and 0.6. The simulation results showed in Fig.7, which showed that the stick region disappeared at σ_{bn1}/τ_{tn1} ratio interval between 0.3 and 0.35 when τ_{tn1}=120MPa.

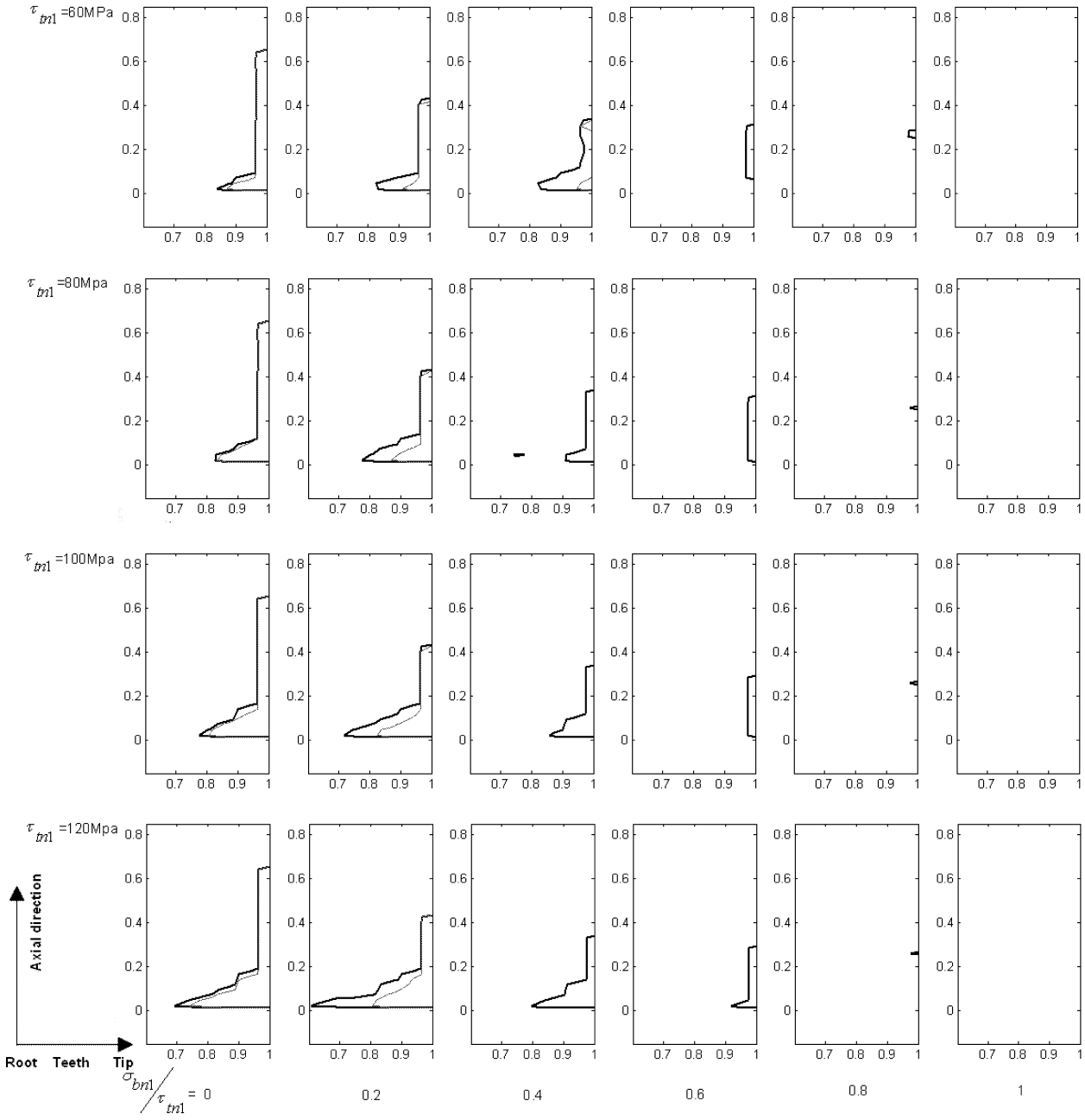


Fig.6. Change of stick and stick-slip regions under alternating loadings-1($\sigma_{bn1} = M_{b1}/W_{b1}$, $\tau_{tn1} = T/W_{t1}$ according to DIN5466, μ = 0.3; Blue lines-boundary of stick region, Black lines-boundary of stick-slip region)

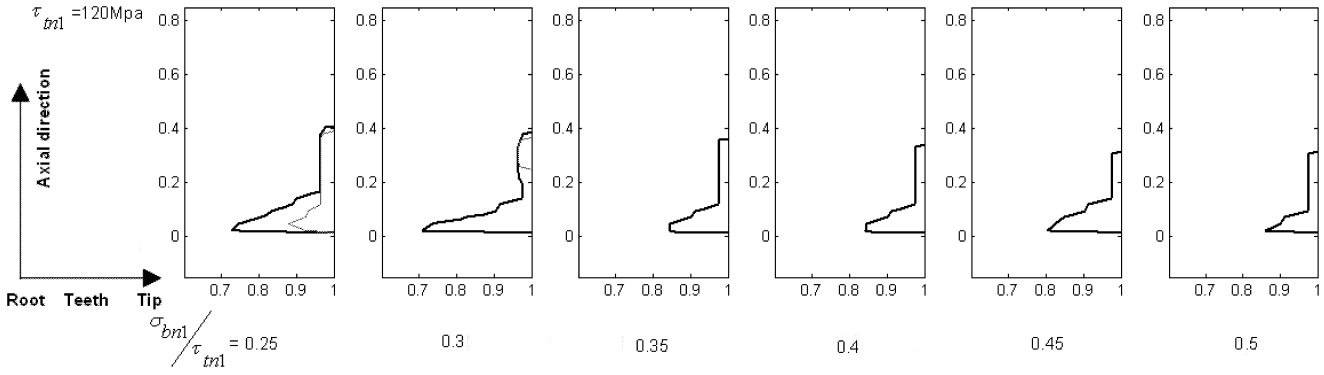


Fig.7. Change of sticking and sticking-sliding regions under alternating loadings-2($\sigma_{bn1} = M_{b1}/W_{b1}$, $\tau_{tn1} = T/W_{t1}$ according to DIN5466, μ = 0.3; Blue lines-boundary of stick region, Black lines-boundary of stick-slip region)

4. DISCUSSION AND CONCLUSION

It is well known that partial slip between contact interface leads to shorter fatigue life and comes to fretting fatigue with the presence of cyclic bulk stresses superimposed on one or both of the contacting components, the threshold between fretting wear and fatigue is also clearly expressed by micro slip amplitude in Fig 1(about 20μm).But, it is not always possible to measure and obtain this kind of microscopic threshold due to the complicated geometry and loadings in practical application. Corresponding to this kind of microscopic threshold, this paper try to develop a macroscopic threshold between fretting wear and fatigue by tracing the evolution of contact status under alternating combining loads of torque and bending moment via intensive FE simulations. The results showed that the stick and stick-slip regions spread toward teeth root and the stick regions disappear at smaller σ_{bn1}/τ_{tn1} ratio loading state with the increase of τ_{tn1} , namely the torque; The settled sticking region disappeared between σ_{bn1}/τ_{tn1} interval 0.3-0.35, and it is more wear predominant when $\sigma_{bn1}/\tau_{tn1} \geq 0.35$. This kind of macroscopic threshold provides more practical instructions in fretting fatigue prediction of spline shaft-hub connection but needs further experimental validation in future works.

REFERENCES

C.H.H. Ratsimba et al. 2004. “Measurement, analysis and prediction of fretting wear damage in a representative aeroengine spline coupling”. Wear 2004(257), 1193–1206.

D A. Hills and A. Mugadu 2002. “An overview of progress in the study of fretting fatigue”. JOURNAL OF STRAIN ANALYSIS, 2002. 37 (6), 591-601.

D A. Hills and D. Novell 2006. “Recent developments in the understanding of fretting fatigue”. Engineering Fracture Mechanics, 2006, 73(2), 207-222.

German institute of standard. 2006. “DIN5480: Involute splines based on reference diameters”. Berlin.

German institute of standard. 2006. “DIN5466: Splined joints, calculation of load capacity”. Berlin.

J. J. Madge et al. 2007, “Contact-evolution based prediction of fretting fatigue life: Effect of slip amplitude”. Wear, 2007. 262(9-10), 1159-1170.

J. Ding et al. 2007. “Fretting fatigue predictions in a complex coupling”. International Journal of Fatigue 2007(29), 1229–1244.

J. Ding et al. 2008. “Finite element simulation of fretting wear-fatigue interaction in spline couplings”. Tribology, 2008. 2 (1), 10-24.

J.J. Madge et al. 2008. “A combined wear and crack nucleation-propagation methodology for fretting fatigue prediction”. International Journal of Fatigue, 2008. 30(9), 1509-1528.

L. Gallego. et al. 2006. “A Comprehensive Method to Predict Wear and to Define the Optimum Geometry of Fretting Surfaces”. Journal of Tribology, 2006. 128(3), 476-485.

M. Ciavarella and G. Demelio 2001. “A review of analytical aspects of fretting fatigue, with extension to damage parameters, and application to dovetail joints”. International Journal of Solids and Structures, 2001 (38), 1791-1811.

O. Vingsbo and S. Soderberg 1988. “On fretting wears”. Wear 126 (1988), 131–147.

P. M. WAVISH et al. 2009. “A multiaxial fretting fatigue test for spline coupling contact”. Fatigue & Fracture of Engineering Materials & Structures, 2009(32), 325-345.

S. Fouvry and K. Kubiak 2009. “Introduction of a fretting-fatigue mapping concept: Development of a dual crack nucleation – crack propagation approach to formalize fretting-fatigue damage”. International Journal of Fatigue, 2009(31), 250–262.

S. Fouvry et al. 2004. “A quantitative approach of Ti–6Al–4V fretting damage: friction wear and crack nucleation”. Wear 257 (2004), 916–929.

S. Medina and A.V. Olver 2002. “Regimes of Contact in Spline Couplings”. Journal of Tribology, 2002. 124, 351-357.

U. Oldendorf 2003. “Passfederbeanspruchungen-Ermittlung der Duerhaltbarkeit von Passfeder- verbindungen”. FVA-Forschungsheft 691, Frankfurt, 2003. (in German)

DESIGN OF NOVEL SELF-ORGANIZATION MECHANISMS THROUGH SIMULATION

Marton Legeny and Borbala Katalin Benko
Department of Telecommunications
Budapest University of Technology and Economics
Magyar tudosok korutja 2
H-1117, Budapest, Hungary
E-mail: {legeny,bbenko}@hit.bme.hu

KEYWORDS

Clustering, Self-organization, Emergence, Spyglass, Simulation of distributed algorithms

ABSTRACT

The ability of self-organization is a common requirement in Ambient Intelligence systems; however, the design and evaluation of these highly distributed algorithms is not trivial due to their emergent properties, and also because of the diversity of the problem space.

This paper summarizes the lessons learnt during the design and evaluation of a novel self-organization algorithm (Spyglass). The emphasis is not solely on the algorithm itself, but also on the road to that led to creating it. We describe a simulation based methodology for designing emergent self-organiztaion mechanisms; tackled with considerations on the simulation and evaluation environment including metrics, problem space coverage, and a novel data mining based meta-analysis for simulation results. All aspects are illustrated with results from Spyglass.

INTRODUCTION

Ambient Intelligence, Self-Organization

In Ambient Intelligence or Pervasive Computing scenarios (Cook et al. 2009; Augusto 2007), multitudes of networked elements provide the user with services in a way that makes use of the distributed yet networked setting. Elements of the network themselves and so their services may be very diverse, and may also be highly dynamic in time. The ability of automatic self-organization is becoming one of the basic requirements in pervasive environments, as networked elements need to cooperate, share resources, communicate, discover each other effectively, without help of an external 'network manager'.

Self-organization often uses simple algorithms that have emergent properties, i.e. the multitude of executions result in an 'intelligent', high-level behaviour. Algorithms are often inspired by biological, chemical or physical phaenomenons, like swarms, insect colonies, human brain, mixture of chemicals or the immune system. These paradigms were used in numerous ways to solve different problems regarding computer networks, including load balancing (Di Caro et al. 2005; Montresor et al. 2002; Montresor et al. 2003; CASCADAS 2007). Self-organization in overlay networks (Jain 2000; Apel and Böhm 2005) is also widely used for deploying distributed applications (mostly used in P2P data sharing systems) without the need of a supervising entity.

Our work focuses on biologically inspired, fully distributed self-organization algorithms for large overlay networks, with an emphasis on load balancing. In this paper we describe the *design and evaluation methodology* we used during the creation of a new self-organization algorithm (called Spyglass); with strong emphasis on the *simulation environment* itself.

Problems of Design and Evaluation of Algorithms in Large Networks

The design and evaluation of distributed self-organization mechanisms for large networks is non-trivial.

As self-organization emerges from the multitude of simple mechanisms, it's often hard to foresee–or even understand– the effect of a small change in the algorithm without experimenting. So, even during design, a tool that enables the creation of test cases with step-by-step execution is critical.

Evaluation in real-life or near-real-life environments is often impossible, or at least cumbersome. It's hard to get even hundreds of people to install a new software just to provide us with measurement results. Usage of testbed may not be a resolution either, as testbeds typically provide fixed network size and architecture, while the excessive evaluation of an algorithm requires very high dynamism (various network size, connectivity and density settings, coverage of extreme cases, and so on).

Simulation is a possible alternative solution.

SIMULATION OF SELF-ORGANIZATION MECHANISMS IN LARGE NETWORKS

Methodology

We believe that simulation is inevitable in the process of designing and evaluating algorithms with emergent properties, especially in case of large distributed networks.

By simulator system we mean a rich environment consisting of a simulator engine and a set of strongly coupled modules that help the process. For a simulator system, the following roles can be distinguished.

- Experimentation facility at design time. The designer of the algorithm uses relatively simple network settings to experiment with and understand the emergent effects of an algorithm.
- Tool for evaluation and proof. The final algorithm is executed on a set of cases; measurement results are collected and evaluated.
- Closed loop model: source of feedback for the designers. The user may get an immediate, first-impression feedback about the algorithm in an easily interpretable way, while collected results may be subject to deeper analysis. The designer uses this feedback to refine the algorithm, thus closes the design loop.

The above roles indicate our design methodology, which is a loop: (1) simulation, (2) evaluation, and (3) re-design based on the evaluation results; (4) repeating until the evaluation shows that the algorithm is either good or is unable to improve any further. The following chapter showcases how a scenario – based on this methodology – works in practice.

Case Study: Design of the Spyglass Algorithm

Motivation and basic idea

Existing clustering methods use rather simple steps, based solely on local information, to achieve a global, self-organizing behaviour. While simplicity ensures very low processing power and data storage requirements (making self-organization possible even in sensor networks (Krishnan 2004; Gburzynski et al. 2007), where preserving battery life is of high priority); there are certain common disadvantages:

- *Clustering is slow*. The algorithm doesn't create enough new links, which means that the reorganization process is slow in reacting to the demands.
- *Clusters are not large enough.* Most of these methods use quite strict limitations when inspecting neighbors for possible link reorganization; which, given an unfortunate network topology, could lead to the build-up of small and ineffective clusters.

In order to achieve speedup and increase the cluster size, we introduced a novel algorithm, called Spyglass. The initial version of Spyglass was basically a modified version of on-demand clustering (CASCADAS 2006) but with a highly increased number of neighbors to be tested: if no proper node (a *candidate*) was found among the match-maker's neighbors, *their* neighbors get checked too (that's where the name comes from – Spyglass 'sees further' than the basic algorithm).

Evaluation through simulation 1

Spyglass and the original self-organization mechanisms – to serve as reference – were implemented and tested in our simulation system.

Results confirmed the improvement in speed and cluster size but also shed light on disadvantages. Spyglass, in worst-case scenario, uses *O(nm)* messages to check all possible nodes in search for a candidate (where we assume that a match-maker has *n* neighbors, with them having an average of *m* neighbors) – thus, uses significantly more messages than on-demand clustering; causing a serious communication overhead, consuming a large amount of network bandwidth.

Refinement based on the results

Since communication overhead is considered to be one of the most significant problems in today's networks and distributed applications, at this point, we decided to resolve the problem by drastically decreasing the number of messages sent by the algorithm. The goal was to do that in a way that preserves the efficiency of cluster creation.

The idea was to use a temporary storage–a Neighbor Cache– on all nodes to store their neighbor list (in other words, we're doing *topology caching*) so that when an adjacent node asks for a candidate we can look it up faster in the cache (which is also sorted by node type). This technique is similar to the one used by DNS caching – non-authoritative responses from cache, authoritative responses from properly checking all neighbors (Mockapetris and Dunlap 1988), and also to the method used in Freenet (Clarke et al. 2000). The build-up of a Neighbor Cache is expensive in terms of messages, but once ready, it serves queries for „free" (with zero communication).

When the topology changes around a node (i.e. it loses existing neighbors or gets new ones), the Neighbor Cache gets updated (active caching).

The goal of the Neighbor Cache was to decrease the number of messages sent during operation, while maintaining the effective clustering capability of the first Spyglass variant.

Evaluation through simulation 2

Once again, simulation was used to evaluate the refined Spyglass. Results proved (see details later) that caching resolved the communication overhead problem; and the resulting system is close to the first Spyglass version in efficiency.

Concluding, this chapter shows how simulation was used while designing emergent networked algorithms: for evaluation, refinement, and getting proof. The next chapter summarizes what kind of requirements the simulation environment should fulfill in order to enable and efficiently support this process.

CONSIDERATIONS TOWARDS THE SIMULATION ENVIRONMENT

Open Set of Metrics

The simulator system should enable the collection of arbitrary information about network components, their communication, and the superset of the previous two.

When designing the simulation environment, our focus was on implementing a system which is able to measure a large and open set of factors. A few examples for each type:

- *Component (Node and Link) related factors*. Number of nodes and links (over time), neighbors per node (over time), job consumption, queue lengths, production factors

of Workload Generators, capacity of Load Balancer Units.

- *Communication and messaging related factors*. Number of messages generated, number of jobs transferred.
- *Metafactors*. Cluster membership, cluster size and build-up time, cluster build-up characteristics (over time), number of unprocessed jobs in a system. Please note that these measures are one level above the previous one, for example cluster size reflects to a superset of nodes.

Openness and configurability are vital. While most of these measures are generally present in all overlay networks (i.e. node / link and communication related information), others are specific to certain simulated algorithms. For example, we defined „relevant cluster" as a set of available matching nodes per Workload Generator (to measure how well a Workload Generator's output is handled), but this metric is meaningless when load balancing is switched off.

Multi-Purpose Support

A truly multi-purpose, and easy-to-use simulator system should be able to:

- *Provide a graphical user interface* that is able to visualize the simulation steps (and also provide a way to conveniently edit network and algorithm properties) to help understanding or demonstrating the self-organization mechanism.
- Have *a scriptable console interface* for batch and large-scale tests, e.g. over 100.000 nodes (visualization of the process is not a requirement here).
- *Display simulation results* in a clear format, using raw data and the aforementioned metrics. Automatic visual feedback (graphs) for the most exciting measures was found to be of great help.

These requirements suggest a rich toolset built around a solid simulation core. All algorithms we used were implemented independently of the user interface. We separated the algorithm core, the user interface and the collection/post-processing of results; which resulted in a flexible yet robust multi-purpose system. The simulator was written in Java language to ensure portability and a simple but meaningful GUI for anyone.

Close-to-reality Model

As the purpose of simulation is to replace traditional evaluation methods such as testbeds, the ability to run algorithms on a close-to-reality model is vital. This includes two aspects: detailed world modeling and realistic algorithm code. In other words, when evaluating, we're interested in the algorithm itself, and not in a high-level abstraction of the algorithm where neglection and abstraction may easily bias the results.

The common model we used for self-organization algorithms for load balancing is as follows. *Jobs* enter the *overlay network* through specific nodes, called *Workload Generators*. *Nodes* consume jobs from their local *queue*. Workload gets distributed in the network, when nodes share jobs with their *neighbors*. Job transfer may be bound to specific conditions (e.g. to a minimal queue length). When a node finds no appropriate neighbor to share its load with, a clustering algorithm is initiated to reorganize the overlay topology and create a new, suitable *link*.

We strongly believe that the aforementioned model is in good correlation with real-life overlays used for load balancing. Nonetheless, if needed, this core module can be extended to meet custom needs.

Our network description model was designed – despite the simplicity of the textual format – to be rich in details, with numerous (more than sixty) node-, link-, and algorithm-related parameters to set; which ensures simulations to run on a model that is as close to real life as possible.

Customized Output Generation

Aside from presenting simulation results in a graphical user interface, creating „processable" (e.g. by means of data mining or visualisation software) output was required. Raw measurement data – in an easy-to-use text format – is provided, including detailed cluster information (cluster members, node degrees), changes in node queues and also in the network topology over time.

About the most important metrics, a direct visual interpretation is generated automatically in order to provide the user with a first impression of feedback – such as node degree distribution graphs, local queue history graphs and aggregated informations (average degree, cluster size, number of messages). This extract may also include simulation-specific metrics.

Customizability of not only the contents of output files but also the format of the file is a key feature in our simulator.

Tools for Problem Space Coverage

The performance of algorithms may largely depend on the simulated setting, so realistic, fair and typical network configurations / user behaviours (e.g. workload generation) should be used during the evaluation. However, it's hard to tell what a typical network or user behaviour – so, a typical problem case – looks like. On the other hand, it may be wise to test the algorithms, after normal situations, also for the extremities.

A good simulation system should include tools for automatic generation of problem cases in a way that the theoretical problem space is sufficiently covered. Without that, results may be biased or unbalanced. In case of Spyglass we used an automatic problem space generator which generates random and scale-free graphs of the given size (using the Erdos-Renyi method (Erdos and Renyi 1960)) according to the actual density, link distribution, workload generation and capacity parameters.

Interfaceability Towards Data Mining Systems

In complex cases when there're multitudes of algorithm settings (and network layouts), it's often hard to find the best parameters for a model. For example, different algorithm parameters may be optimal for a small, dense network and for a large, sparse network.

We used data mining for the meta-analysis of a large number of simulation results. The process flow is the following:

(1) A large number of simulations is performed. (2) Parameters (algorithm settings, problem case settings) and results (extracts of results) are loaded into a database. (3) Data mining (classification and regression model, association rule search) is used to identify trends and patterns.

When all simulations belong to a single algorithm, data mining can be used for determining the optimal algorithm parameters. An advantage is, that this kind of analysis may not only result in concrete values, but may also provide trends (e.g. larger networks require larger timeout). We used regression and classification models for this task.

Another stage is the objective comparison of algorithms. In this case the simulation data belongs to different algorithms, and data mining helps in comparing them, finding general trends and general or local facts. We used classification, regression and frequent item search in case of Spyglass.

RESULTS

This chapter summarizes the results regarding the Spyglass algorithm.

A large number of batched simulations were executed to get an objective picture about the performance of the two Spyglass variants (without and with caching) compared to the reference algorithms.

Although the simulation included multitudes of settings, for this paper, we chose one concrete setting to showcase the results: a random graph network topology (generated with the Erdos-Renyi method), consisting of 1000 nodes and 2565 links, with 5% of the nodes being Workload Generators. The number of maximum links in the network was set to 3000, and the algorithm runtime was maximised in 500 steps.

Comparison included three reference models: On-Demand Clustering (ODC), Passive Clustering and No Clustering (Di Nitto et al. 2008), plus the two Spyglass variants: the first algorithm version (Spyglass 1) and the refined, active Neighbor Caching version (Spyglass NC). As Passive Clustering and No Clustering significantly underperformed ODC in all aspects, we excluded them from the figures.

Here, we characterize the goodness of an algorithm with its clustering speed, number of clustered nodes, cluster size, and number of communication messages. During evaluation, other metrics were also used, like shape of queue length curves, analysis of peak load regions, etc.

Clustering speed is measured with the number of new (useful) links created in the specified time frame. Fig. 1 shows that, as stated before, on-demand clustering is, in general, a slow method for building up clusters–Spyglass 1 is more than 6 times faster. When using Neighbor Cache, clustering speed slightly decreases, but Spyglass NC is still 251% better than the reference algorithm.

Another key feature of self-organizing algorithms is the number of clustered nodes in the overlay network – which depicts 'useful' nodes only – those, which are available from a matching Workload Generator. Results are also shown in Fig. 1. Again, we can see that Spyglass 1 has a 50% advantage over on-demand clustering, while Spyglass NC manages to beat its competitor by 42%.

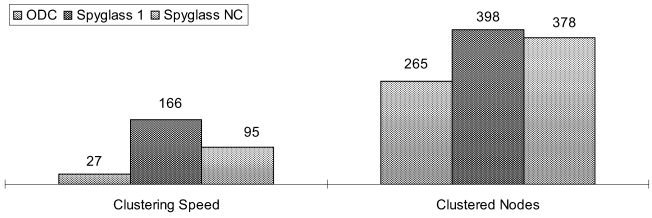


Fig. 1: Clustering Speeds and Number of Clustered Nodes

The size of clusters can be characterized with maximum or average values, shown in Fig. 2. Once again, a significant increment can be observed in the clustering capability, as Spyglass 1 [Spyglass NC] has an 80% [50%] advantage over ODC in maximum and 50% [42%] in average cluster size. Larger clusters mean better load balancing.

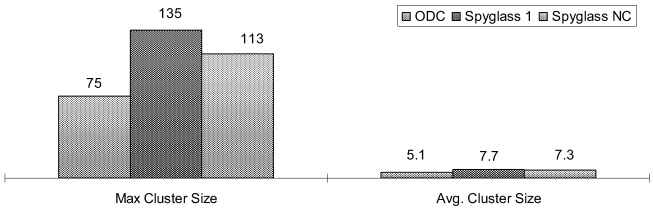


Fig. 2: Maximum / Average Cluster Size After Simulation

Since the number of clustered nodes is probably the most important descriptor of a self-organizing algorithm (this shows how many processors work up the job), simulation results suggest that we managed to create a method that shows clear advantage over 'traditional' clustering mechanisms. The advantage is confirmed by the cluster size measurements, too.

Fig. 3 shows the total number of unprocessed jobs in the network during the simulation, which basically defines the „goodness" of a load balancing algorithm (when sharing the load, the queue is processed faster). Both Spyglass 1 and Spyglass NC have a significant advantage over ODC: the number of unprocessed jobs in the system descents to five 60-70% faster (the last few jobs may take irreally long to process due to the local lack of matching neighbours – hence, a small but nonzero job count is a better descriptor). The graph also shows that Spyglass NC requires longer bootstrap, but once the Neighbour Cache is ready, becomes faster than Spyglass 1.

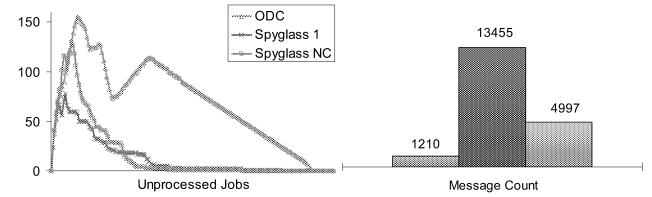


Fig. 3: Num. of Unprocessed Jobs / Num. of Messages Sent

Finally, also on Fig. 3, we can see the total number of messages sent during the simulation, i.e. the communication overhead. As expected, on-demand clustering, while being the least effective in building clusters, also uses the lowest number of messages among the algorithms. Spyglass 1 – as expected – causes a huge increment in the number of messages, while Spyglass NC manages to overcome that problem by decreasing the communication overhead by

170%. A large portion of Spyglass NC's messages is used for cache initialization, which is a one-time cost. The resulting behaviour is acceptable for our goal.

Large-scale tests were also run to analyze the behaviour when the network size increases. Two topologies were used: one consisting of 10.000 nodes and 25.125 links (10k setting) and another with 25.000 nodes and 62.661 links (25k setting). In both cases 5% of the nodes were Workload Generators; maximum number of links in the network was set to 100.000, with algorithm runtime maximised in 5000 steps.

Tests confirmed the stability of our methods for large topologies. All properties of the algorithm were found to scale well (results are shown in Fig. 4).

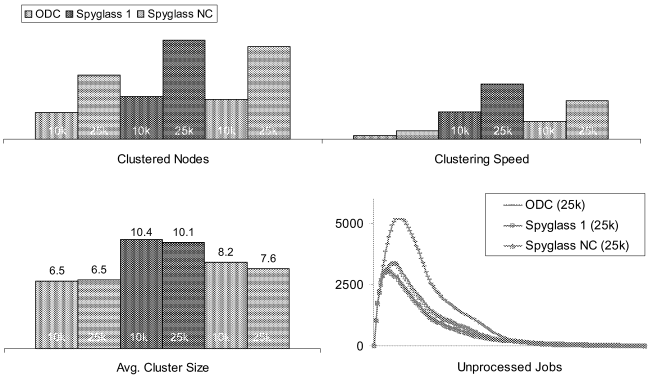


Fig. 4: Clustered Nodes, Clustering Speed, Average Cluster Size, and Unprocessed Jobs in 10k and 25k Networks

The average cluster size stays quasi constant for each algorithm. Compared to the previous 1000 node setting, a slight increment can be observed, due to the more common upspring of large clusters (breaking the 'localty' barrier) which shifts the average. The number of clustered nodes and the clustering speed is linear with the network size. The advantage of Spyglass over ODC is also preserved in the analysis of job count; however, the local lack of matching capacity slows down all algorithms after a while, resulting in a long tail. Spyglass NC's communictaion overhead remained in the acceptable range.

CONCLUSION

In this paper we described the design and evaluation process of a novel self-organization algorithm (Spyglass) that speeds up cluster creation and increases average cluster size compared to reference mechanisms, while keeping the communication overhead under control.

Besides the particular algorithm, we also tackled with general considerations. We discussed the role of simulation in the design and evaluation of emergent mechanisms in large distributed networks; described a simple feedback loop-based design and evaluation mechanism; identified key requirements for a simulation environment in order to enable and effectively aid humans in the process. During that, we also drafted the direction of data mining based meta-evaluation of simulation results for parameter optimization and for fair comparison of algorithms – a tool that shall be of great help in better understanding the simulation results in ambiguous cases.

ACKNOWLEDGEMENT

We would like to thank BME-HIT for funding our research. Special thanks goes to Fabrice Saffre (from British Telecom at that time) for leading us into the world of self-organization algorithms.

REFERENCES

Apel, S. and K. Böhm. 2005. „Self-Organization in Overlay Networks." *Proceedings of CAISE'05 Workshops (Workshop on Adaptive and Self-Managing Enterprise Applications)*, Vol. 2, 139-153.

Augusto, J. C. 2007. „Ambient Intelligence: the Confluence of Ubiquitous / Pervasive Computing and Artificial Intelligence." *Intelligent Computing Everywhere*, Springer Verlag, 213-234.

Clarke, I.; O. Sandberg; B. Wiley; and T. W. Hong. 2000. „Freenet: A Distributed Anonymous Information Storage and Retrieval System." *Workshop on Design Issues in Autonymity and Unobservability*, Berkeley, CA, USA.

Cook, D. J.; J. C. Augusto; and V. R. Jakkula. 2009. „Ambient Intelligence: Technologies, Applications, and Opportunities." *Journal of Pervasive and Mobile Computing*, 5(4): 277-298.

Di Caro, G.; F. Ducatelle; and L. M. Gambardella. 2005. „Swarm Intelligence For Routing In Mobile Ad Hoc Networks." *Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS)*, pp. 76-83.

Di Nitto, E; D. J. Dubois; R. Mirandola; F. Saffre; and R. Tateson. 2008. „Self-Aggregation Techniques for Load Balancing in Distributed Systems." *SASO 2008*, 489-490.

Erdős, P. and A. Rényi. 1960. „The Evolution of Random Graphs". *Magyar Tud. Akad. Mat. Kutató Int. Közl.* 5: 17-61.

Gburzynski, P.; B. Kaminska; and W. Olesinski. 2007. „A Tiny and Efficient Wireless Ad-hoc Protocol for Low-cost Sensor Networks." *Proceedings of the conference on Design, Automation and Test in Europe (DATE'07)*, 1557-1562.

IST CASCADAS *Deliverable D3.1*: „Aggregation Algorithms, Overlay Dynamics and Implications for Self-Organised Distributed Systems." IST CASCADAS Project, 2006.

IST CASCADAS *Deliverable D3.2*: „Report on rule-based modules for unit differentiation using cross-inhibition and/or resource competition." IST CASCADAS Project, 2007.

Jain, S.; R. Mahajan; and B. Niswonger. 2000. „Self-Organizing Overlays." *Technical report*, University of Washington, DC, USA.

Krishnan, R. 2004. „Efficient Self-Organization of Large Wireless Sensor Networks". *Ph.D. Dissertation*, Boston University, College of Engineering.

Mockapetris, P. V. and K. J. Dunlap. 1988. „Development of the Domain Name System." *Proceedings of SIGCOMM'88*, *Computer Communication Review* Vol. 18, No. 4., 123-133.

Montresor, A; H. Meling; and Ö. Babaoglu. 2002. „Messor: Load-Balancing through a Swarm of Autonomous Agents." *Proceedings of 1st Workshop on Agents and Peer-to-Peer Computing (AP2PC 2002)*, Springer Verlag, 125-137.

Montresor, A; H. Meling, and Ö. Babaoglu. 2003. „Toward Self-Organizing, Self-Repairing and Resilient Distributed Systems." *Number 2584 in Lecture Notes in Computer Science*, Springer Verlag, 119-124.

SIMULATION TOOLS

WAVEFORM RELAXATION FOR THE SIMULATION OF COMPLEX HYBRID SYSTEMS IN SCICOS

Damien Chapon
Airbus Operation S.A.S.
316 route de Bayonne
31060 Toulouse Cedex 09
France
damien.chapon@airbus.com

Ramine Nikoukhah, François Delebecque
INRIA-Rocquencourt
Domaine de Voluceau
78153 Le Chesnay Cedex
France
{ramine.nikoukhah,
françois.delebecque}@inria.fr

KEYWORDS
Waveform Relaxation, Hybrid systems, Scicos.

ABSTRACT

The Waveform Relaxation methods are a family of techniques for solving large systems of ordinary differential equations. They are based on the partitioning of a main system into weakly coupled subsystems. The subsystems are then solved independently, either sequentially or in parallel, and the coupling is realized by an iterative procedure. These techniques have been efficiently applied for the simulation of large problems arising from electrical network modelling. We propose in this paper an application of these methods to the simulation of complex hybrid systems arising from industrial applications. This application is performed with the capabilities of Scicos, a toolbox of the scientific computing software ScicosLab for the modelling and simulation of hybrid systems. For this purpose we have developed new Scicos blocks and a new simulation procedure for managing the events and the reconfigurations that events trigger during the integration of complex hybrid systems with a Waveform Relaxation method.

INTRODUCTION

Systems arising from aeronautic, spatial or automotive industrial applications are hybrid and complex. Hybrid means that those systems have variables evolving continuously and punctual events that create discontinuities. Complex means that they can be composed of stiff and non-stiff subsystems coming from different physical domains and that these subsystems can have different time scales. A typical example of such complex hybrid system is a hydraulic actuator connected to the hydraulic power supply system and actuating a physical mechanism, e.g. a landing gears door. Simulating such systems often leads to difficult numerical problems because the systems to be integrated are big and because the occurring physical phenomenon can have very different time scales. Therefore integrating such systems with classical numerical solving codes can be inefficient because the same numerical method and the same time steps are used to represent accurately all the system states. To tackle these problems an idea is to separate the main large system into subsystems having similar time scales and to treat more or less independently each subsystem with possibly different numerical methods and different time step strategies.

The Waveform Relaxation techniques are based on this partitioning idea. Indeed with these techniques the main large system is partitioned into several weakly coupled subsystems that are solved independently over a time window, either sequentially or in parallel, by guessing the behaviour of their surrounding subsystems. The coupling is realized by an iterative procedure. At the end of the subsystems integrations, each integrated waveform is sent to its coupled subsystems in order to update their guesses. The integration and waveform exchange steps are then reiterated until the total convergence of the solution on the time window. The advantages of these techniques are the integration of several small systems instead of the integration of a big system and the possibility to solve each subsystem sequentially or in parallel with eventually different numerical methods and different time step strategies. Obviously, the convergence speed of the Waveform Relaxation methods is faster when the couplings between the subsystems are weak. These methods have been introduced by Leralasmee et al (Leralasmee et al. 1982) and efficiently applied on problems arising from electrical network modelling (White et al. 1985; White and Siangiovanni-Vincentelli 1987). For a more recent review of these methods see Burrage's book (Burrage 1995).

We propose in this paper to apply the Waveform Relaxation methods to the simulation of complex hybrid systems arising from industrial applications. This application is realized by using the capabilities of Scicos (Campbell, Chancelier and Nikoukhah, 2006). Scicos is a toolbox of the scientific computing software ScicosLab for the modelling and simulation of hybrid dynamical systems. For this application we have developed within Scicos new simulation blocks and a new simulation procedure for simulating hybrid systems with the Waveform Relaxation techniques.

This paper is organized in two parts. The first one is devoted to the simulation of continuous systems with the Waveform Relaxation in Scicos. In this part, the new Scicos blocks that have been developed for exchanging waveforms and a generic procedure for using the ScicosLab workspace for the Waveform Relaxation methods are presented. The second part is devoted to the case of hybrid systems and on the new Scicos blocks that have been developed for using the Event-related dynamic partitioning procedure. This procedure is also presented in the second part and has been developed for managing the events occurring during the simulation of hybrid systems with the Waveform Relaxation techniques.

WAVEFORM RELAXATION FOR THE SIMULATION OF CONTINUOUS SYSTEMS IN SCICOS

Using the Waveform Relaxation methods for the simulation of a continuous system requires partitioning the main system into smaller subsystems, organizing the waveform exchanges between the subsystems and verifying their convergences. In order to illustrate how Scicos is used for this purpose, we use the following stiff nonlinear system of ordinary differential equations:

$$(1)\begin{cases} \dot{x} = \dfrac{q_1 y - xy + x(1-x)}{q_2} \\ \dot{y} = \dfrac{-q_1 y - xy + q_4 z}{q_3} \\ \dot{z} = x - z \end{cases}$$

With $x(0) = 0, y(0) = 1$ and $z(0) = 0.$

There is no numerical interest to use the Waveform Relaxation for this small system. It has only been chosen for purpose of illustration.

System partitioning

In order to be efficient, the Waveform Relaxation methods require partitioning the main system into weakly coupled subsystems. Obviously the convergence speed of these methods is faster when the couplings between the subsystems are weak. In a complex system, the separations often occur at the interface between the different physical equipments or the different functional parts of the main system. The partitioning requires lots of experience from the modeler and it is therefore more natural to do it manually than with automatic partitioning techniques. In our simple example, we partition the system into three subsystems, one for each equation. Once the partitioning is available, a simulation model has to be built for each subsystem and the exchange of waveforms has to be organized.

Specifying the computational and waveforms exchange structures

Simulating a continuous system with the Waveform Relaxation techniques in Scicos requires specifying the computational structure of each subsystems and how the waveforms are exchanged during the iterations of the chosen Waveform Relaxation method. The data flow formalism of Scicos is particularly well suited for Waveform Relaxation methods because with this formalism the user should specify the computational structure of the system being integrated. In our example each subsystem is built in a different Scicos diagram. The figures 1, 2 and 3 depict the computational structure of the three equations of our example (1).

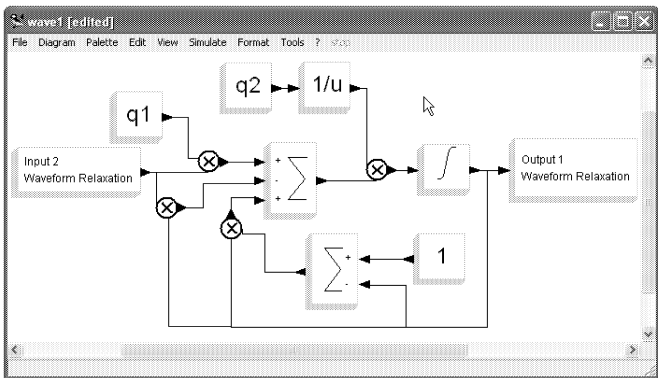


Figure 1 – Scicos diagram of the first equation of system (1) for a simulation with a Waveform Relaxation method.

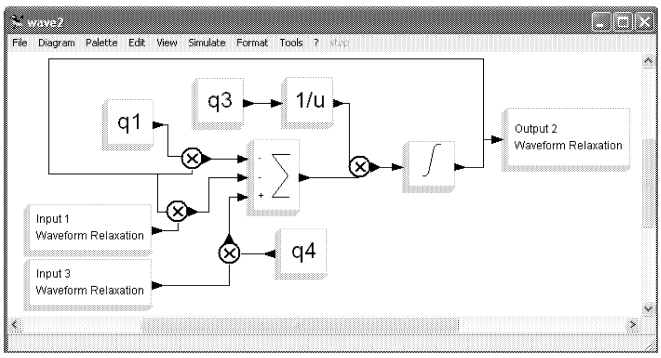


Figure 2 - Scicos diagram of the second equation of system (1) for a simulation with a Waveform Relaxation method.

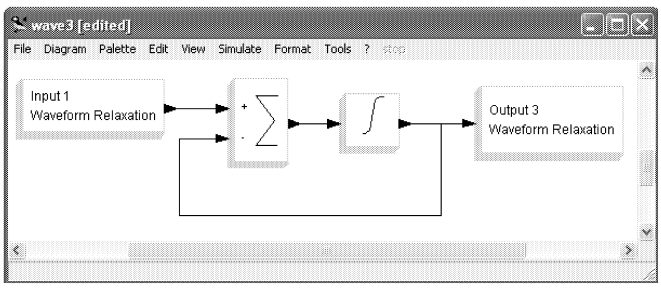


Figure 3 - Scicos diagram of the third equation of system (1) for a simulation with a Waveform Relaxation method.

New Scicos blocks for waveforms exchanges

In the figure 1,2 and 3, the blocks called "Input *x* Waveform Relaxation" and "Output *x* Waveform Relaxation" depict the sending and receipt of waveforms between the different subsystems**.** The number *x* between "Input" (resp. "Output") and "Waveform Relaxation" identifies which waveform is sent (resp. received). These two blocks have been specifically developed for the Waveform Relaxation methods. The block called "Input *x* Waveform Relaxation" returns in output the interpolated values of its stocked waveform at the exact simulation time asked by the Scicos simulator. The block called "Output *x* Waveform Relaxation" stores all the values of its input waveform. These values are stored in a global variable in order to be accessible in the ScicosLab workspace during the simulation.

Organising the simulation in the ScicosLab workspace

Once each subsystem has been modelled within a Scicos diagram, the chosen Waveform Relaxation method has to be organized in a ScicosLab file and launched in batch mode

from the ScicosLab workspace. We have defined a generic algorithm for the use of Waveform Relaxation with Scicos and ScicosLab. We use the Gauss Seidel form of the Waveform Relaxation method (Leralasmee et al. 1982). In order to be used this algorithm should be instantiated in a ScicosLab file with the correct ScicosLab syntax (Refer to the Scicos and ScicosLab help menus for more help):

- Load the precompiled Scicos diagram of each subsystem and save their main data structures;
- Define initial waveforms that are consistent with the initial conditions;
- (Windowing technique) Partition the main integration window into several smaller windows to speed the convergence of the method;
- Initialize the simulations of the Scicos diagrams for performing simulations in batch mode;
- **For** each window **do**:
    - Update the state variables of each Scicos diagram;
    - **While** all waveforms have not converged in the window **do**:
        - Update the global variables storing the waveforms;
        - Launch each diagram simulation;
    - **End While**
- **End For**
- End the simulations of the Scicos diagrams.

By instantiating this algorithm in a ScicosLab file, we obtain the convergence of the solution of the example (1). See the figure 4.

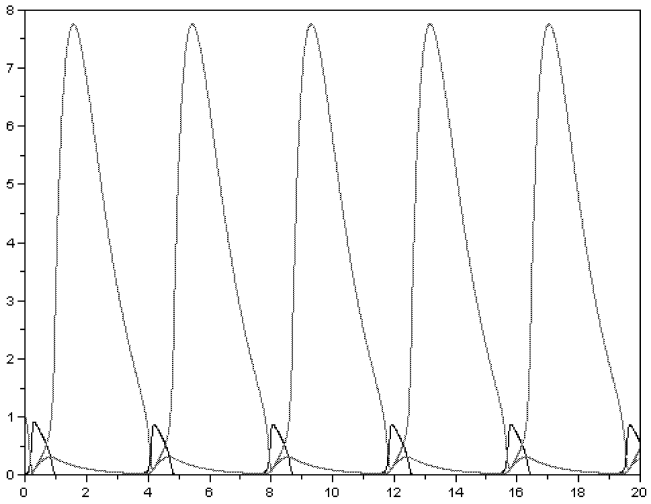


Figure 4 – Results of the simulation of the system (1) with the Gauss-Seidel form of the Waveform Relaxation method.

WAVEFORM RELAXATION FOR THE SIMULATION OF COMPLEX HYBRID SYSTEMS IN SCICOS

Systems arising in complex industrial applications are hybrid. They have variables evolving continuously and punctual events that create discontinuities. Scicos is particularly well suited for the modelling and simulation of hybrid dynamical systems. The classic Waveform Relaxation methods can be applied between two successive events of such hybrid systems because during this time window the integrated systems are continuous. However, in order to apply the Waveform Relaxation techniques on the entire time interval, where unpredicted events and their reconfigurations could appear, the exact location of the events has to be found. Then the Waveform Relaxation methods can be used between these events and the reconfigurations of these events correctly triggered to converge to the right solution. If the events are known before the simulation, this time event location problem is trivial. If the events are not known in advance (e.g. detected by a zero crossing mechanism), this problem is not trivial anymore. Indeed when integrating a subsystem with a Waveform Relaxation method, a waveform is not always the exact solution and detecting an event on this waveform and applying the reconfigurations of a detected event can lead to a divergence of the solution. In consequence we have defined a new procedure, called Event-related dynamic windowing procedure, and new Scicos blocks. This new procedure and these new blocks are used for the management of events detections and for applying their reconfigurations correctly when integrating a hybrid system with the Waveform Relaxation methods.

Event-related dynamic windowing procedure

In order to manage the events detections and their reconfigurations, we use the windowing techniques of White et al. (White, Ruehli, Sangiovanni-Vincentelli and Odeh 1985). This technique aims at partitioning the integration window $]t_0,T]$ into n smaller integration windows $]t_0,t_1]$, ... , $]t_{n-1},t_n]$. This technique improves the convergence speed of the Waveform Relaxation methods and avoids storing too much data. In practice, these integration windows can be chosen automatically during the simulation or manually before the simulation, in order to adapt the integration window to the subsystems stiffness. As we want to apply the Waveform Relaxation method to the simulation of hybrid systems, this technique can help us managing the events that occurs during the waveforms integrations and can help us applying their reconfigurations. If the main integration window is partitioned exactly at the event that triggers reconfigurations, the system to be integrated between these events are continuous and in the continuous case the literature (Leralasmee et al. 1982; White et al. 1985) ensure that the Waveform Relaxation methods will converge. However, when a hybrid system is integrated, the exact location of occurring events is not known in advance. In order to detect them and to partition the integration window according the detected events, we have defined the Event-related dynamic windowing procedure. This procedure is presented in the figure 5 and assumes that the user has already partitioned manually the integration window $]t_0,T]$ into n smaller integration windows $]t_0,t_1]$, ... , $]t_{n-1},t_n]$. It detects the occurrence of events and ensures that the solutions of all the waveforms have converged until the time location of an event before moving the integration windows to the next integration windows. Obviously the next integration window starts at the exact location in time of the detected event, just after the application of the reconfiguration triggered by the event detection.

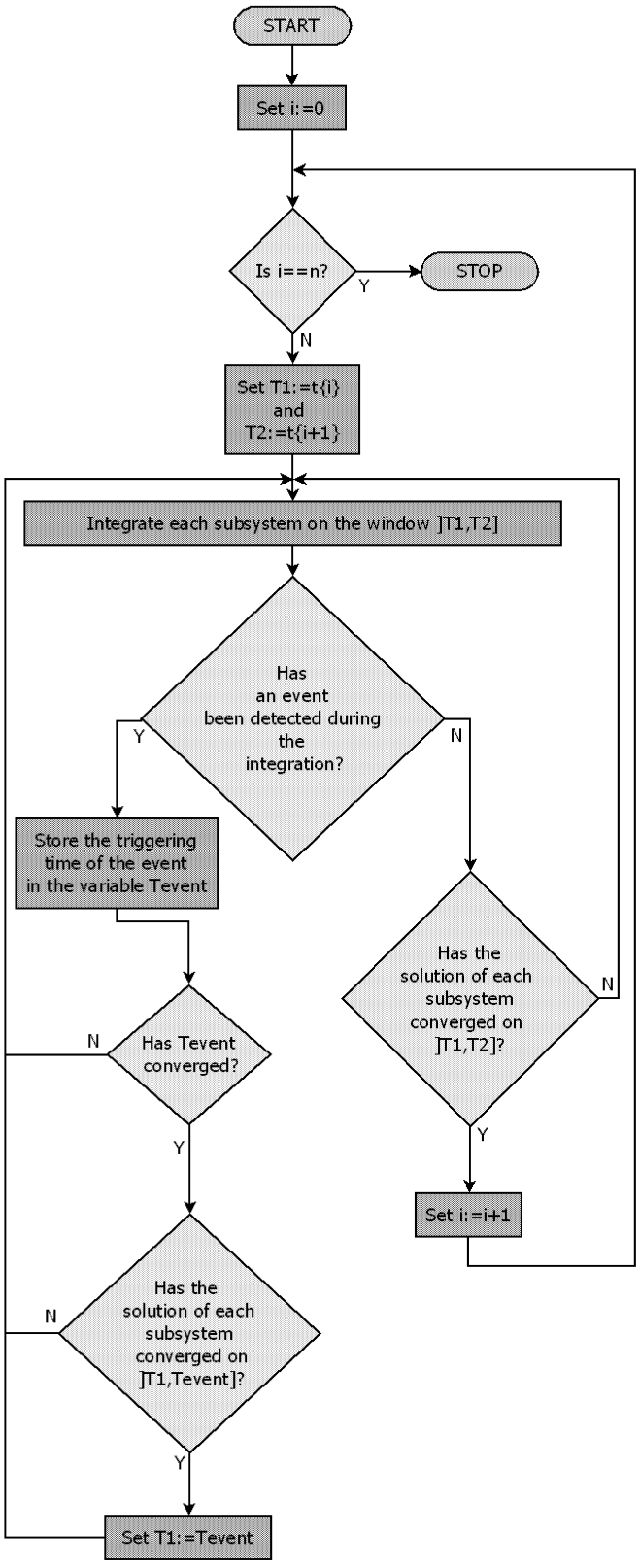


Figure 5 - Event-related dynamic partitioning procedure.

Events management for the application of the Waveform Relaxation methods to the simulation of hybrid systems

When using the Event-related dynamic partitioning procedure, it is necessary to store the triggering time of an event because we want to ensure that an event is exactly located in time and that the solutions of all the waveforms has converged until this event before applying its reconfigurations. Within Scicos, events are detected with the zero crossing mechanism, i.e. an output event is generated from a "Zero Crossing" block when the input signal of the block crosses zero. We have created two blocks for storing the triggering time of a zero crossing and then generating an output event when we are sure that the triggering time has converged:

- The block "ZC time", that stores the triggering time of an input event in a global variable that can be used in the ScicosLab workspace during the simulation;
- The block "Apply", that returns in output of the block a value stored in a global variable and that can be modified in the ScicosLab workspace between the iteration of a Waveform Relaxation method;

In order to use the Event-related dynamic partitioning procedure, these two blocks are used conjointly with the blocks "Zero Crossing", "If-Then-Else", "Stop" and "Integration". These blocks have to be placed in the following way:

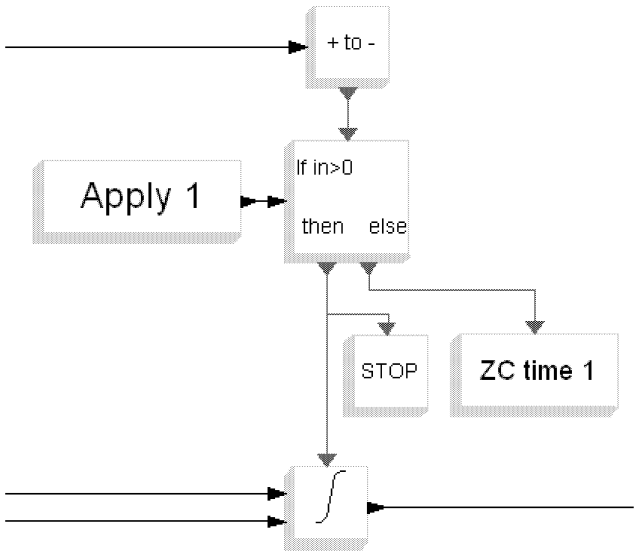


Figure 6 – Blocks positioning for event managing during the Event-related partitioning procedure.

When disposing the blocks as in the figure 6, the output signal of the Zero Crossing is redirected to the ZC time block if the Apply variable is negative or null. The Apply variable is a global variable that can be modified within the ScicosLab workspace between the iterations of the chosen Waveform Relaxation method. This variable is used to specify if the reconfigurations triggered by the Zero Crossing blocks should be applied or not. During the iterations of the chosen Waveform Relaxation methods the newer triggering time of the Zero Crossing block is compared to the previous values in order to state if it has converged. When it has converged the value of the Apply variable is modified and set positive in the ScicosLab workspace between two iterations. At the following iteration the output signal of the Zero Crossing block is redirected to the integrator block and the Stop block. The Stop block stops the simulation exactly at the time of the reconfiguration in order to move correctly the integration window.

Application on a simple example

This example is a bouncing ball linked by a spring to a platform. The platform is also linked by a spring to the ground. The ordinary differential equation system associated with the movement of the ball and the platform is the following:

$$(2)\begin{cases} m_1\ddot{x} = -g - \mu_1\dot{x} - k_1(x - \bar{x}) \\ m_2\ddot{y} = -g - \mu_2\dot{y} - k_2(y - x) \end{cases}$$

With $x(0) = 11, y(0) = 15, \dot{x}(0) = 0$ and $\dot{y}(0) = 1$.

In order to model the bounce of the ball on the platform, we use the following conditions:

"When $y - x$ is becoming inferior or equal than *0* then set

$$\dot{x} := 0.98\frac{m_1\dot{x} + 2m_2\dot{y} - m_2\dot{x}}{m_1 + m_2}$$ „

"When $y - x$ is becoming inferior or equal than *0* then set

$$\dot{y} := 0.98\frac{m_2\dot{y} + 2m_1\dot{x} - m_1\dot{y}}{m_1 + m_2}$$ „

Each equation is modelled in a different Scicos diagram with Input and Output Waveform Relaxation blocks laid as necessary to exchange the waveform between the iterations of the chosen Waveform Relaxation method. The bounce is modeled with the new created blocks positioned as in the figure 6, in order to reconfigure the speed of the ball and the platform when the ball hit the platform.

In order to simulate this example we use the Gauss Seidel form of the Waveform Relaxation method with the Event–related dynamic partitioning procedure of the figure 5 and the new developed Scicos blocks presented above. We obtain the convergence of the method to the exact solution. The simulation results are depicted with the curves of the figure 7.

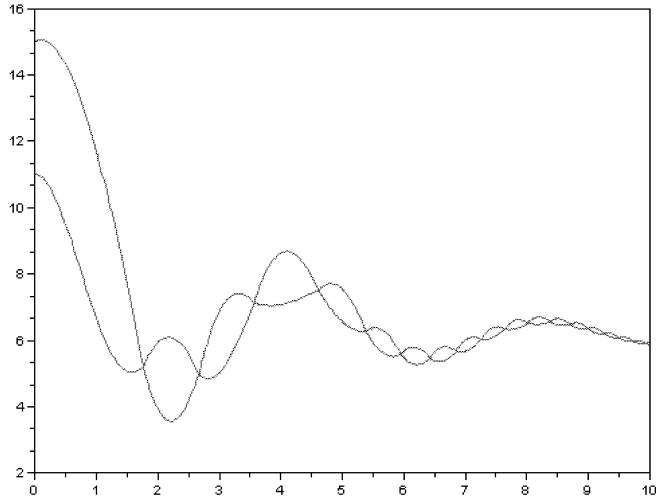


Figure 7 - Results of the simulation of the system (2) with the Gauss-Seidel form of the Waveform Relaxation method and the Event-related dynamic partitioning procedure.

CONLUSION

This paper proposes an application of the Waveform Relaxation methods to the simulation of hybrid systems by using the capabilities of Scicos. For this purpose new Scicos blocks have been created for exchanging the waveforms between the iterations of the chosen Waveform Relaxation method, verifying their convergences in the ScicosLab workspace and managing the reconfigurations triggered by the events occurring during the simulation. Indeed, the main challenge for the application of the Waveform Relaxation method to the simulation of hybrid systems is the management of these events during the iterations of the method. When integrating a system with one of the Waveform Relaxation methods, several iterations are needed to lead the different subsystems' waveform to their exact solutions. In consequence, the events detected on iterations and not on the exact solution are problematic because they can be imaginary, or they can be detected at the wrong time location. Therefore applying their reconfigurations can lead to a divergence of the approximation. To tackle this problem we developed an Event-related dynamic partitioning procedure. This procedure detects the occurrence of events and ensures that the solutions of all the waveforms have converged until the time location of an event before moving the integration windows to the next integration windows. Then the next integration window starts at the exact location in time of the detected event, just after the application of the reconfiguration triggered by the event detection.

REFERENCES

Burrage, K. 1995. *Parallel and Sequential Methods for Ordinary Differential Equations.* Oxford University Press, USA, 276-403.

Campbell, S.L.V.; J.P. Chancelier; R. Nikoukhah. 2006. *Modelling And Simulation in Scilab/Scicos.* Springer.

Lelarasmee, E.; A.E. Ruehli; A.L. Sangiovanni-Vincentelli. 1982. *The Waveform Relaxation Method for Time-Domain Analysis of Large Scale Integrated Circuits.*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1, 131-145.

White, J.; A.L. Sangiovanni-Vincentelli; F. Odeh; A. Ruehli. 1985. *Waveform Relaxation : Theory and Practice.* Transactions of The Society for Computer Simulation, 2, 95-133.

White, J. and A.L. Sangiovanni-Vincentelli. 1987. *Relaxation techniques for the simulation of VLSI circuits.* Kluwer Academic Publishers, Boston

GRAPHICAL OBJECT ORIENTED SIMULATION WITH SIMIO

Patrick Kirchhof, Nicolas G. Meseth
University of Osnabrueck
email: Patrick.Kirchhof@uos.de, Nicolas.Meseth@uos.de

KEYWORDS
Simio, Object-oriented, Manufacturing

ABSTRACT

While object orientation has long become a standard in software development, it has not been fully embraced in simulation modeling. There are many simulation frameworks which offer object oriented modeling based on a programming language but no sophisticated graphical tool. This paper describes Simio, a new simulation software, which simplifies model building by fully integrating an object-oriented modeling approach. Simulation models are built from objects, which are stored in object libraries and can be reused in multiple projects. Models themselves can be instantiated as objects in other models. A comprehensive set of standard objects is contained in Simio, while more refined ones can be developed completely graphically, and without the need for programming. Following the concept of object-orientation, objects are created through composition of existing objects, sub-classing existing objects, and overriding their behavior, or by building new objects from scratch. This paper is to provide a comprehensive overview of the Simio simulation software with a focus on its underlying object-oriented paradigm.

INTRODUCTION

Although the roots of object orientation can be traced back to simulation (Dahl and Nygaard 1966), the paradigm is not widely applied in today's simulation software (Robinson 2005; p. 624). In the field of simulation of manufacturing systems process-orientation is the predominant modeling paradigm of commercial simulation systems. Tools such as Rockwell Arena or Enterprise Dynamics are well known examples belonging to this category. Process-oriented modeling is considered intuitive and effective in practice, but there are also drawbacks. Modularity and reusability are limited and maintenance is difficult due to the dependencies within a model. Minor changes to the simulated system can prove to be very difficult to implement because changes have to be made at many different places in the simulation model (Edwin C. Valentin and Alexander Verbraeck 2005; p. 655). Others argue that the greatest limitation of the procedural modeling paradigm is the lack of extensibility (Joines and Roberts 1999; p. 133).

Object-oriented modeling constitutes a compelling alternative, because it is well suited for simulation and introduces many benefits. Although today many vendors of industrial simulation software claim their system provides object-oriented modeling capabilities, this is only partly correct. No product known to the authors fully supports an entirely graphical and object-oriented modeling approach without the need for programming. Most of today's tools contain a set of generic objects that can be used for model composition. The modeler can customize these base objects and enhance their functionality by developing code in a programming language. Thus, the use of these tools is very limited unless the user is capable of programming in the supported language. Well known examples are Extend, which uses a procedural language similar to C, or Anylogic, which integrates with Java (Edwin C. Valentin and Alexander Verbraeck 2007; p. 788). PlantSimulation (former emPlant) uses SimTalk, a procedural language designed for programming simulation applications (Bangsow 2008; p. 89). Another group of tools such as Simul8 can be used completely graphically without the necessity of programming, but is limited to a fixed set of modeling objects.

This paper's intention is to give an overview of the Simio[1] simulation software, a new system which fully supports object-oriented modeling in a purely graphical manner without the need for programming[2]. The most interesting features are pointed out, and it is shown how the modeler benefits from the underlying object-orientation. Therefore, the following section explains how object-orientation is applied in Simio. Afterwards, three different ways to extend Simio's functionality are described. The paper concludes with a summary of the most important aspects of the object-oriented modeling approach and its implications for the modeler.

[1] Simio ™ is a commercial product of Simio LLC.

[2] The term *programming* in this work is defined as typing code in a programming language such as C# or Java. Creating a program by clicking and arranging symbols is not considered programming in this context.

OBJECT ORIENTATION IN SIMIO

Simio Object Paradigm

In Simio, simulation models are built through the composition of objects and the definition of their behavior and interaction. Objects of any complexity can be stored and categorized in libraries. A set of objects with basic functionality is contained in the standard library included in Simio. It comes with a pre-defined set of basic objects, which includes a *source*, *sink*, *server*, *combiner*, *separator*, as well as different types of *links*. These basic building blocks suffice to build simple models, but in most real life cases they must be extended.
Custom objects are stored in user defined libraries, which can be shared and reused across multiple projects. According to the equivalence principle there is no difference between an object and a model. Every model is by definition also an object and can thus be instantiated into another model (C. Dennis Pegden 2008; p. 230). This is highly efficient because every model can be hierarchically used as a building block in a higher level of models.
It is important to note the difference between object libraries and libraries of process templates known from other commercial process-oriented tools such as Arena. Template libraries in process-oriented simulation tools contain templates and not objects. Templates are hierarchically constructed from basic building blocks, which are a graphical representation of a piece of functionality in a process-oriented simulation language. A typical example would be a template for a complex *workstation*, which is made of several building blocks such as *resource*, *queue*, *delay* etc. Therefore templates are not more than the logical organization of elementary constructs of the underlying simulation or general purpose programming language into a more abstract logical construct, generally combined with a visual parametrization interface.
In contrast, objects in Simio are true objects, which inherit attributes and behavior from their parent classes, allow their behavior to be modified (overridden) and extended, encapsulate their internal structure, and know abstraction and polymorphism. Objects own process definitions that specify their behavior and can be viewed as the equivalent of a method in object-oriented programming. Process definitions are built fully graphical from a comprehensive set of modeling elements called *steps*. The processes of any standard library object are also modeled using steps and can therefore be viewed and edited.
Objects and therefore models can have properties and states. A property can be thought of as a named parameter of an object instance and is usually constant during a simulation run. States are output responses that change throughout the execution of the object logic.
An object can define its own set of events and use those events to trigger processes. Objects are designed following the event-listener pattern and can evaluate events fired by other objects. This is useful to control the interaction between objects.
Building models in Simio is done using the graphical modeling interface, which divides an object (or model) into seven different views. While in the facility window the user composes the model from existing objects, he can specify the resulting object's appearance in the external window. In the process, data, and interface window, the user defines the object's behavior. Moreover, the dashboard and results window allow the user to watch the model's response during the simulation and define and analyze corresponding simulation experiments.
It should be noted that no programming is required and that the skills needed to build custom objects via inheritance are purely modeling skills and not programming skills (C. Dennis Pegden 2008; p. 231). Object libraries in Simio and template libraries share the same idea of user-defined libraries, but differ with regard to the modeling paradigm they are designed for. How objects can be defined and customized is explained in greater detail below.

Object Types

Any object in Simio is a subclass of either one of the types displayed in figure 1. These classes provide basic functionality, which serves as a starting point for the development of more refined objects. They only define generic behavior to provide a very general basis for modeling individual object definitions. Objects in the standard library are also descendants of one of these basic types.

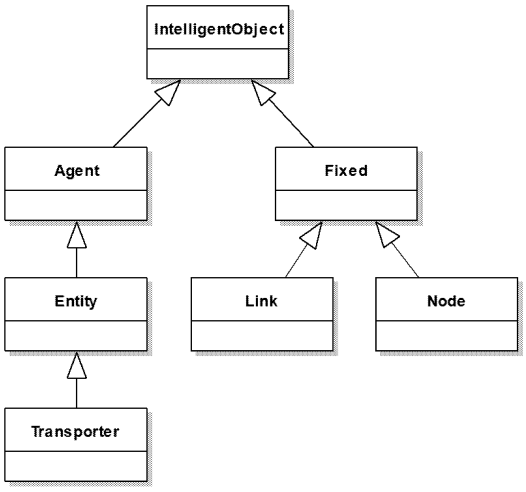


Figure 1: Class diagram showing the basic object types contained in Simio.

Every object is derived from the *Intelligent Object* class. *Intelligent Objects* have the optional ability to be seized,

released or follow a schedule. They are abstract and cannot be instantiated.
Fixed objects have a fixed location and are used to model stationary things in a system like machines in a manufacturing system, a plant in a supply chain model or a counter in an airport.
Objects of the *Agent* class are intended for agent-based modeling approaches in which a large number of independently acting agents form the system's behavior. *Agents* can move freely through continuous space without the need of a specified network of paths. They can detect other objects and react to their actions. In the current version of Simio, *Agents* are not yet fully functional and cannot be used.
Entity objects are derived from the *Agent* type and represent non-stationary objects that have the ability to move along specified paths and nodes. Paths and nodes can be grouped into networks in order to limit the scope of *Entities*. They can enter and exit other stationary object types (*Fixed, Node, Link*) and can be transported. The *Transporter* class is derived from the *Entity* class and can therefore also move along a specified network. *Transporters* add the capability to load, carry, and unload other objects. By default they follow the shortest path to their destination but other modes of behavior such as following a fixed route are also regarded.
Node objects are derived from the *Fixed* object class and are intended to represent vertices in a network, i.e. the starting or ending point of a *Link* object, or the entry or exit points of a *Fixed* object. At a *Node*, *Entity* objects may be loaded or unloaded by a *Transporter*. *Nodes* can be equipped with a crossing logic which can be executed when being visited.
Links represent the edges in a network. *Links* have a length and can be split into sections for their graphical representation. Each *Link* must have a starting and an ending point defined by *Node* objects.

Projects and Object Libraries

In Simio, simulation models are part of a *Project*, which is saved in a project file and contains all the elements of the model. A project comprises at least one model. By distributing a single project file, an entire simulation project can be shared between users.
Objects modeled in Simio can be reused in other projects by loading the respective object into a different project as a library. The models contained in the loaded project file appear as a user defined library below the standard library and can be used for modeling alike. Like models of the standard library they can also be extended and sub-classed. Models can be password protected so that they can be parameterized, used, and distributed, but their inner working is hidden and cannot be changed.
Both, the basic objects and the objects in the standard library described above are domain neutral and can be used in a wide variety of application areas (C. Dennis Pegden 2008; p. 233). Domain specific extensions to generic simulation packages provide a modeling aid in a certain domain and most vendors offer specialized modeling constructs (Edwin C. Valentin and Alexander Verbraeck 2005; p. 654). Simio does not include domain specific extensions, but it is easy to develop and share custom objects and libraries. Thus, domain specific libraries can be created from user defined objects and distributed as a project file.
The past section has given an overview of how the object-paradigm is applied in Simio. As stated earlier, in most scenarios the standard library's objects are not sufficient. Simio offers a number of ways in which it's functionality can be enhanced and they are subject of the next section.

EXTENSIBILITY

The extensibility of a simulation tool is a critical factor for its acceptance in practice. Simio offers three general ways to implement functionality that is not provided by the standard library. The following sections describe those three ways and give examples for each.

Creating Custom Objects

For simple models with little specific logic, the standard library described above is a good start. By nature, though, simulation models tend to require very specific and sometimes complex logic to model real world situations. Simio offers a simple way to extend the library without having to write a line of code.
Consider for example the *server* object from the standard library. It has a set of two nodes, one input and one output node, both having their own queue. Imagine this server should have a second output node, because it should serve two types of customers (entities) separately, and each type should leave the server through a separate exit. The steps required to implement this are straightforward and do not demand any programming skills.
The starting point for creating the new server model *MyServer* is the *server* object from the standard library. By dragging it to the project library, a new object class is created as subclass of the *server*. Subclassing has the benefit that all parts of the parent class are automatically inherited. Those include all states, properties, elements, and processes, and only the extended functionality must be added.
To add the second node and to make it visible to the user, it must be added to the object's external view. This can be done by selecting a node from the ribbon bar and placing it wherever it should appear. Preferably, it should be next to the existing output node. After adding a new node to the external view it must be connected to the object's internal logic.

so called elements. Elements are static parts that have a certain, in most cases general, functionality that they provide the object with. The station element, for example, represents a physical place where entities can reside. The server from the standard library has three station elements to represent the different places an entity can be within the server. The first station is used to model the input buffer where all arriving entities wait to be processed. The second contains an entity for the time it is processed. After the entity is finished processing, it enters the output station, which represents the server's output buffer. If the new output node should have its own output buffer, a new station in the process view must be added and connected to the output node.
Connecting a station to a node in this context means adding the right steps that send (transfer) an entity from the station to the node. The right place to add this step in case of the server object is the *OnEnteredProcessing* process. This process takes care of an entity that was sent to the processing station and makes it wait in a *delay* step until the processing time has elapsed. Afterwards, the process transfers the entity to the output station. Now that *MyServer* has two output stations, the process must be enhanced to decide to which station to send an entity based on some criteria (e.g. the value of an entity's state). This can be done by dragging in a *decide* step from the library and setting the condition accordingly. The second branch of the *decide* step must now be equipped with a *transfer* step to send the entity to the alternative station.
Finally, when the entity enters the new station, it should immediately be sent to the external output node through which it leaves the object (if the node has sufficient capacity left; otherwise the entity waits in the station until the node's capacity becomes available). Therefore, a process that is triggered whenever an entity enters the station must be created. Because the steps are the same as for the first output buffer, one can simply copy the *OnEnteredOutputBuffer* process and rename it.
To subsume, the necessary steps to extend the server to distinguish two types of entities are the following, of which cam all be done graphically:

1. Create a model *MyServer* subclassing the *server* model (drag & drop)
2. Add a node in the external window (drag & drop)
3. Add a station in the process window (drag & drop)
4. Add *decide* and *transfer* step to *OnEnteredProcessing* (drag & drop)
5. Create new process *OnEnteredOutputBuffer2* and add *transfer* step (drag & drop)

This simple example was to show how the user can extend Simio without any programming. Nonetheless, Simio offers a powerful application programming interface (API), which allows flexibility that goes beyond creating custom objects.

Customs Elements and Steps

The first way to programmatically extend Simio is to create custom elements and steps. Simio ships with a class library that contains interfaces which can be implemented to define a new element or step. Because Simio and its API are based on the .NET framework, steps and elements can be programmed in any .NET language. Therefore, Simio is not platform independent and can only be used under Windows systems.

Elements
To define a custom element one must implement the `IElement` interface. It specifies a single method `Initialize()`, which is called at the beginning of a simulation run. Here the programmer specifies what the element does, and has access to the simulation context via an object which is passed in the method as an argument. One possible application would be to schedule events in the event calendar before the simulation begins.
Elements can be viewed as complex objects that belong to a model. Therby elements can fulfill very different functionality. The station element, for example, is a place where entities can sit within a model, and therefore represents a physical location. In contrast, the file element enables an object to read data from a file, and thus represents a connector to the outside world of Simio.
Connectors to different data sources are good examples for custom elements. To enable the model to query data from a SQL Server database, the user could implement a new element and make the connection string configurable as a property of the element.

Steps
Elements are most useful when combined with according steps that in some way reference the element. To create a new step, the user must implement the `IStep` interface and put the desired functionality in its only method `Execute()`. The method is run when the step is executed as part of a process. To continue the example of the SQL server connector, a useful step would be one that executes a SQL query statement against the connection specified by the connector element.
The possibility to define new elements and steps is very powerful. It enables the modeler not only to extend Simio within the tool itself, but to implement logic that interacts with the outside world. A database connector is a good example. Other useful elements could be ones that query web-services on the internet or retrieve data from a structured file format such as XML. It is also possible to implement standardized connectors for widely used ERP and PPS systems, such as SAP or Mi-

crosoft Dynamics NAV.

Design Add-Ins

The second way Simio offers to extend the software programatically is by writing design add-ins. An add-in is a compiled .NET library (.dll). Add-in libraries are automatically displayed in the ribbon menu when they are put in the designated folder, and can be activated by clicking on the corresponding icon. For Simio to recognize a valid design add-in, the class must implement the provided interface `IDesignAddIn`. A design add-in has a name, description, and a custom icon property, all of which are set by the programmer. More important, though, is the interface's sole method `Execute()`; this is the method that is invoked when the user activates the add-in from the Simio software menu.
Within this method, the programmer has access to the currently open facility window and its model through the API provided by the Simio software. The programmer is offered methods to add objects to the model and place them by specifying their coordinates, delete objects from the model, and to set the model's properties. Thus, one has the same possibilities as a modeler who creates a model using the Simio GUI.
The option to create models programmatically makes it possible to dynamically generate models from external data sources. The data-driven simulation approach, where information about objects in the model is retrieved from different applications that already host this information, can thus be implemented with Simio. A good example is to automatically read the bill of materials from an ERP system instead of inserting it manually.
Another application for custom design add-ins is to load models stored in a different file format. For this purpose, the add-in fulfills the task of translating the foreign format into a Simio model. This way, models from other simulation tools such as Arena could be transformed into a Simio model automatically. For companies using other tools, who would like to migrate to Simio, but do not want to re-implement their existing models, this could be an option.
As described in this section, Simio offers three different levels to achieve extensibility. Thereby, creating custom library objects is even possible for non-programmers. This is an enhancement that could widen the range of users that can successfully apply simulation. In most cases, the driver of a simulation study is one who understands the business processes, but is not familiar with programming. Simio has the potential to close this gap.

CONCLUSION

This paper has introduced the new object-oriented simulation software Simio. Simio is the first truly object-oriented simulation software, which embraces all benefits of the object-oriented paradigm in an entirely graphical manner. Most importantly, Simio allows the modeler to easily create custom objects to meet highly specific requirements with no need for programming, taking advantage of inheritance. Nonetheless, the option to extend Simio programmatically is also available and twofold: custom steps and elements (the atomic units of functionality) as well as so called design add-ins (application plug-ins with full access to the software) release the modeler of any restrictions of what can be modeled. Within in the scope of this paper, the author's tried to give an overview of the *benefits* that derive from true object-orientation in a simulation tool; possible drawbacks such as a potential loss of performance, were not commented on. As Simio is a very novel tool, further investigations must be made to fully evaluate the tool with more aspects regarded.

REFERENCES

Bangsow S., 2008. *Fertigungssimulationen mit Plant Simulation und SimTalk: Anwendung und Programmierung mit Beispielen und Lösungen.* Hanser, München.

C. Dennis Pegden, 2008. *Introduction to Simio.* In S.J. Mason (Ed.), *2008 Winter Simuation Conference: Miami, Florida, USA, December 7 - 10, 2008.* IEEE, Piscataway, NJ, 229–235.

Dahl O.J. and Nygaard K., 1966. *SIMULA - an ALGOL-Based Simulation Language. Communications of the ACM*, 9, no. 9, 671–678.

Edwin C. Valentin and Alexander Verbraeck, 2005. *Requirements for domain specific discrete event simulation environments.* In M.E. Kuhl (Ed.), *Proceedings of the 2005 Winter Simulation Conference: Orlando, Florida, USA, December 4 - 7, 2005.* IEEE, Orlando, Fl., 654–663.

Edwin C. Valentin and Alexander Verbraeck, 2007. *Domain specific model constructs in commercial simulation environments.* In S.G. Henderson (Ed.), *Proceedings of the 2007 Winter Simulation Conference: Washington, DC, USA, December 9 - 12, 2007,.* IEEE, New York, NY, 785–792.

Joines J.A. and Roberts S.D., 1999. *Simulation in an object-oriented world.* In P.A. Farrington; H.B. Nembhard; D.T. Sturrock; and G.W. Evans (Eds.), *Proceedings of the 1999 Winter Simulation Conference.* Winter Simulation Conference.

Robinson S., 2005. *Discrete-event simulation: from the pioneers to the present, what next? Journal of the Operational Research Society*, 56, no. 6, 619–629.

FORTIS – AN INTEGRATED SIMULATOR FOR DISTRIBUTED TIME SERIES FORECASTING

Janetta Culita, Dan Stefanoiu
„Politehnica" University of Bucharest, ROMANIA
Dept. of Automatic Control and Computer Science (www.acs.pub.ro)
E-mails: jculita@yahoo.com, danny@indinf.pub.ro

KEYWORDS
Forecasting simulator, MIMO-ARMAX models, Kalman filtering, distributed signals, prediction quality.

ABSTRACT

The problem of natural phenomena forecasting is mainly approached within the framework of distributed MIMO systems. The process provides a collection of geographically distributed time series that are measured by means of a sensors network. Usually, some correlations between output data exists, which allows increasing the prediction accuracy. In this paper, the modeling and prediction of multi-variable signals rely on multi-dimensional ARMAX and state representation models. Three classes of predictors, namely *PARMA*, *PARMAX* and *KARMA*, were integrated in a simulator referred to as *FORTIS*, which is described in the article. A case study with real data is presented in the end. The application regards either one channel or multi-channel prediction algorithms and aims to comparatively assess the predictor performances.

1. INTRODUCTION

Monitoring of geographically distributed natural phenomena (like ecological or meteorological) seems to offer a higher accuracy prediction when using a network of wireless sensors instead of isolated ones [Stefanoiu-Culita, 2009]. Data acquired from different channels (sensors) provide a collection of *time series* (ts), which usually are correlated each other. For example, in Figure 1, the evolution of minimum and maximum temperatures over a geographical area are depicted. One notices the strong correlations between the four channels since the two cities are neighbors (only 50 km far from each other). The temperature variations are measured by means of a real-time wireless network of sensors. The acquired signals are transmitted via internet (satellite) to *VISA* – a versatile data acquisition interface [Stefanoiu-Petrescu, 2008]. After some preliminary processing, data are sent from VISA to a laptop, for more complex processing. The laptop is hosting *FORTIS* (forecasting of time series) – an integrated simulator aiming to perform prediction as well as monitoring of data.

Different approaches have been adopted in modeling and forecasting of geographical distributed phenomena. Up to now, many techniques referring unique (single) ts prediction have been proposed, such as neural network [Geva 1998], wavelet transform and wavelet network [Abiyev 2006]. However, the research in prediction of the distributed and possible correlated ts is rather limited. Usually, predictors are based on either analytical or numerical models. Although the analytical models are more accurate than the numerical ones, their complexity is quite high, which requires powerful computers. Numerical models offer an acceptable loss of accuracy and can be implemented on a regular computer. In our approach, three multi-variable numerical prediction models are considered: ARMA, ARMAX and based on state representation. The first one is more suitable (i.e. sufficiently accurate and less complex) when data channels are not correlated (or the cross-correlations are weak). In this case, each ts is independently predicted. The ARMAX model is more complex and generates better prediction results, as long as the data channels are cross-correlated. In context of MIMO systems, the distributed time series could also be interpreted as a collection of measurable states of an open system [Stefanoiu and Culita, 2009], [Stefanoiu et al. 2009]. The problem of state prediction is solved by applying Kalman filtering [Kalman and Bucy, 1961]. The correlation between data channels is decisive as well for higher prediction quality. The three multi-variable models led to the construction of *PARMA*, *PARMAX* and *KARMA* predictors.

In this article, the main objective is to introduce the FORTIS simulator, which integrates the three predictors above and interacts with the user by means of a newly designed friendly interface. The article is structured as follows: section 2 shortly describes the construction of prediction models; the FORTIS interface makes the subject of the section 3; in section 4 the case study based on the application of Figure 1 is approached; section 5 concludes the paper.

2. PREDICTION MODELS INSIDEFORTIS

Denote by $\mathcal{Y} = \{y_j\}_{j\in\overline{1,ny}}$ the collection of ny distributed ts measured by the set of network sensors. Each ts must contain the same number N_y of (equally) sampled data, otherwise interpolation and re-sampling are necessary. Consequently, one could construct a vector $\mathbf{y} = \left[\, y_1 \cdots y_{ny} \,\right]^T$, which can be understood as the output data of an open MIMO system. Firstly, consider every output signal as a colored noise, produced only by filtering some Gaussian and uncorrelated white noise e_j, where $j \in \overline{1,ny}$. Then, a generic ARMA model is built in order to identify the filter [Söderström and Stoica, 1989].

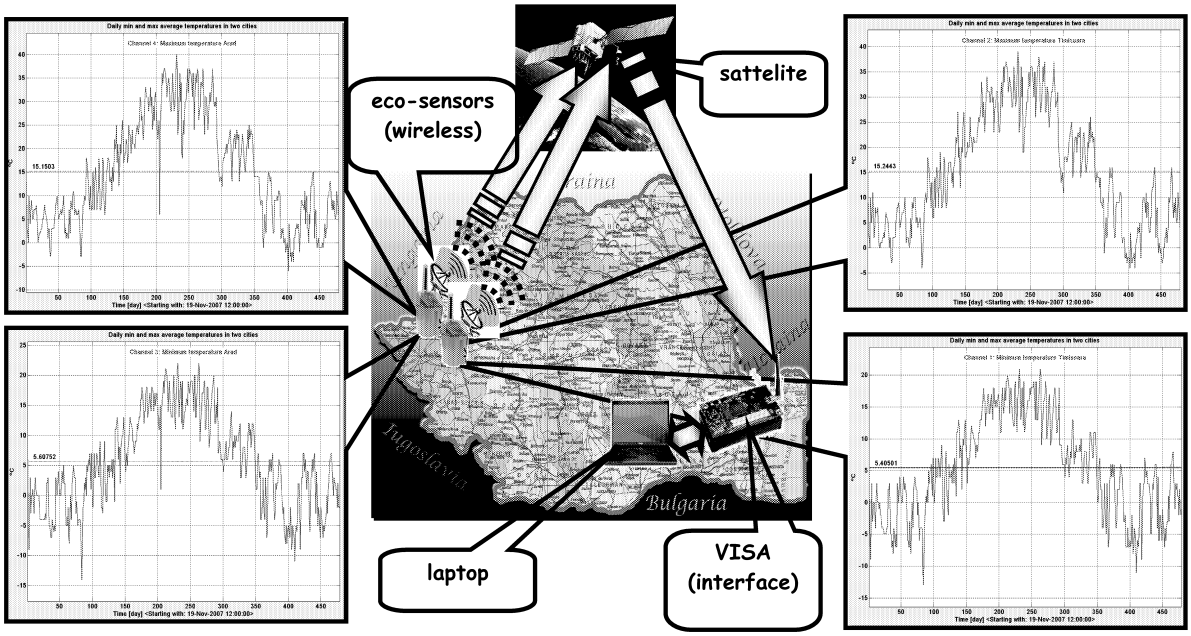


Figure 1. Temperature monitoring in two neighbor cities by means of FORTIS system.

The (SISO-)ARMA filter model is:

$$\mathrm{A}_j\left(\mathrm{q}^{-1}\right)y_j \equiv \mathrm{C}_j\left(\mathrm{q}^{-1}\right)e_j, \quad \forall j \in \overline{1,ny}, \tag{1}$$

where A_j and C_j are (singleton) polynomials for each j-th isolated channel. The polynomial parameters of models (1) as well as the values of white noise e_j are easily estimated via *Minimum Prediction Error Method* (MPEM) [Söderström and Stoica 1989]. A rough estimation of prediction error on each channel is:

$$\hat{e}_j \equiv \hat{\mathrm{A}}_j\left(\mathrm{q}^{-1}\right)y_j + \left[1-\hat{\mathrm{C}}_j\left(\mathrm{q}^{-1}\right)\right]\hat{e}_j, \quad \forall j \in \overline{1,ny}. \tag{2}$$

In case of distributed ts, if no correlations between the output data channels exist, the ts are independently processed. Dealing with each ts apart results in a rough MIMO-ARMA model, (associated to MIMO system), which was extended from model (1). Thus, the global model is:

$$\mathbf{A}\left(\mathrm{q}^{-1}\right)\mathbf{y} \equiv \mathbf{C}\left(\mathrm{q}^{-1}\right)\mathbf{e}, \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{ny\times ny}\left(\mathrm{q}^{-1}\right)$ and $\mathbf{C} \in \mathbb{R}^{ny\times ny}\left(\mathrm{q}^{-1}\right)$ are diagonal polynomial matrices and $\mathbf{e} \in \mathbb{R}^{ny}$ is the vector of Gaussian uncorrelated white noises ($\mathbf{e} = \left[e_1 \cdots e_{ny}\right]^T$). When correlations between various ts are accounted for, the model (3) is upgraded to the MIMO-ARMAX model below:

$$\mathbf{A}\left(\mathrm{q}^{-1}\right)\mathbf{y} \equiv \mathbf{B}\left(\mathrm{q}^{-1}\right)\mathbf{u} + \mathbf{C}\left(\mathrm{q}^{-1}\right)\mathbf{e}. \tag{4}$$

In equation (4), $\mathbf{B} \in \mathbb{R}^{ny\times nu}\left(\mathrm{q}^{-1}\right)$ is a polynomial matrix and $\mathbf{u} \in \mathbb{R}^{nu}$ is the input vector (on nu channels) of the MIMO system. The estimation of the input signal relies on the estimation of white noises from equation (2), i.e. $\hat{u}_j \equiv \hat{e}_j$. This assumption leads to partially reconsidering the hidden correlation between output data, expressed by the colored noises corrupting them. The colored noises are produced by filtering the input signal (determined) as white noises.

Usually, the upgraded model (4) offers improved prediction performances than the initial one. The identification of model (4) becomes very difficult when the polynomials in non diagonal positions of matrices $\mathbf{A}$ and $\mathbf{C}$ are non null. Therefore, in order to ease the identification task, one supposes that matrices $\mathbf{A}$ and $\mathbf{C}$ are diagonal. Such hypotheses are reasonable because the output signals are practically independent (although cross-correlated through inputs). Accordingly, the MIMO-ARMAX model can be divided into ny MISO-ARMAX models:

$$\mathrm{A}_j\left(\mathrm{q}^{-1}\right)y_j \equiv \mathbf{B}_j\left(\mathrm{q}^{-1}\right)\mathbf{u} + \mathrm{C}_j\left(\mathrm{q}^{-1}\right)e_j, \quad \forall j \in \overline{1,ny}, \tag{5}$$

where $\mathbf{B}_j \in \mathbb{R}^{1\times nu}\left(\mathrm{q}^{-1}\right)$ is a row vector of polynomials. The problem is now to identify each MISO-ARMAX model, separately. In view of prediction, all equations (5) have to be solved at each step. Note that the number of input channels is $nu = ny - 1$, since the white noise affecting a channel has already been included in the MA part of model. Therefore, the matrix $\mathbf{B}$ in (5) has null diagonal. The identification of parameters in model (5) is performed by applying MPEM. Furthermore, the noises estimations are retrieved via the ARX approximating model:

$$\hat{e}_j \equiv \hat{\mathrm{A}}_j^{n\alpha}\left(\mathrm{q}^{-1}\right)y_j - \sum_{i=1,i\neq j}^{ny}\hat{\mathrm{B}}_{j,i}^{n\beta}\left(\mathrm{q}^{-1}\right)\hat{u}_i, \quad \forall j \in \overline{1,ny}, \tag{6}$$

where $\mathrm{A}_j^{n\alpha}$ and $\mathrm{B}_{j,i}^{n\beta}$ are polynomials with degrees $n\alpha$ and $n\beta$, respectively. (The structural indexes of ARX model are much grater than in equation (5).) The optimum structural indices are selected through an adaptation of *Particle Swarm Optimization* (PSO) technique [Stefanoiu et al. 2009], [Kennedy and Eberhart 1995].

After estimation of all parameters, the optimal predictor is recursively computed, on a prediction horizon of length $K \geq 1$ (for any $k \in \overline{N_y+1, N_y+K}$):

$$\hat{y}_j\left[k\,|\,N_y\right]=-\hat{a}_{j,1}\hat{y}_j\left[k-1\,|\,N_y\right]-\cdots-\hat{a}_{j,na_j}\hat{y}_j\left[k-na_j\,|\,N_y\right]+$$
$$+\sum_{i=1,i\neq j}^{ny}\left[\hat{b}_{j,i,1}\hat{u}_i[k-1]+\cdots+\hat{b}_{j,i,nb_{j,i}}\hat{u}_i\left[k-nb_{j,i}\right]\right]+$$
$$+\hat{c}_{j,1}\hat{e}_j[k-1]+\cdots+\hat{c}_{j,nc_j}\hat{e}_j\left[k-nc_j\right]; \qquad (7)$$

$$\hat{e}_j[k]=\hat{y}_j\left[k\,|\,N_y\right]+\hat{\alpha}_{j,1}\hat{y}_j\left[k-1\,|\,N_y\right]+\cdots+\hat{\alpha}_{j,k-N_y-1}\hat{y}_j\left[N_y+1\,|\,N_y\right]+$$
$$+\hat{\alpha}_{j,k-N_y}y_j\left[N_y\right]+\cdots+\hat{\alpha}_{j,n\alpha}y_j[k-n\alpha]-$$
$$-\sum_{i=1,i\neq j}^{ny}\left[\hat{\beta}_{j,i,1}\hat{u}_i[k-1]+\cdots+\hat{\beta}_{j,i,n\beta}\hat{u}_i[k-n\beta]\right]. \qquad (8)$$

Equations (7) and (8) take into account the correlations between different output data. One notices that all output channels provide the estimation of the next predicted value.

The prediction performance is expressed in terms of *prediction quality* (PQ) cost function, defined as follows, for any channel $j\in\overline{1,ny}$:

$$\mathrm{PQ}_j\overset{def}{=}100/\left(1+\frac{\sqrt{\sum_{k=1}^{K}\hat{\sigma}_{j,k}^2}}{\hat{\lambda}_{e_j}\sqrt{\mathrm{SNR}_j}\sqrt{\mathrm{SNR}_j^K}}\right)[\%],\ \forall\, j\in\overline{1,ny}. \qquad (9)$$

where: $\left\{\hat{\sigma}_{j,k}^2\right\}_{j\in\overline{1,ny}}$ are the prediction error variances; $\mathrm{SNR}_j\overset{def}{=}\sigma_{y_j}^2/\hat{\sigma}_{j,1}^2$ and $\mathrm{SNR}_j^K\overset{def}{=}\left(\sigma_{y_j}^K\right)^2/\left(\sigma_{y_j-\hat{y}_j}^K\right)^2$ are signal-to-noise ratios with σ_{y_j}, $\sigma_{y_j}^K$ as standard deviations of data on measuring and prediction horizons, respectively; $\sigma_{y_j-\hat{y}_j}^K$ is the standard deviation of prediction error. The bigger the norm of $\mathbf{PQ}=\left[\mathrm{PQ}_1\ \cdots\ \mathrm{PQ}_{ny}\right]^T$, the better the prediction performance.

The third approach deals with the identification and prediction of state representation models like below:

$$\begin{cases}\mathbf{x}[k+1]=\mathbf{A}_k\mathbf{x}[k]+\mathbf{B}_k^{\mathbf{x}}\mathbf{u}[k]+\mathbf{F}_k\mathbf{w}[k]\\ \mathbf{y}[k]\ \ =\mathbf{C}_k\mathbf{x}[k]+\mathbf{B}_k^{\mathbf{y}}\mathbf{u}[k]+\mathbf{D}_k\mathbf{v}[k]\end{cases},\quad \forall\, k\in\mathbb{N}, \qquad (10)$$

where: $\mathbf{A}_k\in\mathbb{R}^{nx\times nx}$, $\mathbf{B}_k^{\mathbf{x}}\in\mathbb{R}^{nx\times nu}$, $\mathbf{B}_k^{\mathbf{y}}\in\mathbb{R}^{ny\times nu}$, $\mathbf{C}_k\in\mathbb{R}^{ny\times nx}$, $\mathbf{D}_k\in\mathbb{R}^{ny\times ny}$ and $\mathbf{F}_k\in\mathbb{R}^{nx\times nx}$ are matrices including all variable (but already estimated) parameters of some stochastic process; $\mathbf{x}\in\mathbb{R}^{nx}$ is the unknown state vector; $\mathbf{u}\in\mathbb{R}^{nu}$ is the vector of (measurable) input signals; $\mathbf{y}\in\mathbb{R}^{ny}$ is the vector of (measurable) output signals; $\mathbf{w}\in\mathbb{R}^{nx}$ is the (unknown) endogenous non auto-correlated system noise with covariance matrix $\mathbf{\Psi}_{\mathbf{w}}[k]=E\left\{\mathbf{w}[k]\mathbf{w}^T[k]\right\}$; $\mathbf{v}\in\mathbb{R}^{ny}$ is the (unknown) non auto-correlated exogenous noise, that affects the output data, whose covariance matrix is $\mathbf{\Psi}_{\mathbf{v}}[k]=E\left\{\mathbf{v}[k]\mathbf{v}^T[k]\right\}$. Usually, $\mathbf{w}$ and $\mathbf{v}$ are uncorrelated to each other. The discrete stochastic states and outputs of model (10) are predicted by means of an adapted version of basic Kalman-Bucy algorithm [Kalman and Bucy 1961].

- *Input data: a small collection of time series values (the training set $\mathcal{D}_0=\left\{\mathbf{y}[n]\right\}_{n\in\overline{1,N_0}}$) yielding initialization.*

1. *Initialization. Produce the first state representation (10). Then complete the initialization by setting: an arbitrary state vector $\hat{\mathbf{x}}_0$, the covariance matrices $\hat{\mathbf{P}}_0=\alpha\mathbf{I}_{nx}$ (with $\alpha\in\mathbb{R}_+^*$), $\mathbf{F}_{-1}\mathbf{\Psi}_{\hat{\mathbf{w}}}[-1]\mathbf{F}_{-1}^T=\mathbf{0}_{nx}$ and $\mathbf{D}_{-1}\mathbf{\Psi}_{\hat{\mathbf{v}}}[-1]\mathbf{D}_{-1}^T=\mathbf{0}_{ny}$.*
2. *For $k\geq 0$:*
   - 2.1. *Estimate the exogenous mixed noise:* $\mathbf{D}_k\hat{\mathbf{v}}[k]=\mathbf{y}[k]-\mathbf{C}_k\hat{\mathbf{x}}[k]-\mathbf{B}_k^{\mathbf{y}}\mathbf{u}[k]$.
   - 2.2. *Update the covariance matrix of exogenous noise:* $\mathbf{D}_k\mathbf{\Psi}_{\hat{\mathbf{v}}}[k]\mathbf{D}_k^T=\frac{1}{k+1}\left(k\mathbf{D}_{k-1}\mathbf{\Psi}_{\hat{\mathbf{v}}}[k-1]\mathbf{D}_{k-1}^T+\mathbf{D}_k\hat{\mathbf{v}}[k]\hat{\mathbf{v}}^T[k]\mathbf{D}_k^T\right)$
   - 2.3. *Compute the auxiliary matrix:* $\mathbf{Q}_k=\mathbf{C}_k\hat{\mathbf{P}}_k$.
   - 2.4. *Invert the matrix:* $\mathbf{R}_k=\mathbf{D}_k\mathbf{\Psi}_{\hat{\mathbf{v}}}[k]\mathbf{D}_k^T+\mathbf{Q}_k\mathbf{C}_k^T\in\mathbb{R}^{ny\times ny}$.
   - 2.5. *Evaluate the sensitivity gain:* $\mathbf{\Gamma}_k=\mathbf{Q}_k^T\mathbf{R}_k^{-1}$.
   - 2.6. *Compute the auxiliary matrix* $\mathbf{S}_k=\mathbf{A}_k\mathbf{\Gamma}_k$.
   - 2.7. *Update the covariance matrix of endogenous noise:* $\mathbf{F}_k\mathbf{\Psi}_{\hat{\mathbf{w}}}[k]\mathbf{F}_k^T=\frac{1}{k+1}\left(k\mathbf{F}_{k-1}\mathbf{\Psi}_{\hat{\mathbf{w}}}[k-1]\mathbf{F}_{k-1}^T+\mathbf{S}_k\mathbf{D}_k\hat{\mathbf{v}}[k]\hat{\mathbf{v}}^T[k]\mathbf{D}_k^T\mathbf{S}_k^T\right)$.
   - 2.8. *Update the covariance matrix of estimation error:* $\hat{\mathbf{P}}_{k+1}=\mathbf{F}_k\mathbf{\Psi}_{\hat{\mathbf{w}}}[k]\mathbf{F}_k^T+\mathbf{A}_k\left(\hat{\mathbf{P}}_k-\mathbf{\Gamma}_k\mathbf{Q}_k\right)\mathbf{A}_k^T$.
   - 2.9. *Predict the state:* $\hat{\mathbf{x}}[k+1]=\mathbf{A}_k\hat{\mathbf{x}}[k]+\mathbf{B}_k^{\mathbf{x}}\mathbf{u}[k]+\mathbf{S}_k\mathbf{D}_k\hat{\mathbf{v}}[k]$.
   - 2.10. *Predict the output:* $\hat{\mathbf{y}}[k+1]=\mathbf{C}_k\hat{\mathbf{x}}[k+1]+\mathbf{B}_k^{\mathbf{y}}\mathbf{u}[k+1]$.
   - 2.11. *Acquire new data:* $\mathcal{D}_{k+1}=\mathcal{D}_k\cup\left\{\mathbf{y}[k+1]\right\}$.
   - 2.12. *Update the state model.*

- *Output data:*
  - *predicted time series values $\left\{\hat{\mathbf{y}}[k]\right\}_{k\in\mathbb{N}^*}$;*
  - *estimated covariance matrices $\left\{\mathbf{D}_k\mathbf{\Psi}_{\hat{\mathbf{v}}}[k]\mathbf{D}_k^T\right\}_{k\in\mathbb{N}^*}$.*

To update the state model from acquired data (step 2.12), a subspace identification algorithm can be implemented [Overschee and Moor 1996]. The predictor performance uses the cost function PQ, where $\left\{\hat{\sigma}_{j,k}^2\right\}_{j\in\overline{1,ny}}$ is the diagonal of each matrix $\mathbf{D}_{N_y+k}\mathbf{\Psi}_{\hat{\mathbf{v}}}[N_y+k]\mathbf{D}_{N_y+k}^T$. The optimum structural indices (i.e. the polynomial degrees and the states number) are selected through the PSO technique, as well.
Three predictors have been implemented previously within MATLAB environment: PARMA (eq. 3), PARMAX (eq. 5-8) and KARMA (the algorithm above), according to the prediction models.

3. FORTIS INTERFACE

A graphical user interface has been designed within MATLAB environment for FORTIS, in order to allow

communicating with the user. Figure 2 displays the main screen. As one can notice, the aim was to integrate, in a unified approach, under a joined interface, the predictors constructed (implemented) before and even to leave room for another ones (such as FORWAVER [Stefanoiu et al. 2008]). Basically, the simulator interface yields running any of the individual predictors, according to some (parameter) settings. During the run, the prediction performances are graphically displayed. The main window (support) of the simulator consists of four panels. The two panels on top are designed to select the configuration parameters of the single channel predictor (PARMA) or multi-channels predictors (PARMAX, KARMA), respectively The south-west panel provides the user with helping messages and intermediate results. Finally, the south-east panel reveals the performances of running predictor. When starting the simulator framework, the message panel asks for the name of prediction algorithm and its working parameters.

In Figure 3, the panel for single channel prediction is emphasized (the north-west panel in Figure 2). For example, the PARMA predictor was selected. (FORWAVER has to be integrated in this category as well.) Then, the number of data blocks containing the measured ts (e.g. 479) must be introduced before any other parameter. The next fields let the user to select the configuration parameters from default values. The stability test box must be checked only if the predictor has to run with stable models (which is however significantly slowing down the optimization procedure). As the prediction algorithm is executing, the messages panel (Figure 4) shows the partial results of PQ, as computed by PSO searching procedure.

Figures 5 and 6 deal with the multi-channel forecasting (the north-east panel in Figure 2). Here lie PARMAX and KARMA. This panel is almost similar to the left side one, except that few parameters must be specified in addition. In case of PARMAX (Figure 5), two components are inserted: the maximum order of exogenous polynomials and the channel index on which the PQ cost function is maximized. When KARMA is selected, the maximum number of states must be set as well (like in Figure 6).

The graphics panel of Figure 7 (the south-east panel in Figure 2) is designed to visualize the final forecasting results for each output channel of MIMO model. After completion of prediction algorithm, a rich list of figures is provided. They depict 4 groups of variations for each channel, which are bond to the predictor performance: the original ts together with its optimal trend, the estimated white noise on measuring horizon; the predicted values and the PQ (the last one being displayed in Figure 7). Each variation could be separately displayed by selecting the corresponding figure and saved for later retrieval. Moreover, the performances of other predictors previously employed, could be reloaded by simply browsing the variations saved before. Subsequently, the performances of the different predictors could be compared to each other.

4. SIMULATION RESULTS

The case study described next relies on the application illustrated in Figure 1. Daily minimum and maximum temperature in two neighbor cities have been monitored and predicted by means of the real-time wireless monitoring system in Figure 1 and FORTIS simulator.



Figure 2. The main screen of FORTIS interface.

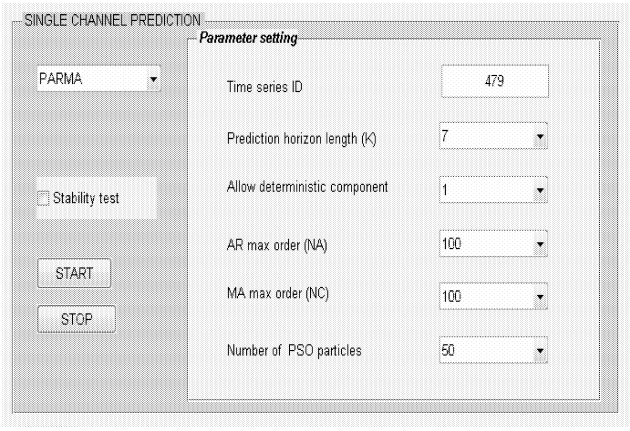


Figure 3. Single channel prediction panel of FORTIS.

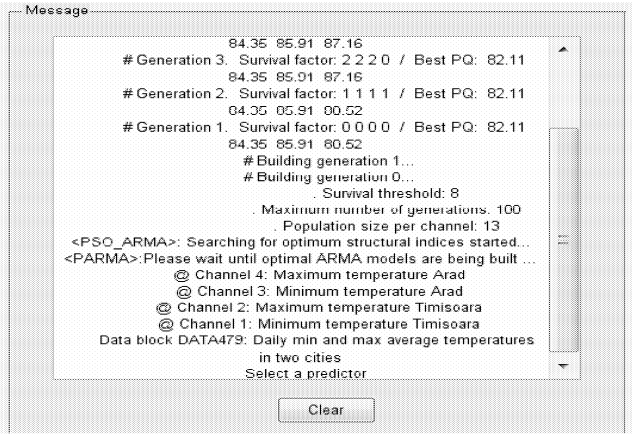


Figure 4. Messages panel of FORTIS.

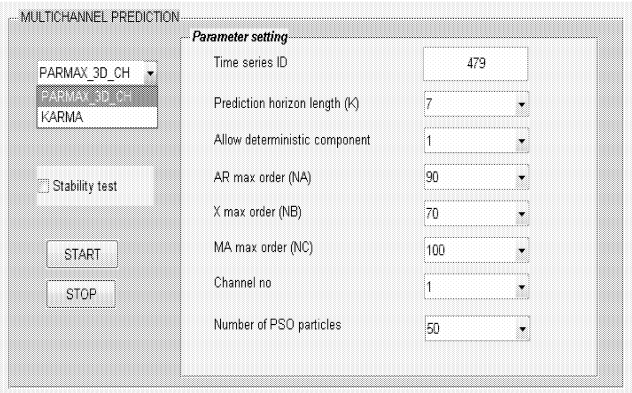


Figure 5. Multi-channel prediction with PARMAX.

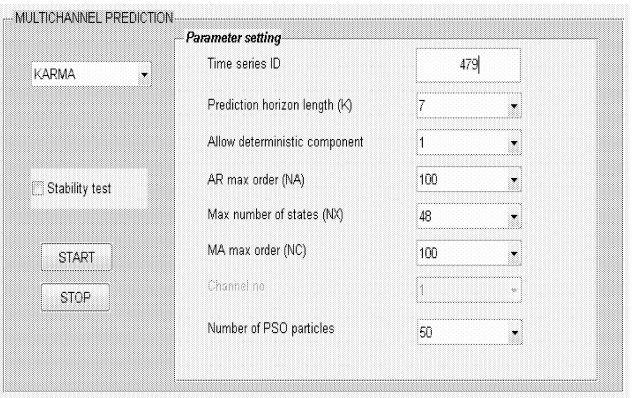


Figure 6. Multi-channel prediction with KARMA.

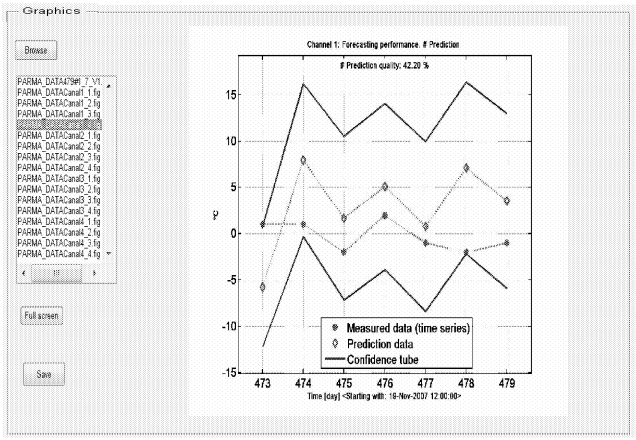


Figure 7. Graphics panel of FORTIS.

The main goal is to accomplish a comparison between PARMA, PARMAX and KARMA algorithms, in terms of PQ criterion.

The monitoring system is structured on three hierarchical levels: the lowest level consists of a set of eco-sensors that transmit the measured data to the middle level, by means of some eco-nodes (radio nodes), via satellite; the middle level is represented by a versatile acquisition interface VISA that deals with some preliminary operations: acquiring, de-noising (filtering) and organizing data in specific data bases, depending on their nature; then, the data are sent to the superior level – the FORTIS simulator, hosted by a laptop. The laptop is also connected by means of a gateway module to the other components of the acquisition system, in order to set some network parameters (nodes structure, sampling rates, data transmission instants, monitoring parameters etc.). The mobile computer is mostly responsible for efficiently processing the data and implementing the forecasting interface with subsequent prediction algorithms. Besides, prediction can be executed almost in real-time, for sufficiently large amounts of data (up to few hundreds). Usually, meteorological phenomena act slowly. So, it is not necessary for the sampling rate to be high, which gives enough time to predictors to complete. For about 500 data on 4 channels (like in Figure 1), FORTIS needs 40 minutes to 3 hours to provide a forecasting (the laptop being at the current level of performance, i.e. dual core type, 2 GB cash memory and 2.8-3 MHz operating frequency). The PSO procedure is the most time consuming, as the number of models is quite high (several tens of millions). Of course, the PSO algorithm is not testing all of them, but only a very small number (few thousands). However, identification of each model is the slowest operation integrated in PSO algorithm, as part of PQ computing. As expected, PARMA is the fastest predictor. The other two predictors (PARMAX and KARMA) are 3-5 times slower.

The acquired temperatures on the 4 channels have 479 samples each (in Celsius degrees). The sampling period for monitoring was set to 1 minute and max-min

values are recorded daily. So, the sampling period for prediction is one day. The predictors try to provide forecasting of max-min values within for the next 7 days. In order to compute the PQ, the last 7 acquired data of each channel are preserved, without being involved in models identification, like never existed. Therefore, from the 479 acquired data per channel, only 472 are effectively employed to set the predictors. For PARMA predictor, the parameters were set as follows (Figure 3): the number of prediction steps is 7; the deterministic component was set to 1 (alternatively 0, when no trend is computed for the ts); the maximum values of AR and MA polynomial orders are the maxim value 100, the number of PSO particles is 50 (from the default maximum value of 100). For PARMAX and KARMA the running parameters were set as in figures 5 and 6, respectively.

Figures 8-11 show the performances of the three predictors above in terms of PQ and confidence tube. Each predicted value has a trusting probability that defines the tube. As the prediction instant goes away from the measuring horizon, the tube becomes larger and larger. This means the predicted values are less and less reliable.

One observes that PQ of PARMA predictor has the worst values comparing to the other two predictors, as expected, because the correlations between channels are omitted. However, PARMA works faster and should be preferred in case of weak correlations between data. On the contrary, one can see that PARMAX and KARMA predictors offer higher accuracy of prediction due to hidden cross-correlation between channels. The two cities are only 50 km far from each other and, moreover, they are both located in a region with smooth small hills (almost a plane). Therefore, it is natural that the temperatures (among other meteorological phenomena) be so correlated. Thus, any of the last two predictors should be used when the channels are (strongly) correlated. The figures also reveal that PARMAX is seemingly a better predictor than KARMA, on the most channels.

5. CONCLUSION

This article introduced FORTIS – a simulator that integrates three predictors for distributed time series: PARMA, PARMAX and KARMA, under the same user friendly interface. In view of predicting the geographical distributed phenomena, the designed interface was implemented on a mobile computer that works together with some wireless monitoring system. The goal of designing the interface is to facilitate the comparative evaluation of the predictors performance. One concludes that PARMAX and KARMA predictors are more accurate than PARMA, as long as data are cross-correlated. Conversely, if their prediction accuracy is higher than PARMA can provide, then one expects that the output channels are rather cross-correlated. As future work, more predictors have to be integrated into FORTIS, especially the ones that take into account the time varying spectral characteristics and/or fractal behavior of acquired data (like FORWAVER).

REFERENCES

Abiyev H. Rahib, 2006. “Time Series Prediction Using Fuzzy Wavelet Neural Network Model”. *In Artificial Neural Networks- ICANN 2006*, Springer Berlin-Heidelberg, pp. 191-200.

Geva Amir. B., 1998. “Scalenet – Multiscale Neural Network Architecture for Time Series Prediction”. *IEEE Transaction on Neural Networks,* vol. 9, pp. 1471-1482.

Kalman R.E. and Bucy R.S., 1961. „New Results in Linear Filtering and Prediction Theory”, *Transactions of ASME, Journal of Basic Engineering*, Series D, Vol. 83, pp. 95-108.

Kennedy J., Eberhart R., 1995. „Particle Swarm Optimization”, *IEEE International Conference on Neural Networks*, Piscataway – N.J., U.S.A., pp. 1942-1948.

van Overschee P., de Moor B., 1996. *Subspace Identification of Linear Systems: Theory, Implementation, Applications*, Kluwer Academic Publishers, Amsterdam, Holland.

Söderström T., Stoica P., 1989. *System Identification*, Prentice Hall, London, U.K.

Stefanoiu D., Culita J., Ionescu F., 2008. „FORWAVER – A Wavelet Based Predictor for Non Stationary Signals”, *Proc. of EUROSIS-ISC-2008*, Lyon, France, pp. 377-381.

Stefanoiu D., Petrescu C., 2008. „VISA – A Versatile Interface of Signal Acquisition and Wavelet Based Data Fusion”, *The 5-th Europe-Asia Mechatronics Congress*, Annecy, France, Paper no. 167.

Stefanoiu D., Culita J., 2009. „PARMAX – A Predictor for Distribted Time Series”, *Proc. of EUROSIS-ISC-2009*, Loughborough, UK, pp. 5-12.

Stefanoiu D., Seraficeanu C., Culita J., Musca G., 2009. “Identification of MIMO-ARMAX Models for Glycemia and Sodium Ions Tests through Particle Swarm Optimization” – *The 7th IEEE International Conference on Control and Automation,* Christchurch, New Zealand, December 9-11.

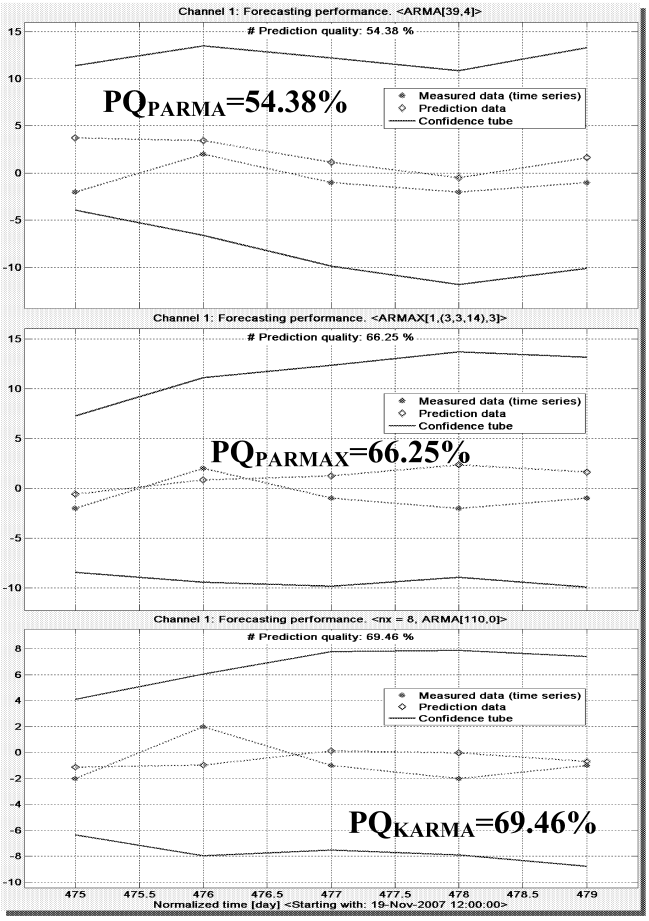


Figure 8. Predictors performance on channel 1.

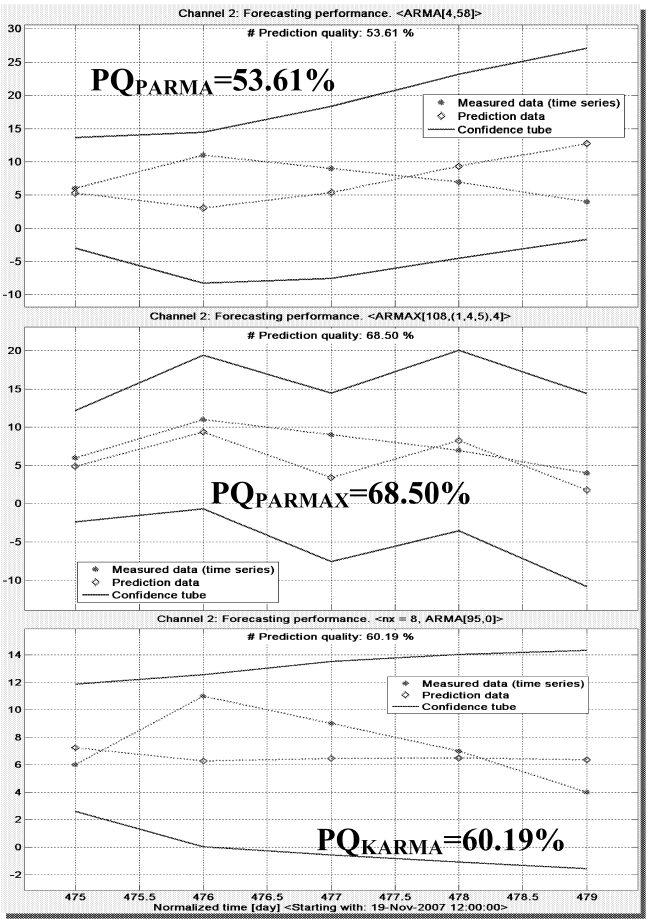


Figure 9. Predictors performance on channel 2.

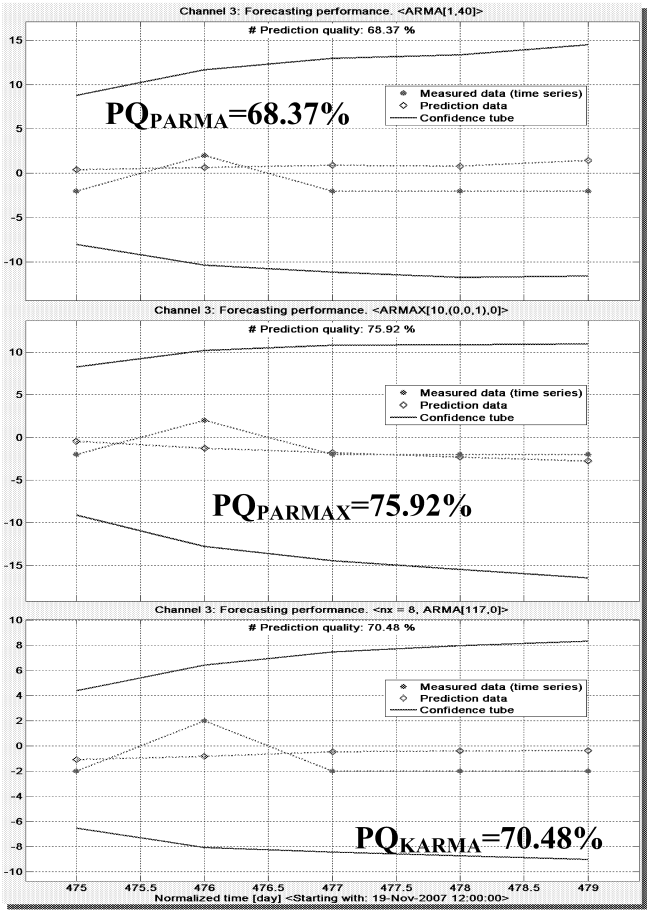


Figure 10. Predictors performance on channel 3.

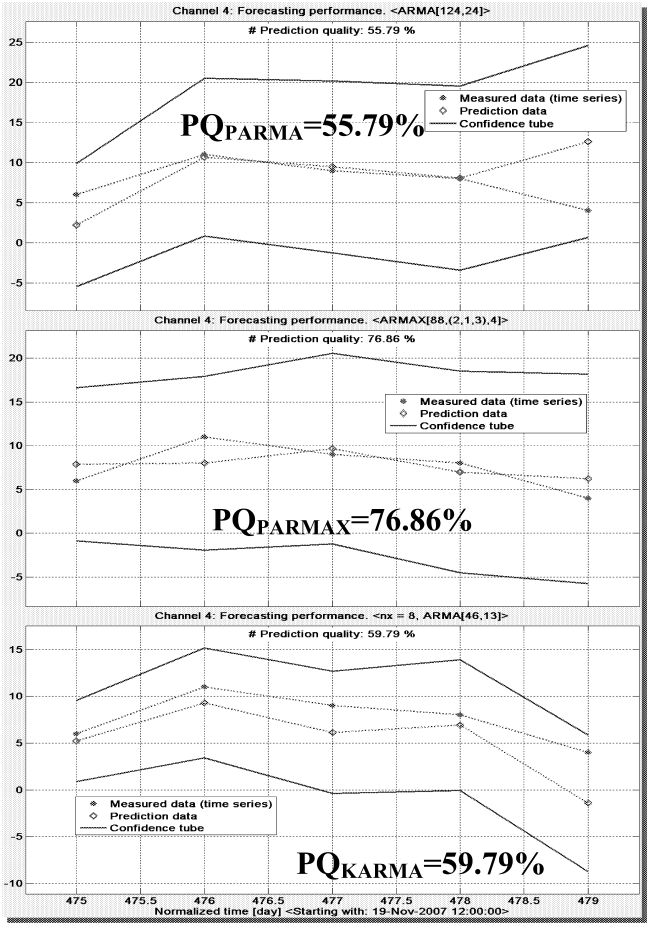


Figure 11. Predictors performance on channel 4.

ViennaIPD - An Input Control Language for Scientific Computing

Josef Weinbub
Karl Rupp
Siegfried Selberherr

Institute for Microelectronics, Technische Universität Wien
Gußhausstraße 27-29 / E360
A-1040 Vienna, Austria
E-mail: {weinbub|rupp|selberherr}@iue.tuwien.ac.at

ABSTRACT

A powerful control language, named ViennaIPD, has been developed. The library based software is capable of reading complex input datasets which can be accessed from C/C++ applications by a programming interface. ViennaIPD supports a convenient C-like input file language, object oriented structuring of datasets, powerful inheritance mechanisms and a unit system. The input file language as well as the programming interface is discussed in detail.

INTRODUCTION

Applications for scientific computing require a powerful control language to satisfy the need of control parameters, e.g. material properties, models to take into account, model parameters, process definitions, simulation modes, iteration schemes, and numerical behavior [1, 2, 3, 4]. As a result, control files grow in size, which results in decreased maintainability. Therefore a powerful control language has been developed as part of the MINIMOS-NT simulation software package [5, 6]. To enlarge the field of application the control language has been extracted, revisited, and packaged to be distributable as a standalone control language named ViennaIPD. ViennaIPD offers a powerful inheritance concept, unit conversion capabilities and the ability to directly access existing input files. The latter is especially of interest for reasons of post-processing simulation results. Figure 1 depicts a comprehensible overview how ViennaIPD is used.

ViennaIPD provides a powerful input file language which aims to deal with the manifold parameter demands of simulation software. The library itself is written in C, however, applications can be based on either C or C++. The most fundamental element is a *variable*. To describe dependencies to other variables, a variable contains an expression, that is evaluated at runtime. Different variable types are supported as well as accessibility control. Furthermore, a *section* provides a means of structuring variables to enable associative variable definitions. A section holds an arbitrary set of variables and subsections.

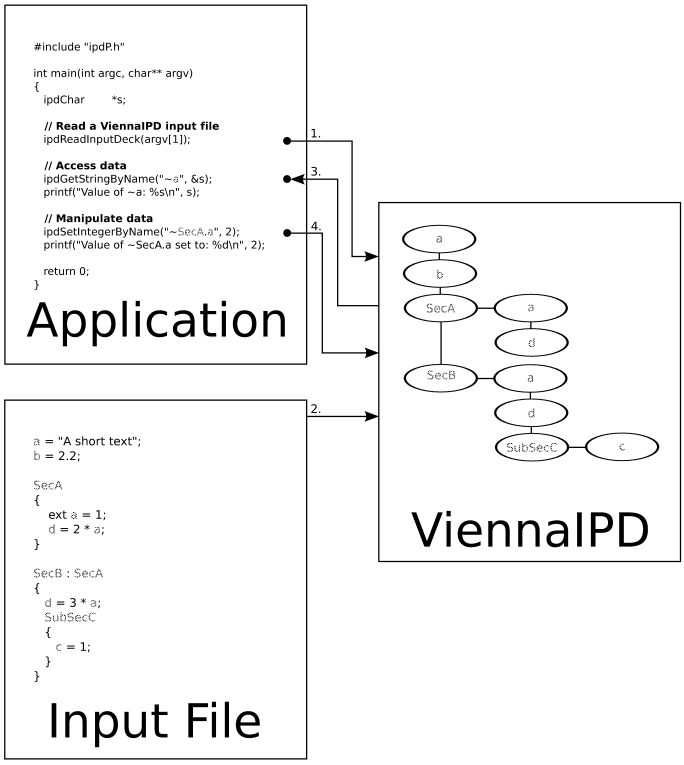


Figure 1: ViennaIPD workflow. The application passes the filename of the input file to ViennaIPD (**1.**). The input file is used to populate the tree based datastructure (**2.**). The data can be accessed (**3.**) and manipulated (**4.**) by the application.

Therefore any kind of data set hierarchy can be built. In this context, inheritance of sections is supported to enable, for example, reusability of section elements. Finally a large set of functions is provided to enable mathematical computations within the input files.
The paper is organized as follows: The first section provides an overview of the ViennaIPD control language, whereas the second section outlines the programming interface. Furthermore, the third section depicts a use case example. Finally, the fourth section compares ViennaIPD to various other input control approaches.

INPUT FILE SPECIFICATION

The input file specification is based on a comprehensible language. The usage is similar to common programming languages, e.g. C. Additionally, object oriented features are provided to enable structuring of large data sets, i.e. inheritance mechanisms. Furthermore, a powerful unit system is provided which supports the validity of computations based on quantities.

Variables

An arbitrary number of variables can be defined. The general definition of a variable is

```
<type> <variable_name> = <expression>;
```

<type> must be a valid variable type. The available variable types are listed in Table 1. Note that if no <type> is mentioned, the default key type is used. <variable name> can be of any length, can contain capital and small letters, an underscore, and numbers.

<*type*>	Description
ext	external variable - read- and writeable
key	keyword variable - read-only (default)
aux	auxiliary variable - hidden

Table 1: Types of Variables

However, the name must not begin with a number. An expression can be assigned to a variable by using the assignment operator = and a valid data-type (Table 2).

Type	Example
Boolean	`a = true;`
Integer	`a = 3;`
Real	`a = 3.1415;`
Complex	`a = 4.3 + 3.1 j;`
Quantity	`a = 3.1415 "m/s";`
String	`a = "This is a string";`
Array	`a = [1, "pi", 3 A];`

Table 2: Types of Data

The following snippet depicts a few examples which make use of the available variable types.

```
1       some_float = 3.3;
2   key some_text  = "some Text";
3   ext x          = 32;
4   ext y          = 4;
5   aux a          = sqrt(x * x + y * y);
6       ex         = x / a;
7       ey         = y / a;
```

Lines 1,2,6,7 depict keyword variables. Hence, the assigned data can be accessed only, not altered. Lines 3,4 depict external variables, for which data can be accessed and altered. Line 5 outlines an application case for the auxiliary variable. This variable cannot be accessed by the programming interface, it is hidden.

However, this variable type can be used to temporarily store data, for example the result of a computation.

Sections

ViennaIPD supports structures of variables, so called sections, to manage large input files. A section is defined with

```
<section name> { ... }
```

The following snippet depicts a section named `Solve` with a read-only keyword variable.

```
Solve { maxStepSize = 1e30; }
```

Sections can be nested to arbitrary depth

```
BaseSection {
   SubSection {
       SubSubSection { }
   }
}
```

ViennaIPD enables also inheritance of sections.

```
BaseSection { x1 = -3; }
DerivedSection : BaseSection;
```

`DerivedSection` is inherited from `BaseSection`. Consequently, `DerivedSection` contains all elements (variables and subsections) of `BaseSection`.
Note, that the variables of `BaseSection` *transparently* exist in `DerivedSection`. Hence, if a variable in `BaseSection` changes, it changes in `DerivedSection` too.
Furthermore, multiple inheritance is supported.

```
BaseSection1 { a = 1; }
BaseSection2 { b = 3; }
DerivedSection : BaseSection1, BaseSection2;
```

`DerivedSection` consists of two variables `a` and `b` inherited from `BaseSection1` and `BaseSection2`, respectively.
Additionally, conditional inheritance enables to inherit from sections based on specific conditions.

```
Section : BaseSection1 ? expression1,
          BaseSection1 ? expression2,
          ... { }
```

Conditional inheritance is an extension to multiple inheritance. Several base sections can be specified for inheritance, an expression can be assigned to each one. The following snippet depicts the usage.

```
Switch = 0;
DerivedSection : BaseSection1,
                 BaseSection2 ? (Switch > -1),
                 BaseSection3 ? (Switch <  1)
{ }
```

`DerivedSection` is always inherited from `BaseSection1`. As `Switch` is 0, the conditions for `BaseSection2` and `BaseSection3` both hold.

Therefore, `DerivedSection` is additionally inherited from `BaseSection2` and `BaseSection3`. If `Switch` is set to 2, `DerivedSection` is solely inherited from `BaseSection1` and `BaseSection2`.
Elements of inherited sections can be locally modified.

```
BaseSection {
    x = 1;
    y = 2;
}
DerivedSection : BaseSection { y = 4; }
```

`DerivedSection` consists of the two variables `x` and `y`, where the variable `x` is inherited from `BaseSection` and the variable `y` is locally modified. So the variable `DerivedSection.x` equals 1 and the variable `DerivedSection.y` equals 4.

Functions

ViennaIPD enables the use of functions within input files. Functions can be used in expressions which are stored in variables.

```
<type> <variable_name> = <function>(parameters);
```

An arbitrary number of parameters is supported.

```
a1 = func1();              // 0 Parameter
a2 = func2(1 + 2);         // 1 Parameter
a3 = func3("hello", a1); // 2 Parameter
```

ViennaIPD provides built-in functions, for example a large set of mathematical functions, e.g. `sin()`, `pow()`, `sqrt()`, to list a few of them. The following snippet depicts the computation of the sine of a complex number.

```
a = sin(4.42 + j*5.9);
```

Furthermore support functions are provided, e.g. `if()` statement, conversion functions, array sizes, random number generator, to name a few.

Units

ViennaIPD provides a powerful unit system. A unit is used in combination with the *quantity* datatype (Table 2) and consists of a valid unit name, of which many are provided. Quantity examples are:

```
q1 = 3.0 A;
q2 = 5.5 "V * A";
q3 = 4.2 "kg m/s s";
```

Note that if a quantity is based on several unit names, double quotes have to be used (Lines 2,3). Unit names can be separated by a whitespace or an asterisk `*` (Line 2). The use of a single slash `/` indicates the beginning of the denominator. Hence, Line 3 represents the unit $kg\ m/s^2$. Quantity based computations are checked for semantically correct operations. Therefore, an error occurs, if two quantities with different units are summed up, as this would result in a mixed up physical quantity and, therefore, looses its validity. The following snippet outlines this behavior.

```
Q1 = 3 "A";
Q2 = 2 "m";
Q3 = Q1 * Q2;   // OK
Q4 = Q1 + Q2;   // Error
```

PROGRAMMING INTERFACE

The ViennaIPD Programming Interface is based on the C programming language. It provides an interface to the internal tree-based datastructures of ViennaIPD. Data can be traversed and accessed.

Interface Setup

To be able to use the interface the ViennaIPD header file has to be included.

```
#include "ipd.h"
```

The following code snippet depicts the initialization and the population of the datastructure from an input file.

```
// basic initialization
ipdInit(NULL, NULL);
// create a new database
ipdCreateBase("NameOfTheDatabase", 0);
// read the inputfile
ipdReadInputDeck(argv[1]);
```

Accessing Data

Specialized getter and setter functions are provided to access and alter specific data types. The following code snippet depicts the reading and writing process based on these functions.

```
// declare a variable of type long
ipdLong i;
// get the "c" integer value
ipdGetIntegerByName("~c", &i);
// set the "c" integer value
ipdSetIntegerByName("~c", 2);
```

Note the ~ prefix, which indicates the root level of the datastructure hierarchy. Therefore variables can be accessed by absolute path addressing. For example accessing an integer variable named `a` within a section named `BaseSection` would be implemented by the following:

```
// declare a variable of type long
ipdLong i;
// get the "BaseSection" member value "a"
ipdGetIntegerByName("~BaseSection.a", &i);
```

Data can also be accessed without absolute addressing. This approach can be utilized by setting the current section. In relation to the previous example, the current section can be set to `BaseSection` as follows:

```
ipdSetCSByName("~BaseSection");
```

Consequently, the data can be accessed directly

```
// get the integer data associated with the "a"
// variable in the current section
ipdGetIntegerByName("a", &i);
```

Iterators

ViennaIPD provides iterators for convenient traversal of datastructures. The following snippet creates a new iterator on the root level

```
// define a new iterator
ipdIterator_t  *iNode = NULL;
// set the iterator to traverse the root level
ipdIteratorNewAtRootSection(&iNode, ipdANY);
```

All elements, e.g. variables, sections, on the root level can be traversed.

```
// traverse as long as there are elements on
// the root level
while (ipdIteratorIsValid(iNode)) {
   // retrieve the name of the current element
   ipdConstString itemName =
      ipdIteratorGetItemName(iNode);
   // is the element a variable?
   if (ipdIteratorGetType(iNode) == ipdVARIABLE)
      printf("variable: %s\n", itemName);
   // is the element a section?
   else
   if (ipdIteratorGetType(iNode) == ipdSECTION)
      printf("section: %s\n", itemName);
   // increment the iterator to point
   // to the next element
   ipdIteratorDoNext(iNode);
}
```

USE CASE EXAMPLE

The following depicts an exemplary ViennaIPD input file which contains a small dataset.

```
1  Electrons {
2     uL300      = 0.143    "m^2/V*s";
3     uLImin300  = 0.008    "m^2/V*s";
4     Cref300    = 1.12e17  "cm^-3";
5     CrefexpT   = 3.2;
6  }
7  Holes : Electrons {
8     uL300      = 460      "cm^2/V*s";
9     uLImin300  = 45       "cm^2/V*s";
10    Cref300    = 2.23e17  "cm^-3";
11 }
```

Note the different units for the same parameters in Lines 2,3,8,9. The unit system automatically converts the units into SI units, hence, the quantities presented in Lines 8,9 are internally converted to: $0.046 m^2/s \cdot V$ and $0.0045 m^2/s \cdot V$, respectively. Further note, that Electrons and Holes have different quantities for the $Cref300$ parameter, which too get converted (Lines 4,10). Due to inheritance, both sections use the same value for the $CrefexpT$ parameter (Line 5). The following depicts the access of some data of the Holes section.

```
1  double ul, uLImin;
2  char*  unit;
3  ipdGetRealQuantityByName("~Holes.uL300",
4     &ul, &unit);
5  ipdGetRealQuantityByName("~Holes.uLImin300",
6     &uLImin, &unit);
7  std::cout << ul     << " " << unit << std::endl;
8  std::cout << uLImin << " " << unit << std::endl;
```

COMPARISON

A comparison of different input control software packages is presented next. The comparison is based on the following features.

- *convenient scripting language:* The input language should be intuitive to scientists with slight background in programming. This results in an easy familiarization which, furthermore, reduces the overall implementation effort.
- *unit system:* Typically scientific computing relies on physical quantities. Hence, it is of major interest to provide a computation facility which respects units and therefore enables a correct setup of physical quantities.
- *memory consumption:* The memory consumption, for example the peak heap memory usage, should be kept small.
- *execution performance:* The input control software should not decrease the overall execution performance of the scientific application.

ViennaIPD is compared to the common scripting languages, Lua [7], AngelScript [8], and Python [9] in the following. All of the introduced software packages are supported with a convenient and intuitive language.

Lua is a so called extension programming language which extends the functionality of mature programming languages, like C. Lua is a minimalistic, C based library which solely exists embedded in host applications. Lua can be used by different programming languages, e.g. C, C++, Fortran.
AngelScript is a flexible cross-platform scripting library. C and C++ functions can be called within an AngelScript environment, hence code can be reused efficiently.
Python is a mature programming language, but is also used as a scripting language. Python offers a C-API, and a C++ library for Python interoperability is also available [10].
All these input control languages offer interoperability with C and C++. Note, that most of the available programming languages, for example Fortran, are able to interoperate with C.
In the following, benchmark results are presented and discussed. The test platform is a PC with an AMD Phenom II X4 - 965 CPU and 8GB of RAM. The operating system is a Gentoo Linux 64-bit with a 2.6.32 kernel.

Table 3 depicts a comparison of the peak memory consumption and the execution time of a reference application which is driven by the introduced control languages. To depict the overhead of the input control languages, the standalone application, without any input control, is investigated too. The memory usage evaluation is based on Valgrind [11].

Software	*Peak Heap Memory* [KB]	*Execution Time* [ms]
No Input Control	0.8	6
Lua	32	18
AngelScript	34	162
Python	2132	627
ViennaIPD	543	32

Table 3: Comparison of peak memory consumption and execution time.

The following table compares the features of the introduced input control languages.

Software	*Units*	*Mem*	*Exec*
Lua	-	++	++
AngelScript	-	++	+
Python	++	o	o
ViennaIPD	++	+	++

Table 4: Feature Comparison of input control languages. Feature status: ++ excellent, + good, o fulfilled, - missing

Consequently, ViennaIPD is especially attractive for applications which need a robust, fast, and light-weight unit system at the input file level.

CONCLUSION

A powerful control language for scientific computing has been introduced. The key features are

- C-like language for input files
- arbitrarily nested expressions
- powerful inheritance mechanisms
- unit system
- C/C++ programming interface

The application usage has been investigated in depth. Furthermore, the input file specification and the programming interface have been discussed in detail. Input file snippets as well as code snippets for the programming interface have been presented to outline the usage. ViennaIPD is part of the Viennese TCAD simulation tools which are provided by the Institute for Microelectronics [12].

REFERENCES

[1] D. Meeker, "Finite Element Method Magnetics - User's Manual," 2007. [Online]. Available: http://www.femm.info/wiki/HomePage

[2] A. Olabi and A. Grunwald, "Computation of magnetic field in an actuator," *Simulation Modelling Practice and Theory*, vol. 16, no. 10, pp. 1728 – 1736, 2008.

[3] J. Iniguez and V. Raposo, "Numerical simulation of a simple low-speed model for an electrodynamic levitation system based on a halbach magnet array," *Journal of Magnetism and Magnetic Materials*, vol. 322, no. 9-12, pp. 1673 – 1676, 2010.

[4] M. Meinel, "FlowSimulator: A Python-controlled Approach to Unify Future CFD Simulation Workflows," in *EuroSciPy 2009*, 2009.

[5] S. Wagner *et al.*, "MINIMOS-NT User's Guide," Institute for Microelectronics, Technische Universität Wien.

[6] R. Klima, "Three-Dimensional Device Simulation with Minimos-NT," Dissertation, Technische Universität Wien, November 2002.

[7] R. Ierusalimschy, L. H. de Figueiredo, and W. Celes, "Lua 5.1 Reference Manual." [Online]. Available: http://www.lua.org

[8] "AngelScript Manual." [Online]. Available: http://www.angelcode.com/angelscript/

[9] "Python Documentation." [Online]. Available: http://www.python.org

[10] "Boost Python C++ Library." [Online]. Available: http://www.boost.org

[11] "Valgrind." [Online]. Available: http://valgrind.org/

[12] "Institute for Microelectronics, Technische Universität Wien." [Online]. Available: http://www.iue.tuwien.ac.at

BIOGRAPHY

JOSEF WEINBUB received the BSc degree in electrical engineering and the degree of "Diplomingenieur" in microelectronics from the "Technische Universität Wien" in 2009. He is currently working on his doctoral degree, where his scientific interests include modern programming techniques for scientific computing, high-performance computing, computational topology and adaptive mesh generation.

KARL RUPP received the BEng degree in electrical engineering from the "Technische Universität Wien" in 2006, the MSc in computational mathematics from Brunel University in 2007, and the degree of "Diplomingenieur" in microelectronics and in technical mathematics from the "Technische Universität Wien" in 2009. He is currently working on his doctoral degree, where his scientific interests include generative programming of discretization schemes such as the finite element method for the use in multiphysics problems.

SIEGFRIED SELBERHERR was born in Austria in 1955. He received the degree of "Diplomingenieur" in electrical engineering and the doctoral degree in technical sciences from the "Technische Universität Wien" in 1978 and 1981, respectively. Dr. Selberherr has been holding the "venia docendi" on computer-aided design since 1984. Since 1988 he has been the chair professor of the Institute for Microelectronics. From 1998 to 2005 he served as dean of the Faculty for Electrical Engineering and Information Technology. His current research interests are modeling and simulation of problems for microelectronics engineering.

SIMULATION OF REAL-TIME ROBOT CONTROL SYSTEMS USING PLAYER/STAGE SOFTWARE

Vitalijs Komasilovs
Egils Stalidzans
Department of Computer Systems
Faculty of Information Technologies
Latvia University of Agriculture
Liela iela 2, Jelgava
Latvia
E-mail: vitalijs.komasilovs@gmail.com

Keywords
Simulation interfaces, Real-time system

Abstract

Simulations are widely used in the robotics field as useful tools that provide reasonable methods for verifying various aspects of developed device, such as the control system behaviour testing. However simulated results almost always differ from real world experiments because of combination of various aspects such as sensor perception noises, inaccuracy of actuators or faults of control logics. This paper addresses well known simulation related problem in context of the real-time enabled control systems and proposes the possible approach for improving simulated results that is based on the open-source *Player/Stage* software package and it follows principles of separation of the control logics from the simulation core calculations.

Introduction

In recent years the robotics field has become a widespread interdisciplinary industry. In order to satisfy growing demand of precise and stable systems various approaches were introduced, such as simulations. Nowadays simulations are used to verify physical parameters of a robot design as well as to test behaviour of control systems.

Robotic systems (especially industrial production lines) are often considered as real-time systems that have strict deadlines of system response time. Control system design and development for such devices is a challenge for engineers and vast field for scientific investigations. (Choi et al. 2009; Nawawi et al. 2008; Posadas et al. 2008)

Simulation tools can help to increase a development speed and improve a quality of a robot's control system but they also have various shortcomings. Simulations are usually performed on conventional (not real-time) computers by contrast to real-time systems. Task scheduling algorithm (prioritization) is the main difference of these two approaches. Non-real-time systems can have high performance, but it is not guaranteed that the certain task will be completed by the specified deadline.

Related difference is in computing resources available for a system. Simulations are usually run on high-end platforms, while real hardware controllers sacrifice performance in favour of reliability. Also simulated environments usually tend to be perfect, while real devices always will have inaccuracy: sensors will perceive noises while parameters of actuators will differ from vendor's specified values.

As a result, control algorithm well researched and tested in a simulation environment fails on a real hardware because it cannot provide results in predefined timing window (works too slowly) or it is not suitable for the real world environment.

The problem is well known in the robotics research community and therefore many approaches were developed in order to improve quality of simulated results. (Kyriacoua et al. 2008) This paper addresses the analysis of the architecture of the open source simulation package in a scope of the real-time constraint.

Materials and methods

There is a wide spectrum of simulation packages available for robotics researches. High-end software provides maximized accuracy of the simulation (like *Webots*), while others offer usability (like *MRDS*) or different benefits.

Player/Stage open-source software package developed within The Player Project is widely used for multi-robot and distributed sensor research. (Gerkey 2003)

The core of this package is *Player*, the cross-platform robot device interface and server, also called as robotics infrastructure that has to be installed on operation system of robot. Control system (client) communicates with *Player* (server) using TCP (or UDP) based protocol, while the server acts like a proxy between the control program and the real devices, it provides data from the sensors and

forwards commands to the actuators. *Player* uses pluggable system of drivers in order to control specific hardware. Therefore such architecture ensures that a control code is not dependent from an underlying hardware.
Stage is a plug-in for *Player* that provides simulation facilities. It offers special drivers for each *Player's* proxy and they control simulated devices instead of real hardware. From the control system's perspective there is no difference between real and simulated devices (see Fig. 1). *Stage* simulation tools provide additional options for user like environment visualisation, sensors' data rendering and time controls.

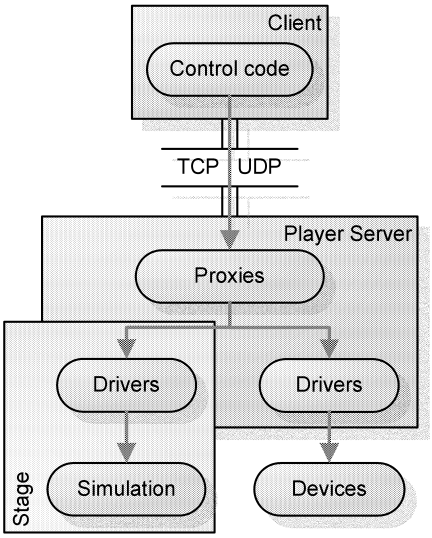


Fig. 1: Player/Stage Architecture

Additionally, *Stage* can perform as stand-alone simulation package (without *Player*) by using pre-compiled plug-in modules. It is possible to create a native library with the special call-back functions and link it to *Stage* simulator in order to control specific model within simulated world.

These two approaches give an additional field for researches and experiments. The second approach is similar to common simulation packages in such way that the control logic (the code in general) is usually integrated into the simulation and executed as a part of every iteration of the simulation. Such approach usually shows high performance and ensures the stability of the simulation process. Theoretically every iteration total execution time consists of the sensor reading time, the control logics execution time (the user code), the actuators update time, then the simulated environment processing time and finally some system related latency. It can be expressed by the following formula:

$$t_{total} = t_{sens} + t_{ctrl} + t_{act} + t_{sim} + t_{sys}, \qquad (1)$$

where t_{total} – total execution time of every iteration, t_{sens} – sensors reading time, t_{ctrl} – control logic (the code) execution time, t_{act} – actuators update time, t_{sim} – simulation environment update time, t_{sys} – system (background) latency.

By contrast, the first approach separates simulation process from the control logics. Therefore these two parts are executed independently that is much closer to the real hardware control process, where control system should handle devices fast enough to ensure required performance and precision. Thus formula (1) transforms to the inequality, where the control logics execution time with additional network latency (if any) should be less then processing time of sensors, actuators and simulated environment:

$$t_{ctrl} + t_{net} + t_{sys1} \le t_{sens} + t_{act} + t_{sim} + t_{sys2}, \qquad (2)$$

where t_{net} – network latency.

This paper proposes that simulation process separation from the control algorithm can be used to improve accuracy of simulation results of real-time control systems.

Results and discussion

In order to confirm previously described hypothesis a simple simulation environment (world) was created.
In some degree it represents a mine with the several halls connected by the narrow passages. There are some objects randomly placed in this mine (e.g. minerals) that a robot should collect (see Fig. 2). Let's assume that the objects are very fragile. In such case the robot should pick up them precisely. The robot has a differential type drive, grippers and various sensors. There are camera based blob finder, which is used to locate the objects, and the sonar sensors used to detect and avoid obstacles. Robot's wandering around the world is based on the steering behaviour (Reynolds 1999). In order to overcome the gripper problems in *Stage* release 3.2.2 there where used artificial object manipulation (through simulation engine) instead of realistically simulated gripping process.

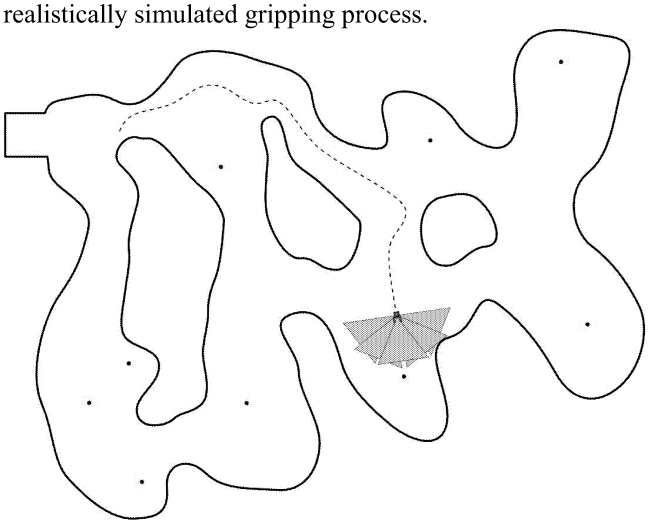


Fig. 2: Simulated World

Two controllers had been created that utilize the same algorithm but use different simulation approaches described before: one is a TCP client to the *Player* server, while other is a *Stage* plug-in. Simulation time resolution as well as

control system cycle latency were set to 100 ms and the simulation was run without acceleration (real-time).

Developed control algorithm was tuned at first to be suitable for the desired environment. Then it was tested in two different controllers. They showed very similar results: the robot controlled by any of these controllers was able to navigate through narrow passages in the environment as well as it successfully collected all objects in the mine. The only difference was related to asynchronous processes in *Player* based controller; and because of them simulated robot occasionally hit the walls.

In order to show undesirable behaviour the control logic processing was artificially prolonged using the sleep function. This demonstrates situations when the control algorithm is correct but it cannot be processed fast enough.
As expected the *Stage* plug-in based controller showed exactly the same results except that simulation processing was slowed down (simulated time unit was longer then real-time unit). Therefore it makes no sense to analyse these experiments.
Contrary the *Player* based controller with the prolonged processing time was unable to navigate the robot through the narrow passages and it often got stalled. The number of experiments where made: for various delay values five simulations where executed up to the 3 simulated minutes and various indicators where recorded (see Table 1).

Table 1: Experiment Results

Latency (ms)	Object hits (qty.)	Wall hits (qty.)	Collected (qty.)
0	4.0	4.8	6.6
100	17.4	7.0	4.4
200	15.8	9.8	1.2
500	5.3	52.0	0.3

The table above shows the robot average object and wall hit count as well as the collected object number in 3 minutes long simulations for the appropriate processing latency. Increased processing time first of all affects object hit count because object collection process was designed to be very precise. Additional processing time increase leads to navigation problems and eventually the robot is unable to get close to the objects through the narrow passages.

Realized experiments confirm the theory described above. *Stage* plug-in based controllers provide stable and precise simulation results with powerful time controlling options (simulated time acceleration is very useful for long running experiments). But it does not meet real hardware controlling approaches. *Player* client based controller cannot provide excellent precision for simulated results and it has processing overhead because of various intermediate software layers, but it reflects real hardware controlling approaches.

In theory simulation time resolution can be lowered down to scheduler resolution of operation system and it mainly depends on performance of system and complexity of simulated environment. Considering all additional latency overheads system can perform on about 30–50 ms resolution. Additionally, real-time enabled operating system can be used to process simulation.

Conclusions

Both described simulation processing methods have strengths and weaknesses. Control logics execution as part of the simulation iteration ensures stable and predictable results. Simulation separation from control logics is conceptually closer to the real hardware control process and theoretically can provide finer results.

Theoretical hypothesis was confirmed by the experimental results that showed clear dependence between the control logics processing performance and the simulated results. Increased execution time of the control system leads to inaccurate behaviour of the controlled device as well as to the decreased useful performance. Described approach can be used for the testing and debugging of the control system before installing it to a real hardware for environments, where timing is essential.

Academic study and publication financed by the project "Support for doctoral studies in LUA" / 2009/0180/1DP/1.1.2.1.2/09/IPIA/VIAA/017/ agreement Nr. 04.4-08/EF2.D1.05.

References

Choi B.W., Shin D. G., Park J. H., Yi S.Y., Gerald S. 2009. *"Real-time control architecture using Xenomai for intelligent service robots in USN environments"*. Intel Serv Robotics (2009) 2, 139–151.

Gerkey B., Vaughan R.T. and Howard A. *"The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems"*. 2003. Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), 317–323.

Kyriacoua T., Nehmzowb U., Iglesiasc R., Billingsd S.A. 2008. *"Accurate robot simulation through system identification"*. Robotics and Autonomous Systems 56 (2008), 1082–1093

Nawawi S. W., Ahmad M. N. and Osman J. H. S. 2008. *"Real-Time Control of a Two-Wheeled Inverted Pendulum Mobile Robot"*. World Academy of Science, Engineering and Technology 39, 214–220

Posadas J.L., Poza J.L., Simo J.E., Benet G, Blanes F. 2008. *"Agent-based distributed architecture for mobile robot control"*. Engineering Applications of Artificial Intelligence 21 (2008), 805-823

Reynolds C. W. *"Steering Behaviors for Autonomous Characters"*. 1999 Proceedings of Game Developers Conference 1999, 763–782.

MANAGEMENT SIMULATION

WE CAN SAVE OUR WORLD
IF WORLDS COLLIDE

William Conley
Departments of Business Administration and Mathematics
University of Wisconsin-Green Bay
Green Bay, Wisconsin 5431-7001
U.S.A.
Conleyw@uwgb.edu

KEYWORDS
Aerospace, Nonlinear programming, Probabilistic, Optimization

ABSTRACT

Before the space age began, anyone studying all of the craters in the earth's moon soon realizes enough of them were caused by asteroids collisions, to make one worry about the long term fate of the earth. Many people believe it's only a matter of time before another big asteroid hits the earth causing untold damage.

However, with the advanced rocket and engine development continuing in the 21st century, the world's space powers should arrange for little asteroids to crash into a big threatening asteroid to move it a degree or two off of its collision course with the earth. In other words, let's arrange asteroids to smash into the big bad asteroid before it has a chance to do this to the earth.

Presented here will be three multivariate nonlinear optimization problems along the line of asteroid defense featuring the Multi Stage Monte Carlo Optimization (MSMCO) solution technique and some comments about some of the inherent technical problems that must be overcome along the way.

HOW MANY BACKUP ENGINES ARE NEEDED?

A small asteroid has been selected for engine mounting and subsequent crashing into a big bad asteroid that threatens the earth. Five competing engines with their costs and failure patterns are given here. Therefore, for each of the five choices of engines for the problem at hand, how many engines (engine plus back-ups) should be put on the little asteroid to have an 85% chance that the asteroid hits its target? The idea here is to do several of these so that it's almost a certainty that at least one or more of the little asteroids hits the big dangerous asteroid.

Specifically, the failure patterns of the five engines and their costs in millions of dollars are:

	Failure Patterns	Costs
1	$.5 + .03 \cdot 2 - .015$	6
2	$.5 + 03 \cdot 3 - .015(2)$	7
3	$.5 + .03 \cdot 4 - .015(3)$	8
4	$.5 + .03 \cdot 5 - .015(4)$	9
5	$.5 + .03 \cdot 6 - .015(5)$	10

So we try to minimize

$$CFP = 6 \cdot X_1 + 7 \cdot X_2 + 8 \cdot X_3 \cdot 9 \cdot X_4 + 10 \cdot X_5$$

Subject to $X_K \geq 0$ and whole numbers

and

$$\prod 1 - (.5 + .03\,(K+1) - .015 \cdot K)\ **X(K) \geq .85$$

where the X_K's are the number of engine K to select plus back-ups and ** means raise to a power.

We use the MSMCO simulation technique, which is further reviewed in (Conley, 2009), with a six stage run drawing 50000 sample answers at each stage. The printout yields a solution of $X_1 = 7$, $X_2 = 7$, ,$X_3 = 6$, $X_4 = 6$, $X_5 = 6$ with minimum total cost of CFP = 253 million dollars.

If putting engines and guidance systems on little asteroids is a problem, an alternative approach could yield the same successful results. That is to launch rockets from the earth with big empty cargo bays that open in the front like the Boeing "super guppy" for rocket and space shuttle parts or the Antanov12 or KC huge lifters that also open in the front.

Then have these rockets catch up to several little asteroids and open up and "swallow" them into their cargo bay to add weight (or kinetic energy) and subsequently send these unmanned rockets themselves to smash into the dangerous asteroid. Please see Figures 1 and 2.

Also, if some of these little asteroids are tumbling in space, gathering them into the cargo bay should stop the tumbling too.

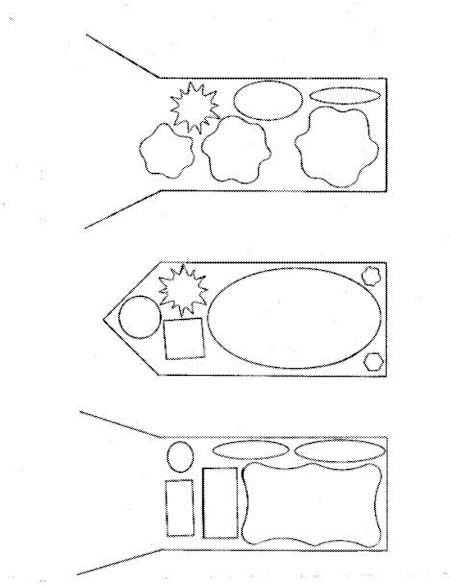


Figure 1. Three Wide Body Rockets Collecting Six Asteroids for More Kinetic Energy for Attack on Big Asteroid

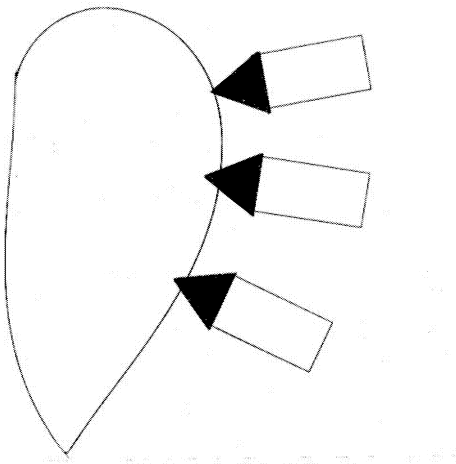


Figure 2. Three Asteroid Laden Rockets Attacking the Big Bad Asteroid

ROCKET BACKUP SYSTEMS WITH ALTERNATIVE SUPPLIER

The actual unmanned rocket that is launched from the earth to deliver the engines to the asteroids (or to collect the little asteroids in its cargo bay) must have a number of backup systems so that the rocket has at least, say a 98% chance of success for example.

Specifically, the six stage rocket taking an engine up to a small asteroid ƵYX to defend the earth from a big asteroid has 10 systems that must work for the rocket to get to little asteroid ƵYX. Therefore, if the system's failure patterns (on a mission of this type) and system's costs are given below, how many backups should each system have so that there is a 98% chance the mission will be successful at minimum cost?

System K1, 1 failure pattern	System K1, 2 failure probability
.4 - .01k1 + .016	.4 - .01K1 + .016 · 2

For K1 = 1, 2, 3, …. 10

and

System K1, 1 cost	System K1, 2 cost
7 + .8K1 - .5	7 + .8K1 -1

Also for

K1 = 1, 2, 3 … 10

So we try to minimize the total cost TC which is Σ the summation 10 right hand side terms

TC = (7 + .8K1 - .5) x (K1, 1) + (7 + .8K1 – 1) x (K1, 2)

Subject to K1 = 1, 2, 3, …. 10

and $1 - XY \cdot \text{ƵY} \geq .98$

where

$XY = \prod (.4 - .01K1 + .016)$ ** X (K1, 1)

$\text{ƵY} = \prod (.4 - .01K1 + .016 \cdot 2)$ ** x (K1, 2)

And again K1 = 1, 2, 3 …. 10

with an MSMCO simulation of 11 stages drawing 45,000 sample answers at each stage. The printout reveals a solution of 660.3 in millions and the twenty variable values of

6	1
4	3
4	3
2	4
3	4
4	2
4	2
1	5
4	2
3	2

NONLINEAR SYSTEMS EXAMPLE

Many aerospace problems involve multivariate nonlinear optimization problems. Let us try to illustrate the generality of the multi stage Monte Carlo Optimization (MSMCO) solution technique by using it to try to find an all whole number solution to the following 12 x 21 system in the range $0 \leq X_i \leq 200$ for the i = 1, 2, 3, ….12 X_i variables to minimize

$$g(X_1, X_2, X_3, \ldots X_{12}) = \Sigma \left| L_j - R_j \right|$$

where L_j and R_j are the left and right hand side of equation j for j = 1, 2, ….21.

We use an eight stage MSMCO program drawing 25,000 sample answers at each stage. This core of the program is run in a loop 25 times and any X_i that occurs at least 20 out of 25 times (80% or more) in the final printout is fixed for

the second run of 25. Then on the second loop of 25 any additional variables whose value is the same on at least 20 out of 25 tries is additionally fixed for the third loop and so on for as many outer loops as are needed.

$$X_5^3 + X_1X_2 + X_8 + X_5 = 7,539,318 \quad (1)$$

$$X_{10}^3 + X_2X_5 + X_2 + X_8 = 2,650,768 \quad (2)$$

$$X_6^3 + X_8X_{10} + X_3 + X_4 = 6,235,434 \quad (3)$$

$$X_9^3 + X_5X_9 + X_{10} + X_{11} = 30,273 \quad (4)$$

$$X_4^3 + X_8X_{12} + X_6 + X_5 = 5,360,534 \quad (5)$$

$$X_7^3 + X_1X_{10} + X_{10} + X_8 = 6,870,633 \quad (6)$$

$$X_8^3 + X_5X_9 + X_1 + X_4 = 74,863 \quad (7)$$

$$X_5^3 + X_7X_{12} + X_3 + X_6 = 7,533,427 \quad (8)$$

$$X_3^3 + X_6X_4 + X_2 + X_4 = 945,163 \quad (9)$$

$$X_8^3 + X_2X_3 + X_9 + X_{11} = 80,167 \quad (10)$$

$$X_{10}^3 + X_8X_{10} + X_{10} + X_5 = 2,634,064 \quad (11)$$

$$X_2^3 + X_7X_{12} + X_6 + X_{11} = 1,524,731 \quad (12)$$

$$X_1^3 + X_6X_9 + X_1 + X_4 = 577,381 \quad (13)$$

$$X_6^3 + X_2X_5 + X_{12} + X_6 = 6,252,247 \quad (14)$$

$$X_1^3 + X_7X_{10} + X_8 + X_9 = 598,077 \quad (15)$$

$$X_{11}^3 + X_6X_{11} + X_7 + X_{12} = 249,945 \quad (16)$$

$$X_3^3 + X_3X_6 + X_{10} + X_7 = 930,849 \quad (17)$$

$$X_2^3 + X_4X_7 + X_7 + X_{12} = 1,554,334 \quad (18)$$

$$X_{12}^3 + X_4X_{11} + X_9 + X_5 = 17,934 \quad (19)$$

$$X_4^3 + X_1X_{10} + X_2 + X_7 = 5,371,134 \quad (20)$$

$$X_5^3 + X_4X_{11} + X_1 + X_5 = 7,540,665 \quad (21)$$

In this case, after the first loop of 25 solution tries six variables had the same value, a total of 20 out of 25, 25 out of 25, 25 out of 25, 25 out of 25, 22 out of 25 and 22 out of 25 solution tries. These were fixed and on the second pass of 25 four more variables had the same value repeat 25 out of 25 tries, 25 out of 25, 25 out of 25, and 24 out of 25 tries.
These were additionally fixed and in the third loop of 25 solution tries (with only two free variables left) the solution was produced each of the twenty-five tries.

It is $X_1 = 83$, $X_2 = 115$, $X_3 = 97$, $X_4 = 175$, $X_5 = 196$, $X_6 = 184$, $X_7 = 190$, $X_8 = 41$, $X_9 = 29$, $X_{10} = 138$, $X_{11} = 62$ and $X_{12} = 19$ with a total error of zero.

Also, if there should be no exact solution this modal averaging can frequently yield a useful approximate solution.

ENGINE FARMS ON ASTEROIDS

Maybe at some point in the future, engine development will have reached the point where we can land on a big dangerous asteroid and put an engine on it powerful enough so that when we run it, it moves the dangerous asteroid off of the collision course with the earth.

Another idea for the future would be to land on the dangerous asteroid and install 100 engines (to greatly increase the thrust) to make it even more certain that it is moved out of its path toward the earth. We could call it an engine farm similar to those windmill farms with 100 windmills that generate electricity on earth.

Also, a smaller "farm" of say 15 or 20 engines, could be put on the smaller asteroids to give them more power to ram the dangerous asteroid in question.

Additionally, if engines are not powerful enough to move smaller asteroids at a useful speed, the small asteroid in question could be cut up into smaller pieces, and some of the engines put on these even smaller asteroids. There is already a fairly open discussion in the space communities about mining the asteroids some day. If mining will be possible, it could certainly be possible to use fractional pieces of them to collide with the big dangerous asteroid.

Certainly the engine development now and in the future will be crucial to the success of this approach to asteroid defense. Also, when the engines to power the little asteroids are sent up to them, these should be unmanned (cheaper and safer) flights. Therefore, much work on guidance and attaching these permanently to the little asteroids is needed.

A ONE DEGREE SHIFT RESULTS

Using the formula for the circumference of a circle we can see if a big threatening asteroid headed toward the center of the earth is knocked one degree off of its path, 100 million miles from the earth, it should miss the earth by about 1.7 million miles (give or take a little, depending on the gravitational pulls it encounters along the way). Similarly, a one degree shift 10 million miles from earth should cause it to miss the earth by 170,000 miles, etc.

Therefore, it is best to hit the problem asteroid far away from the earth or as it's near the earth but receding from it as it goes on another loop.

Now, if the big asteroid is a rock the size of an aircraft carrier and the little asteroid is the size and weight of a ping pong ball that will not budge the big rock at all.

However, if the little asteroids collected in the cargo bay of our big rocket are like a huge sack of bowling balls and the rocket hits the big rock (aircraft carrier size) at 25,000 miles per hour that may be more than enough kinetic energy to accomplish our goal of path diversion. Also, while the world is continuing to develop more powerful engines, remember that the world's current space powers are very adept at using a planet or moon's gravity with well timed engine burners to sling-shot rockets to great speeds.

Therefore, a lot of kinetic energy and momentum calculations should be done with the asteroids and engines available to see how viable this approach is. It will obviously be viable eventually as engine, rocket and space development continue. But we should not be complacent in the 21st century. Let us get ahead of the problem, or, as the title of former flight director of NASA, Gene Kranz's book (Kranz, 2000) about crucial space projects, says it all, "Failure is Not an Option."

OTHER POSSIBLE DEFENSES

The real problem with a large asteroid hitting the earth (Stone, 2008) is that the heat builds up on it to such a great temperature that the rock explodes with the force of a huge nuclear bomb. If the asteroid did not explode, the big thud it made would cause less damage compared with the white hot explosion. Therefore, perhaps in the future we could land on the dangerous asteroid and put a heat shield or cooling mechanism of some kind on it so it would not explode. All possibilities should be considered. Maybe they aren't feasible now, but as science advances in the future, they might be practical.

Additionally, for an asteroid deemed to be too large for the little asteroids to deflect it, it could be bombed into ten or twenty smaller pieces. Then an armada of small asteroids (with engines) could attack the ten or twenty pieces to make sure they are moved off the path toward the earth. Another idea to investigate would be to have the crash into the big asteroid by a small asteroid (or asteroid laden rocket) capable of bouncing off the big asteroid and looping around and crashing into it again and again until either the little asteroid or the crashing rocket falls apart. Perhaps for the rocket it would be good to drop its nose cone away before the crash so it hits the big asteroid with a pliable snub nose that would bounce off more easily. Every possible approach should be considered.

CONCLUSION

A review of the problem of making our world forever safe from a large asteroid crash was presented and three multivariate nonlinear optimization problems were solved along those lines with the general purpose multi stage Monte Carlo Optimization (MSMCO) solution technique. The first one was to solve for how many backup engines to have on the little asteroids to be almost certain of success. The second problem involved how many of ten backup systems to have in the rocket that is actually launched from the earth to start the asteroid defense process. The third one was a general purpose nonlinear optimization problem demonstrating the flexibility of MSMCO.

Also, if fitting engines and guidance systems on little asteroids is currently impractical, rockets with big empty cargo bays could be launched from the earth to collect small asteroids (See figures 1 and 2) to add weight and lots of kinetic energy and then crash the main rocket (or rockets) into the big asteroid to drive it off its collision course from the earth.

Other defenses were mentioned as all forms of defenses against this potentially catastrophic menace should be considered. The key is to plan ahead and be ready when that inevitable day of reckoning for the earth comes about.

REFERENCES

Conley, W. C. "Linear and Nonlinear Input Output Analysis." Proceedings of 7th International Industrial Simulation Conference, ISC, 2009. Loughborough, U.K. June 1-3, EUROSIS-ETI, Gent, Belgium, pp. 39-41.

Kranz, E. F. Failure is not an Option. Simon and Schuster, New York and London, 2000.

Stone, R. "Target Earth." National Geographic. Washington, D.C., August 2008, pp. 134-149.

AUTHOR BIOGRAPHY

WILLIAM CONLEY received a BA in mathematics (with honors) in 1970, an MA in mathematics in 1971 from Western Michigan University and an MS in statistics in 1973 and a PhD in mathematics (computer statistics) in 1976 from the University of Windsor. He has taught mathematics, statistics and computer programming in universities for 35 years. He is currently professor emeritus of Business Administration and Mathematics at the University of Wisconsin at Green Bay. The developer of multi stage Monte Carlo optimization and the CTSP correlation statistics, he is the author of five books and 203 publications worldwide. He is a member of the American Chemical Society, a fellow in the Institution of Electronics and Telecommunication Engineers and a senior member of the Society for Computer Simulation. Career highlights

include presenting a paper at a National Aeronautics and Space Administration (NASA) conference in Houston and a speech at the NASA Goddard Space Flight Center outside of Washington, D.C. on the occasion of the tenth anniversary of the data center.

THE LARGE ASTEROID PROBLEM FROM AN INDUSTRIAL MANAGEMENT PERSPECTIVE

William Conley
Departments of Business Administration and Mathematics
University of Wisconsin at Green Bay
Green Bay, Wisconsin 54311
U.S.A.
E-mail: Conleyw@uwgb.edu

KEYWORDS
Aerospace, Decision making, Optimization, Monte Carlo

ABSTRACT

The world is at work mapping the orbits of the asteroids. If a small asteroid threatened the earth, it easily could be bombed into harmless pebbles by the world's space powers. However, the conventional wisdom is that if a large asteroid were on a collision course with the earth, we could not avoid the impending catastrophe. Presented here will be a way to possibly solve the large asteroid problem and prevent the disaster from ever occurring.

First, two hypothetical multivariate nonlinear optimization problems involving large asteroid defenses will be presented and solved with the simulation technique multi stage Monte Carlo Optimization (MSMCO). Then a review of some of the work already being done on this problem will be mentioned. Finally, the industrial management approach to organizing the solution to this problem will be addressed.

INTRODUCTION

The asteroid Apophis may possibly threaten the earth (Stone, 2008) in 2029 or again in 2036. It is less than 1000 feet across, but still large enough to do enormous damage to the earth. The asteroid that hit Siberia in 1908 was only about 150 feet across and it leveled a forest of several hundred square miles.

However, the world's space agencies (European, United States, Russia, China, India, Japan, Brazil and others) have enough rockets and bombs to blow up a 1000 foot rock before it gets anywhere near the earth.

The real perceived asteroid problem is a threat from a much larger asteroid. What should be done in that case is to put engines on some of the smaller asteroids and bring them into range of the orbit of the big bad asteroid. Then the little asteroids can be sent into a coordinated attack on the big asteroid. These kinetic energy weapons (small asteroids) are already up in outer space, so they do not have to be launched from the earth (no heavy lifting). All they have to do is hit the big asteroid with enough kinetic energy to knock it a fraction of a degree off of its orbital course, millions of miles before it approaches the earth. Then it will miss the earth. The small asteroids do not have to destroy the big asteroid. They just have to hit it with enough kinetic force to move it away from its collision course with the earth. Let us look at two hypothetical examples first. Then we can discuss the industrial engine development and management expertise that may be needed to coordinate the permanent defense of the earth from this problem. The goal is a safer, more secure world.

A FOUR ASTEROID COORDINATED ATTACK

It is New Year's Eve at some time in the future and a big asteroid is getting closer to the earth. However, the space agencies of the world know that this asteroid will be at the rendezvous point 1 (1000000, 1000000, 1000000) in miles on April 14 at 4:00 AM Greenwich Mean Time (9000000 seconds into the new year) where the third coordinate is on the axis from the north pole of the earth to the north star and the first two coordinates are on a plane tangential to the north pole and perpendicular to the third axis. It is traveling at 25,000 miles per hour straight at the North Pole. Therefore, it will also be heading toward the rendezvous points,

Point 2 (990000, 990000, 990000)
Point 3 (980000, 980000, 980000)
Point 4 (970000, 970000, 970000)

at about 48 minute intervals.

However, the space agencies have four smaller asteroids with engines and guidance systems on approximate parallel courses to the large threatening asteroid. Let x_i be the time into the new year $0 \leq x_i \leq 9000000$ when the little asteroids turn on their engines and head toward the four rendezvous points. When they do this they will be at the points (from largest to smallest asteroid in order)

Biggest

$D1 = 500000 + .05x_1$
$D2 = 480000 + .052x_1$
$D3 = 300000 + .10x_1$

2nd Biggest

$E1 = 600000 + 03x_2$
$E2 = 650000 + .031x_2$
$E3 = 395000 + .09x_2$

3rd Biggest

$F1 = 725000 + .016x_3$
$F2 = 700000 + .02x_3$
$F3 = 450000 + .049x_3$

Smallest One

$G1 = 900000 + .005x_4$
$G2 = 875000 + .004x_4$
$G3 = 700000 + .0045x_4$

and they will head toward the four rendezvous points in order largest to point 1 at 1000 mph getting 50 miles per unit of fuel. The second largest one will head to point 2 at 5000 mph getting 15 miles per unit of fuel. The third largest will head toward rendezvous point 3 at 6000 mph getting 14 miles per unit of fuel. Finally, the smallest asteroid will head toward point 4 at 2500 mph getting 35 miles per unit of fuel.

Therefore, when into the new year should the four little asteroids turn on their engines (x_1, x_2, x_3, x_4) in seconds to head toward their target and minimize fuel usage and arrive within 9000000 seconds or less into the new year. If they arrive early they can turn and head up the course of the big asteroid to make more of an impact and adjust for its deviation from its course from previous crashes. The goal is to knock it into a path that takes it away from the earth or hit it so hard that it breaks apart into harmless smaller pieces. If the smaller pieces are not harmless, they could be bombed with conventional weapons or more asteroids.

This hypothetical problem started the attack at a distance of 1,732,050.81 miles from the earth. A real attack should probably be done many more millions of miles away from the earth or near the earth only as the big asteroid is heading away from the earth (for another loop) so if the large asteroid is knocked slightly off if its course, that will magnify how much it misses the earth by the next time around.

A multi stage Monte Carlo Optimization (MSMCO) simulation was set up and run to solve this problem. A 25 stage simulation (with the 4 time variables) was set up starting with a search region of $0 \le x_i \le 9000000$ seconds for i=1, 2, 3, 4. A sample of 25,000 sets of 4 numbers was looked at at each of the 25 ever narrowing search stages and the sample answer using the least fuel within the time constraints was stored as the simulation proceeded. This program was run twice (in a few seconds of computer time) producing similar answers of x_1= 8,028,783.51 seconds, x_2= 7,612,054.99 seconds, x_3=8,892,627.44 seconds, x_4=8,653,953.71 seconds and second answer x_1= 8,025,414.55 seconds, x_2= 7,607,305.59 seconds, x_3=8,892,627.44 seconds, x_4=8,653,955.72 seconds, in each case using up a minimum of 37,310.8 units of fuel.

The possible problem of mounting engines on a little asteroid could be addressed by launching a rocket, whose storage bin opens up and collects the little asteroid. Then the rocket could proceed to the big dangerous asteroid and smash into it with the extra mass for maximum results.

A NINE ASTEROID CORRAL NEAR A SPACE STATION

Chapter 10 of (Conley, 1983) has the Star Walker Space Corporation launching five rockets from different locations to rendezvous fairly close together in space.

Let us continue with our large asteroid defense by trying to corral nine small asteroids with engines on them that are in need of the sophisticated guidance systems that the workers can bring over from the space station. Therefore, our problem is to move all of our asteroids and the space station close enough to each other to make it easier for the workers (from the space station) to install the guidance systems and sophisticated engine controls.

The C_1, C_2, and C_3 are the coordinates for the space station (in Table 1), while the D's through R's are the coordinates for the nine asteroids. They can only move in an approximately linear fashion until the sophisticated controls are installed.

Table 1: The Paths Without The Engines On

$C_1 = -50{,}000{,}000 + 6x_1$	$H_1 = 1{,}540{,}000 - .7x_6$
$C_2 = -82{,}000{,}500 + 4x_1$	$H_2 = 800{,}000 - 1.6x_6$
$C_3 = 91{,}000{,}990 - 8x_1$	$H_3 = 300{,}000 - 5x_6$
$D_1 = 150{,}000{,}000 - 7x_2$	$K_1 = -1{,}600{,}000 + 1.8x_7$
$D_2 = -167{,}000{,}300 + 3x_2$	$K_2 = -2{,}500{,}000 + 5.3x_7$
$D_3 = 300{,}900{,}400 - 11x_2$	$K_3 = -3{,}600{,}400 + 6.8x_7$
$E_1 = 1{,}240{,}000 - .8x_3$	$L_1 = -1{,}010{,}100 + 1.4x_8$
$E_2 = 800{,}000 - x_3$	$L_2 = 1{,}527{,}000 - 3.6x_8$
$E_3 = 3{,}300{,}300 - 7x_3$	$L_3 = -2{,}600{,}999 + .3x_8$
$F_1 = -16{,}000{,}000 + 1.6x_4$	$M_1 = -6{,}000{,}000 + 4x_9$
$F_2 = -21{,}000{,}000 + 8.6x_4$	$M_2 = -7{,}111{,}888 + 3x_9$
$F_3 = -41{,}600{,}000 + 6.5x_4$	$M_3 = 8{,}300{,}000 - 8x_9$
$G_1 = -19{,}000{,}000 + .6x_5$	$R_1 = 15{,}000{,}000 - 9x_{10}$
$G_2 = 16{,}825{,}000 + 5x_5$	$R_2 = -17{,}915{,}000 + 1.8x_{10}$
$G_3 = -25{,}999{,}000 + 5x_5$	$R_3 = 8{,}500{,}000 - .35x_{10}$

Therefore, we seek to minimize subject $F(x_{1....}x_{10}) = \sqrt{\Sigma}(C_i - D_i)^2 + \sqrt{\Sigma}(C_i - E_i)^2 + \sqrt{\Sigma}(C_i - F_i)^2 + \sqrt{\Sigma}(Ci - Gi)^2 + \sqrt{\Sigma}(C_i - H_i)^2 + \sqrt{\Sigma}(C_i - K_i)^2 + \sqrt{\Sigma}(C_i - L_i)^2 + \sqrt{\Sigma}(C_i - M_i)^2 + \sqrt{\Sigma}(C_i - R_i)^2$ subject to $0 \leq x_i \leq 100{,}000{,}000$ where the x_i's are how long they do their engine burn in hundredths of a second, and the C's through R's are in units of feet.

We use a ten variable thirty stage multi stage Monte Carlo Optimization simulation, drawing 10,000 sample answers at each stage. Multi Stage is a fairly versatile solution technique (Conley, 2008). The Partial Stage Printout and Solution Follows:

Stage 1	1,152,910,991
Stage 2	617,090,418
"	"
"	"
"	"
"	"
"	"
"	"
"	"
"	"
Stage 30	M=373421910

x_1 = 11209998, x_2 = 25789825, x_3 = 752638, x_4 = 1612216, x_5 = 11987996, x_6 = 1649480, x_7 = 0, x_8 = 10875746, x_9 = 108757, x_{10} = 0

Therefore, the space station should run its engines for x_1=112,099.98 seconds and x_2=257,898.25 seconds is how long to run the engine on the first asteroid. The x_3=7,526.38 seconds is the run time for another asteroid, etc. The two asteroids that have zeros, x_7=0, x_{10}=0 seconds means those asteroids should stay where they are. They can do this by running and reversing their thrusters as needed.

The minimum distance in stage 30 of the printout is M=373,421,910 total feet or about 70,000+ total miles. Therefore, each asteroid will be a little less than 8000 miles from the space station (on the average), which is manageable for the guidance installers.

ENGINE DEVELOPMENT

The National Aeronautics and Space Administration (NASA) of the United States, the European Space Agency, and the space agencies of other countries and their engine contractors and suppliers should look into the engines and guidance systems that are currently available to see if some of them would work on the small asteroids. Also, NASA announced to the press recently that their new 1-X rocket is for long distance flights, so that is a possible help with this problem.

NASA has already landed on an asteroid with its Near Earth Asteroid Rendezvous Shoemaker Mission of 1996 (N.E.A.R. Shoemaker). Also, data from NASA's Deep Impact 2005 mission and the European Space Agency's close fly by of Halley's Comet (Giotto Mission, 1986) may be useful in studying collisions in space.

However, new engine development may be necessary to deal effectively with this problem. Also, some type of beacon or homer could be attached to the large threatening asteroid, to guide the little asteroids in attacking it in order to move it off line from its dangerous course.

One theory is that there were only a few hundred asteroids billions of years ago, but that they kept crashing into each other at random. Now there are millions of little asteroids as a result of those random crashes. Therefore, in future years let us arrange a few nonrandom crashes of little asteroids into a big threatening asteroid to protect and defend our earth.

As the world's space agencies look for new and meaningful goals for future space exploration, this should be a high priority, because it is not a question of if a large asteroid will threaten the earth again, but only a question of when. We should be prepared to stop it once and for all time.

MANAGEMENT EXPERTISE

One could say that the space age began on October 4, 1957 with Russia's launch of Sputnik and subsequently Yuri Gagarin being the first human to go into space. Russia also did the first space walking (floating). Then NASA, in December 1968, had the Apollo 8 mission (Borman, Lovell, and Anders) orbit the moon and sent back the stunning photographs of our fragile and beautiful earth. That was followed by a moon landing in July 1969. Since then, many astronauts (and private citizens) have gone into space. Also, the European Union's Space Agency has launched many missions from its base in French Guiana. Japan, India, and China have had space launches. Certainly these many dozens of space projects have found the world admiring the bravery and skill of the astronauts and cosmonauts and the technical expertise and scientific brilliance of the engineers, scientists and technicians who have worked on these projects.

However, what is sometimes overlooked is the tremendous management expertise that is necessary to organize and carry out these large-scale missions. Richard Stone (Stone, 2008), the Asian editor of Science Magazine, points out in his overview of the large asteroid "problem," that individuals, private corporations and space agencies are at work to find all of the large asteroids (big enough to be a potentially serious problem some day) and map their orbital paths as accurately as possible. He implies that

there should be much more funding and assistants to help these people so that the world can understand how many of these problem asteroids there are and how much time we have before we have to take action.

The engine development and selection is crucial for outfitting the small asteroids so they can get in the way of the big asteroids. Also, all other means of dealing with this problem should be looked at.

Perhaps certain asteroids will require one solution and others will require another. Also, depending on what the mapping reveals, this may not be an urgent problem, but it is very important. We probably should prepare quickly just in case the mapping misses a "rogue" asteroid that is hidden somewhere. Note that if two asteroids that suddenly crash into each other and deflect one of them toward the earth at say 10,000 miles per hour, it could hit the earth in about a year, even from 90 or 100 million miles away. Therefore, we should not be complacent in dealing with this dangerous problem.

CONCLUSION

Presented here is a new approach to asteroid defense. There is so much kinetic energy (trillions and trillions of joules of it) already in outer space, we only need to harness a little bit of it (selected small asteroids) to ram into a big asteroid and knock it off its collision course if it threatens the earth. Work needs to be done on engine development, guidance, and continually mapping of outer space for the paths of the big asteroids. Also, considerable scientific, industrial and management expertise needs to be brought to bear on this problem.

Two hypothetical examples also were presented here to stimulate further research. As a bonus, the MSMCO solution technique used here should help on other large scale planning problems (Conley, 2008) because it is a general purpose solution technique (Conley, 2009). One thing the managers should do is consider this new approach very seriously, along with other possible solution approaches being worked on all over the world.

Some day we might even be able to develop an engine so powerful that we can land on the large threatening asteroid, and attach the engine to it. Then when we run the engine directed perpendicular to its path of motion, it will drive the large asteroid out of its path toward the earth, without having to orchestrate these collisions with the smaller asteroids. Every viable option needs to be considered.

REFERENCES

Conley, W. C. BASIC II ADVANCED. Petrocelli Books, New York, Princeton, 1983.

Conley, W. C. "Linear and Nonlinear Input Output Analysis." Proceedings of 7th International Industrial Simulation Conference, ISC, 2009. Loughborough, U.K. June 1-3, EUROSIS-ETI, Ghent, Belgium, pp. 39-41.

Conley, W. C. "Ecological Optimization of Pollution Control Equipment from a Management and Planning Perspective." Journal of Systems Science, Vol. 1, 2008, pages 1-7.

Stone, R. "Target Earth." In National Geographic, Washington, D.C., August 2008, pp 134-149.

AUTHOR BIOGRAPHY

WILLIAM CONLEY received a BA in mathematics (with honors) in 1970, an MA in mathematics in 1971 from Western Michigan University and an MS in statistics in 1973 and a PhD in mathematics (computer statistics) in 1976 from the University of Windsor. He has taught mathematics, statistics and computer programming in universities for 35 years. He is currently professor emeritus of Business Administration and Mathematics at the University of Wisconsin at Green Bay. The developer of multi stage Monte Carlo optimization and the CTSP correlation statistics, he is the author of five books and 203 publications worldwide. He is a member of the American Chemical Society, a fellow in the Institution of Electronics and Telecommunication Engineers and a senior member of the Society for Computer Simulation. Career highlights include presenting a paper at a National Aeronautics and Space Administration (NASA) conference in Houston and a speech at the NASA Goddard Space Flight Center outside of Washington, D.C. on the occasion of the tenth anniversary of the data center.

BUSINESS PROCESS MODELLING

STORYBOARDS IN BUSINESS PROCESS MODELING

Veronika Juchova, Svatopluk Stolfa, David Ježek, Ivo Vondrak
Department of Computer Science
VSB - Technical University of Ostrava
17.listopadu 15, Ostrava-Poruba
Czech Republic
E-mail: { veronika.juchova, svatopluk.stolfa, david.jezek, ivo.vondrak }@vsb.cz

KEYWORDS
Business Process, Business Modeling, Storyboard.

ABSTRACT

Effective business process description is one of the most important tasks to achieve the effectiveness of the company. The typical problem that arises when describing the processes to stakeholders is the lack of understanding to the formal business process notations and definitions. That is why we need a business modeling method that is easy to understand for a stakeholder and that has a formal background for an analyst. In this paper we are introducing the concept of storyboards. Storyboard is a sequence of pictures and multimedia that represent one possible flow of the process. Each picture represent process activity performed by actor and is described by multimedia and/or text description. Combination of the well-structured storyboard flows describes the whole process and can be transformed into the desired formal notation e.g. UML activity diagram, BPMN etc.

INTRODUCTION

Managing a company is not easy. Managing the company effectively is even harder. Description and effective enactment of company business processes is one way how to do management effectively. Business processes are formally described, enacted and then controlled by managers. Everything works effectively and everybody is satisfied.

The problem arises when the company processes must be presented to the people that are not familiar with the standard business modeling methods and notations. Communication between business process analyst and stakeholders of the processes is important part of the process description. Process analyst has to be introduced to the company environment. Company representatives and employees must understand the description of the processes. Common language is the notation or tool that is used for the process description. Business analysts are using many different formal or informal methods and notations that are usually not easy to understand. Thus, the stakeholders of the processes have to learn new language/notation to understand the company processes to be able to perform them.

In our opinion, more effective way is to develop and use an intuitive method that is very easy to understand, is formal and can be automatically transformed to standard modeling notations. This transformation request is important for the usage of methods that are already used by process analyst to analyze and simulate business processes.

This paper describes the storyboard method that is being developed at our university. In section two we describe our understanding of the business modeling and definitions of the business modeling terms. In next sections we introduce our proposed method and show transformation to the semi formal description by UML activity diagram. Last section summarizes the work and proposes future developments.

BUSINESS MODELING

What is the business modeling? Business modeling is the activity that is performed by the business analysts. Analysts try to model the current state of the enterprise business and also its predicated future state. Business process modeling (BPM) tools are used to do that. BPM tools capture the business flow of activities by the business processes and describe the many additional pieces of information by other techniques (Laguna, 2004; Penker, 2000; Vondrak, 1999).

The purpose of the business modeling is to model the reality of the enterprise business. Models could be used then in many ways. For example for the *Business Process Re-engineering* (BPR) that are the methods that support activities by which an enterprise reexamines its goals and how it achieves them, followed by a disciplined approach of the business process redesign. Business modeling is also the first step in the process of *Enterprise Resource Planning* (ERP) information systems implementation. ERP information systems integrate all the manufacturing and related applications for an entire enterprise. Business processes are also used for the Workflow Management Systems (WMF) systems that are generic software systems used for the definition, management, enactment, and control of the business processes.

Now, let's introduce some definitions of the basic terms used in the area of the business process modeling.

Business process – is a set of one or more linked procedures or activities, which collectively realize a business objective or policy goal.

Activity – is the description of the task that represents one atomic step of the business process.

Role – a group of users, defined within an organization. Role has a defined set of skills that are needed to perform a particular activity.

Resource – any persons, equipment, or material needed to perform an activity.

Those are the most important definitions to understand the process description by the various modeling techniques.

There are three approaches that could be used to model the business process. These approaches are based on the type of the view that we need. These three approaches are:

Functional View. The functional view is focused on the activities as well as on the entities that flow into and out of these activities. That means how the input is transformed to the required output.

Behavioral View. The behavioral view is focused on when and/or under what conditions activities are performed. The behavioral view captures the control aspect of the process model.

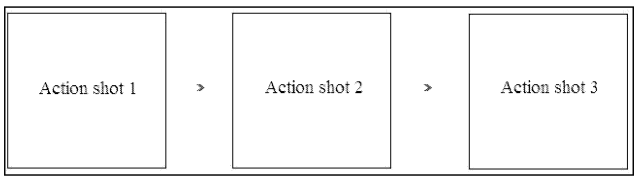
Structural View. The structural view is focused on the static aspect of the process. It captures objects that are manipulated and used by a process as well as the relationships that exist among them.

Each of them has its own reason why it should be used. The main difference is at the highlighting of the particular characteristic that is important for the creation of the process and suppression of the others. When the analyst chooses to highlight e.g. the behavioral characteristic then he chooses some of the methods and tools that are intent to do that rather than to use the inappropriate method that could be confusing to the both the analyst and their internal or external customers. For example functional view can be described by IDEF0 method, behavioral view by UML activity diagram, BPMN, or EPC, structural view by UML class diagram etc. There are many other methods and approaches that can be used e.g. Petri Nets, Workflow Nets, BPML, etc. (Vondrak, 1999; van der Aalst, 1999; van der Aalst, 2000; van der Aalst, 2002; BPMN, 2010; IDEF, 2010; OMG, 2010). Thus, the intent of the business modeling activity in the company is important for the appropriate understanding and description of the processes and their future usage.

STORYBOARD METHOD

Storyboard

Storyboard is the multimedia picture frame that contains sequence of successive shots. Each shot represents one business process fragment – one action that is performed in business process. Actions are sequenced the same way as they are performed in the business process. Audio or video files can be added to each action shot.



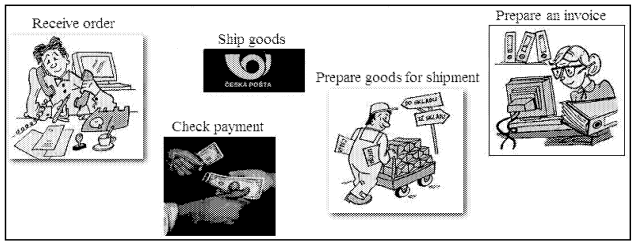
Figures 1: Sequence of successive shots

How to create storyboard

Employees are responsible only for some parts of the processes. Each part is one action, one storyboard shot. First step is to get a description of one or more actions from each employee. These shots are the basic elements of the storyboard. Actions are not sequenced in this step.

Example

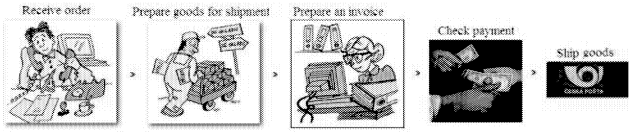
We want to create storyboard of business process called **Order fulfillment**. Stakeholders from the company that realizes this process tells us basic information about the process. Orders are received by telephone, operator receives the order. Then, it is necessary to prepare goods for shipment, prepare an invoice, ship goods and check payment. Now, we are able to create storyboard shots.



Figures 2: Unsorted action shots of Order fulfillment process

Lets describe the action prepare goods for shipment in detail. First of all, we have to assign a unique name: '**Prepare goods for shipment**'. Activity '**Prepare goods for shipment**' is the responsibility of storekeeper. He needs filled out order for the preparation of the goods. That order contains information about goods. As soon as the order is ready, goods can be prepared for shipment. Whole procedure of preparing goods lasts approximately 30 minutes. Stores foreman can keep an eye on storekeeper during his work.

When all shot descriptions are known, we are able to put process together. This means to make a transformation of independent shot components to ordered sequence of shots.



Figures 3: Storyboard of Order fulfillment process

Storyboard content

Storyboard is a multimedia picture frame that presents actions executed in business process in time-sequential manner. This is very simple description of process that is not very formal and strict. Action scenario should be added to describe the process in more detail. The scenario contents detailed description of the activity execution. Storyboard description by scenarios can be completed with audio, video and other files that are comp.

Process name – each business process is identified by unique name within the company environment. In our example, the process name is **Order fulfillment**.

Action name – each action is identified by unique name. Example – **Prepare goods for shipment**.

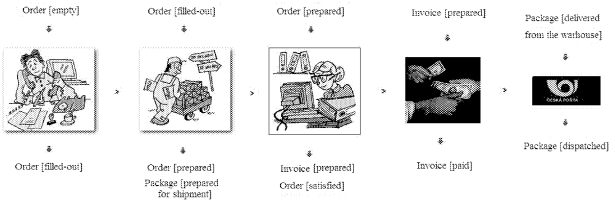
Worker – One of the main attributes in action is role. Role is represented by worker, or group of workers that takes care of the same actions. It must be clear which role takes

care of action execution. Example – **Storekeeper** takes care of 'Prepare goods for shipment'.

Action input/output – input and output elements are very important. They are called artifacts. Artifacts are entities that are created, used or modified during the process. For example, it might be document, device, other file etc. We must be able to determine all the entities and their statuses from the description of the activities and we must be able to state which entities are necessary for the execution of the process and which of them are process execution products. Each entity has its unique name and it can occur in several different statuses during process execution.

Example - Storekeeper needs filled-out order to prepare ordered goods for shipment. An input entity is **Order** and its status is **filled-out**. Afterwards, when the goods are prepared **order** status is changed to **prepared**. Prepared goods are represented by the second output entity **Package** that is prepared for shipping – status **prepared for shipping**.

We can add inputs and outputs to the storyboard picture. Storyboard with input and output objects describes the process more clearly.



Figures 4: Storyboard with inputs and outputs

Action execution time – all actions need definite amount of time for execution. If we are able to determine the exact time, we write it down to this attribute. If we are not able to determine the time exactly, we can write down an estimated time. In our example, the execution time is approximately 30 minutes.

Access privileges – Only workers that are included in roles can be responsible for activity execution. There are specified access privileges for these roles. Access privileges specify which roles have what access to the activity. Example - **Storekeeper** and **stores foreman** have access to the activity prepare for shipment.

Crossing – The same action can appear in several different processes. We must write down for each action that this action appears in other processes. Example – the action **Prepare goods for shipment** can appear in **Sales return** process too.

Parallel shot – parallel shot that includes the action scenario exists for each action shot. Highlighted key information is shown there. All conditions and restrictions are described, including conditions and restrictions that are evident from the picture.

We write down:

- activity name
- process name that contains the action
- roles
- input and output artifacts
- activity execution time
- access privileges for roles
- names of other processes that include this action

	Activity name	Prepare goods for shipment
	Process name	Order fulfillment
	Scenario	Storekeeper is responsible for the preparation of goods that are specified by the filled-out order. As soon as the goods are collected and prepared, package can be sent. Whole procedure of preparing goods lasts approximately 30 minutes. Stores foreman can keep an eye on storekeeper during his work.
	Roles	Storekeeper
	Inputs	Order [filled-out]
	Outputs	Order [prepared], Package [prepared for shipment]
	Executing time	30 minutes
	Access privileges	Storekeeper, Stores foreman

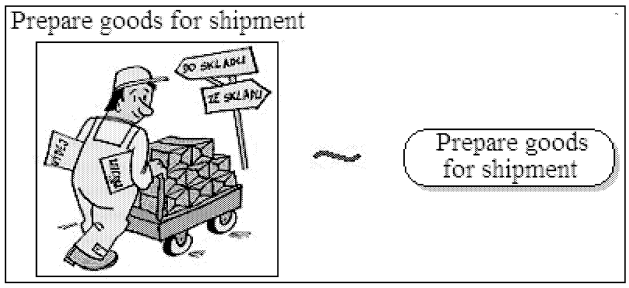
Figures 5: Activity Prepare goods for shipment parallel shot

TRANSFORMATION TO UML – ACTIVITY DIAGRAM

UML activity diagram describes business process by activities that represent tasks performed in the process and transitions between these tasks. Transition is executed after the relevant activity is finished (OMG, 2010).

Storyboards can be used as an alternative to activity diagrams. Since storyboards are easier to understand, they can be used for business process specification when analyst discusses with stakeholder. Except activity diagrams, storyboards and tools that will support them keep open the possibility of attaching auxiliary files like video files, audio files and documents.

Activity – each shot in storyboard represents one activity in activity diagram.



Figures 6: Activity Prepare goods for shipment

Swimlines – Swimlines are related to relevant activities and they are used to identify roles that are responsible for the realization of these activities. Information about execution responsibility in action that is in storyboard is described by attribute 'role' in parallel shot that corresponds with the activity.

Artifacts – are important elements in activity diagrams. Artifacts are used during process execution and are necessary process components.

These entities are described in activity shot (picture) and in parallel shot description in storyboard.

Decision – decision in activity diagram is a symbol that is used for the description of the beginning of alternative scenarios.

Storyboard method has one alternative storyboard for each alternative pass of the process. This means, that we create new storyboards for every alternative scenario that can occur during its execution.

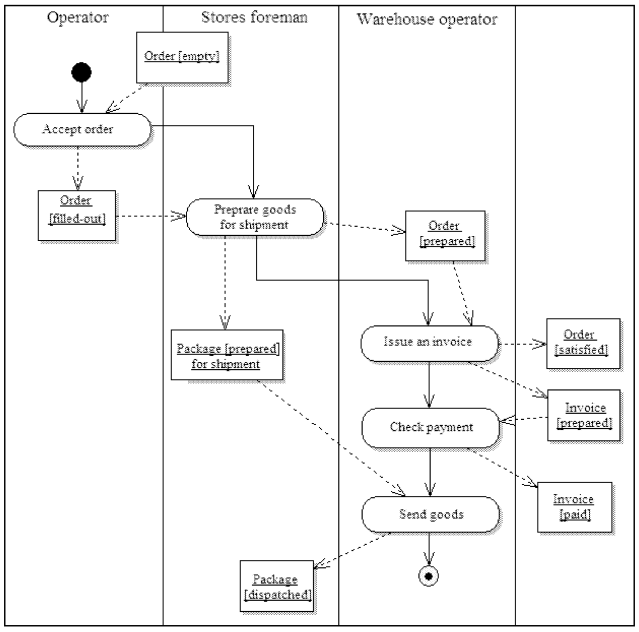
Synclitism – In the real world, several activities can be executed concurrently and process cannot continue until all concurrent activities are finished.

Storyboard method inserts all concurrent activities into one shot.

Table 1: Mapping from the storyboard to the activity diagram

Process name	Activity diagram name
Shot	Activity
Worker	Role
Action input element	Activity input artifact
Action output element	Activity output artifact
Crossing	Decision
Actions synclitism	Fork

Result



Figures 7: Resulted activity diagram

Semi automatic activity ordering

Previous chapter describes mapping from storyboards to activity diagrams. Particular element from activity diagram is clearly defined for each source element form storyboards, but there is no transformation rule for ordering of activities.

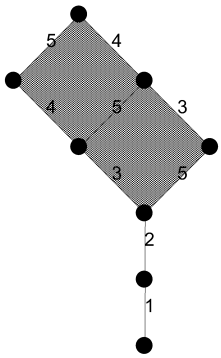
There is a method described in (Jezek, 2003) that constructs higher dimensional automaton (Pratt, 1991) from input and output objects of activities. Constructed automaton represents a process composed of activities.

Higher dimensional automaton is composed of geometrical elements - cubes (generally n-dimensional cubes), where each geometrical element, such as vertex, edge, surface, etc., represent the state of the process. For example, each edge represents the state of the process, where exactly one activity is performed. Surface represents the state, where exactly two activities are preformed concurrently. One of the big advantages of the higher dimensional automata is strong mathematical theory that can be gained through Chu spaces (Jezek, 2004) (Pratt, 2000)(Gupta, 1994).

Table 2: List of activities and input and output objects

Activity	Inputs objects	Outputs objects
1. Accepted order	Order [empty]	Order [filled-out]
2. Prepare goods for shipment	Order [filled-out]	Order [prepared], Package [prepared for shipment]
3. Issue an invoice	Order [prepared]	Invoice [prepared]
4. Check payment	Invoice [prepared]	Invoice [paid]
5. Send goods	Package [prepared for shipment]	Package [dispatched]

When we apply method form (Jezek, 2003) to our example (table 2), we get higher dimensional automata (figure 8) that describes quite different ordering of activities than the manually created activity diagram form figure 7. The process described by the automaton from figure 8 starts with sequential ordering of activities **1. Accepted order** and **2. Prepare goods for shipment** after that the section with two concurrent threads starts. First thread consists of the activity **5. Send goods** and the second thread from sequential ordering of activities **3. Issue an invoice** and **4. Check payment**.



Figures 8: Higher Dimension Automaton

The resulted automaton is correct, but there is problem with input data (table 2) for construction algorithm. Problem rest in absence of an object generated by activity **Check payment** and consumed by activity **Send goods**.

We can interpret that difference as a fault in previous analysis of the process. The fault can be removed in two ways:

- Missing object should be added. For example, object **Order payment confirmation record** that is produced by the activity **Check Payment** and consumed by the activity **Sent goods** or we can use object **Invoice [paid]** in the same way.
- We can search optimization of the process with the concurrent execution, because there is no strong dependency between the activities.

CONCLUSIONS

This paper presents a storyboard method that is being developed at our university. Our intent was to introduce the concept of this method as a communication language between business process analysts and stakeholders of the processes. The concept is based on the need of notation and method that is easily understandable, formal, and is transformable to other notations and methods for formal analysis and future use.

Pure storyboard is a communication platform between the analyst and stakeholder. Storyboard is informal, simple and easily understandable. Formal description of the process is achieved by the addition of other attributes, artifacts that are consumed and produced by the process activities, roles that are responsible for the performance of the activities etc. Combination of the several extended storyboards to the one complex description of reality allows transforming the complex description to any other semiformal or formal description. This procedure was demonstrated by the transformation of the storyboards to the UML activity diagram notation

Since the development of our method is at the beginning, no complex studies were performed. It is obvious that the future development of our method will be difficult and the problems will arise. The most important problem seems to be the connection between the external informal and internal formal representation of the process in this method. The right definition of the connection between these two descriptions has the significant impact for the future usage and transformation of this method to other specifications. We are also developing a tool that supports implementation of our method and guide user through the description process. Development of the library that is describing the standard approaches and presents analysis patterns and processes is being developed as the integral part of this tool. We will be able to describe, manage, analyze and transform storyboards with this tool.

The priority of our research is to develop a method that is not only ‘yet another analysis method’. Our goal is to present our method as a communication platform that offers opportunities to combine informal and formal approaches and opens up formal analysis of business processes to the wider audience.

REFERENCES

Laguna, Manuel and Marklund, Johan. 2004. *Business Process Modeling, Simulation, and Design* . s.l. : Prentice Hall, 2004. ISBN-10: 0131099795.

Penker, Magnus and Eriksson, Hans-Erik. 2000. *Business Modeling With UML: Business Patterns at Work.* s.l. : Wiley, 2000. ISBN-10: 0471295515.

Vondrak, I.; Szturc, R.; Kruzel, M. 1999. Company Driven by Process Models, In *European Concurrent Engineering Conference ECEC ’99*, SCS, Erlangen-Nuremberg, Germany, pp. 188-193, 1999

van der Aalst, Wil. 1999. Formalization and Verification of Event-driven Process Chains. Information and Software Technology, 41(10):639-650, 1999.

van der Aalst, Wil. 2000. Workflow Verification: Finding Control-Flow Errors using Petri-net-based Techniques. In *Business Process Management: Models, Techniques, and Empirical Studies*, volume 1806 of Lecture Notes in Computer Science, pages 161-183. Springer-Verlag, Berlin, 2000.

van der Aalst, Wil and van Hee, Kees. 2002. Worklflow Management, Models, Methods, and Systems. MIT Press, 2002

BPMN. 2010. Business Process Modeling Notation [online] 2010-01-30. [Cited: 2010-01-30]. http//www.bpmn.org.

IDEF. 2010. Integrated Definition Methods [online] 2010-01-30. [Cited: 2010-01-30]. http://www.idef.com.

OMG. 2010. UML® Resource Page [online] 2010-01-30. [Cited: 2010-01-30]. http://www.uml.org.

Ježek, D., & Vondrák, I. (2004). *HDA and Resource Modeling in Business Process.* Ghent: ECEC 2004.

Vondrák, I., Ježek, D., & Snášel, V. (2003). Automated Construction of Model of Concurrency in Business Process Modeling. ECEC 2003.

V. R. Pratt. Modeling Concurrency with geometry. In Proc. 18th Ann. ACM Symposium on Principles of Programming Languages, page 311-322, January 1991.

V. R. Pratt. Higher dimensional automata revisited. In Math. Structures in Comp. Sci. Vol. 10, page 525-548, 2000

Gupta, Vineet. Chu Spaces: A Model Of Concurrency. 1994. p. 98. CS-TR-94-1521.

ACKNOWLEDGEMENT

This research has been supported by internal grant agency of VSB-TU of Ostrava - SP/2010214 Modeling, simulation and verification of software processes II.

AUTHOR BIOGRAPHY

VERONIKA JUCHOVA was born in Havirov, Czech Republic. Since 2002 she studied computer science at VSB – Technical University of Ostrava, where she obtained her degree in 2007. Then she started her doctoral studies at VSB – Technical University of Ostrava. Since 2006 she has been working for VRK, Robert Kubacek, Ostrava.

DAVID JEZEK was born in Ostrava, Czech Republic and went to VSB – Technical University of Ostrava, at their Ostrava site, where he studied computer science and obtained his degree in 2000. Afterward he worked for PVT a.s. Ostrava and in 2001 moved back to VSB – Technical University of Ostrava and start his doctoral studies. He obtains Ph.D. degree in 2009 and now works in VSB Technical University of Ostrava as an Assistant professor.

BARRIERS TO THE APPLICATION OF SIMULATION FOR TACTICAL DECISION MAKING

PJ Byrne
Enterprise Process Research Centre
Dublin City University Business School
Dublin City University
Ireland

Paul Young & John Geraghty
Enterprise Process Research Centre
School of Mechanical &
Manufacturing Engineering
Dublin City University
Ireland

Cathal Heavey
Enterprise Research Centre
Department of Manufacturing &
Operations Engineering
University of Limerick
Ireland

Barry Kennedy
Irish Centre for Manufacturing
Research
IR2-16 Collinstown
Co. Kildare
Ireland

KEYWORDS
Simulation, Tactical Decision Support, Industry, Multi-National Enterprises, Complex Manufacturing Systems

ABSTRACT

Simulation has been, and will continue to be a critical element in assisting competitive organisations. Historically, simulation and in particular discrete event simulation has been used predominantly for strategic decision making in both the design and operation of productive systems. Although the worth of simulation is well documented, the use of the technique as an ongoing decision support in modern manufacturing systems is not yet evident.
This paper presents a portion of the results of a larger ongoing study into manufacturing research on the island of Ireland. In particular the paper reviews the use of simulation in seven multi-national enterprises with a manufacturing base in Ireland. This study reviews the current exposure of these organisations to simulation and their difficulties in successful ongoing deployment, particularly in the realm of tactical decision making.
The study finds that there is a need for the development of a framework for sustaining the use of simulation based decision support for tactical decision making and reducing the associated cost of ownership for an organisation.

INTRODUCTION AND MOTIVATION

Simulation Modelling as a technique has been in existence for a considerable period of time, with the concepts of a discrete event simulation (DES) originating in the 1960's (Jahangirian et al., 2010). Since its inception the role and functionality of DES has continued to expand and evolve to the needs of an ever changing marketplace. This changing role and functionality has continued to pick up pace and will continue to accelerate in line with business needs over the short to medium term. Such advances in DES has been facilitated by the vast improvements in ICT in the past 10 to 15 years and the predicted advancements in the coming years.
DES is a commonly used approach in cycle time estimation and performance analysis of complex manufacturing systems. Simulations are often used to analyse systems that are too complicated to attack via analytic methods (Goldsman, 2007 and Ingalls, 2001). Simulation models can be adapted precisely to meet various experimental purposes. There are countless instances reported in the literature which relay positive reports of the use of DES. However, for the most part these reports are using simulation in the once-off "what if" mode. For examples of such see Arisha, Young and Baradie (2004) and Byrne, Kabak and Heavey (2007). The common factor across many of these papers is that DES is used predominantly for strategic decision making.
While simulation Modelling is the most practical solution in many instances it also has some inherent shortcomings (Fowler and Rose, 2004, Shannon, 1998, Chance et al., 1996). This has resulted in low adoption by industry and especially in the support of short term planning and tactical decision making. For instance Shannon (1998) notes that failures in this field can usually be traced to one of the following: goal definition, inadequate planning, inadequate user participation, code writing, too much detail, wrong mix of skills and lack of trust.
More recently, Fowler and Rose (2004) note that there is a need for the pervasive use of modelling and simulation for decision support in current and future manufacturing systems, and highlight that several challenges need to be addressed to realize this vision:

- The order of magnitude reduction in problem-solving cycles is needed to deliver solutions in a timely manner.
- The development of real-time, simulation-based problem-solving capability in order to integrate this type of analysis into everyday operations.

- The need for true plug-and-play interoperability of simulations and supporting software.
- The challenge of convincing management to sponsor modelling and simulation projects instead of, or in addition to, more commonly used manufacturing system design and improvement methods such as lean manufacturing and six sigma is called out.

In addition to these the majority of simulation projects are conducted on a desktop PC, and the simulation models used are no longer utilised once a decision is made (Hanisch, Tolujew and Schulze, 2005, Fowler and Rose, 2004, Pfeiffer, et al., 2008).

In relation to software provision, DES modelling is well resourced in terms of software providers. A small selection of the major software platforms and providers includes Arena (www.ArenaSimulation.com), Simul8 (www.Simul8.com), ExtendSim (www.ExtendSim.com), Plant Simulation (www.plm.automation.siemens.com). The challenge is not the software but a framework for using the software in a rapidly changing environment, and not only for strategic decision making but ongoing tactical decision making on a daily basis.

To progress forward, simulation as a technique can and should take that a step toward becoming a proactive decision support tool that can answer "what now" questions (Gupta, Sivakumar and Sarawgi, 2002). With this in mind, factories of the future will strain the boundaries of existing modelling methodologies ("what if" analysis carried out in one off settings). It will no longer be sufficient to use advanced modelling techniques for strategic decision making alone. Modelling will also become an imperative in day to day tactical decision making.

In summary, as described by (Gupta, Sivakumar and Sarawgi, 2002), simulation is predominantly used to describe and analyse the behaviour of productive systems, and answering "what if" questions, to assist in their design. However, simulation as a technique has been reviewed in similar contexts in the past (Hollocks, 1992, Hlupic, 1999 Kuljis and Paul 2001), but still its continued use remains in question. The following reported study addresses this question in an industrial based context.

STUDY GROUP

This work was undertaken with a group of companies as part of the Irish Centre for Manufacturing Research (ICMR) who had identified modelling and simulation as one of the tools needed to improve their operations. They are all Multi-National Enterprises (MNEs) with manufacturing facilities on the island of Ireland. The formation of the group came as a result of an initiative by the Irish government, through the development agencies Enterprise Ireland (EI) and the Industrial Development Agency (IDA), to promote industry-led and focused research (www.competencecentres.ie). Such manufacturers form a significant portion of Irelands manufacturing sector as they cover the three areas of Chemicals/Pharmaceuticals, Electrical & Electronic Equipment and Medical Devices/Instruments as outlined in Figure 1.

As such the group was self-selecting, however the drive to participate comes from the need to respond to the rapid changes to local and world economic challenges. Those partners with challenges in the area of management of the manufacturing lines and systems noticed that, in addition to meeting more exacting performance metrics, flexibility, adaptability and agility were impaired by the inability to base decisions on sound analysis and prediction. In particular, the opportunity to take a decision and have time to see whether the effect achieves the desired intent is no longer available to manufacturers and reliance on experience alone is no longer sufficient for such time-critical activities.

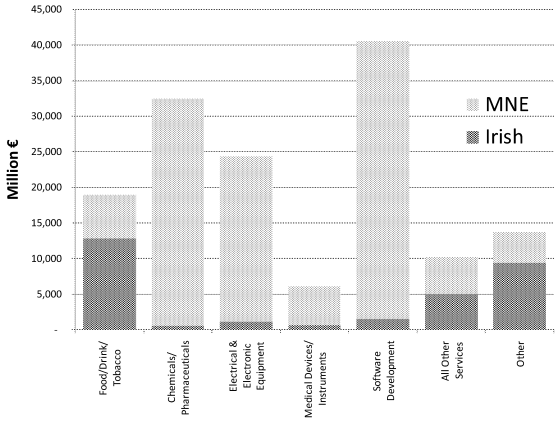


Figure 1: Total Sales of Manufacturing and Internationally Traded Services in Ireland for 2008 – Data source: Forfas ABSEI, 2008

Decisions must now be made on the basis of firm knowledge of the impact they will have on the ability to deliver on the particular demands of the day. In general, their production processes are complex, and cannot be understood and analysed based on the relatively simple examples found in most academic papers.

Those partners with challenges in the area of management of the manufacturing lines and systems noticed that, in addition to meeting more exacting performance metrics, flexibility, adaptability and agility were impaired by the inability to base decisions on sound analysis and prediction. For all of the participants the challenge is based on improving the performance of their existing systems through more intelligent use of existing resources, reconfiguration of lines and systems and (relatively) minor capital investment programmes. As with most MNE production facilities, these installations are not unique within the organization and must compete internally for investment and production as well as with external competitors. For these, the competitive advantage cannot be on the basis of the process and capabilities of the installed lines, there must be some greater efficiency, performance and productivity over sister plants in other locations. This is generally achieved through their human capital – the knowledge and understanding that their managers, engineers and technicians have of the individual machines

(loading patterns & constraints, maintenance etc.), supply systems, raw material and parts flow, line profile.

The philosophy of manufacturing changes as a manufacturing facility matures. In the start-up phase, the focus is on getting the lines up and running, getting product out the door and every effort is focused on making the system operational. Once a consistent movement of product is established, the quality of every item in production gains focus and improvements at process level to ensure that waste is reduced are key. As with the initial phase, this brings huge productivity improvements as the number of quality assured parts reaching the end of the line brings more profitability. At this stage, if the planners have made the right decisions, the market will be eager to purchase everything produced, however once the market moves on to other products, the challenge changes to that of adapting the manufacturing facility to respond to the changing demand. This is the situation facing most MNEs in Ireland. Their plants are well established and the focus is on competitiveness and ability to respond to the rapid changes in demand to ensure survival.

Each of the participants is facing different elements of this challenge. The economic downturn has led to complete cessation of some activities for some, partially completed production halted at a point where the WIP has considerable added value for others, and sudden increases in demand for those for whom their position in the supply chain mean that they feel the effects of anticipated recovery of demand for the final product.

Several Irish government reports (Forfas, 2008 and Forfas, 2010) have shown the need for this improvement in the ability to respond to demand and also to engage both academic and industry researchers to gain traction in this area, and this study represents the beginning of such activity where both have worked together to determine the best approach to delivering on the needs identified.

STUDY

The study was initiated in mid 2009 with seven MNEs from the eleven ICMR member companies actively participating. A principle aim of the study was to identify a research roadmap for IMCR for the next 3 to 5 years to enable MNEs to sustain the use of model-based decision support tools for the analysis of tactical decisions. Figure 2 broadly illustrates the research methodology adopted during the study.

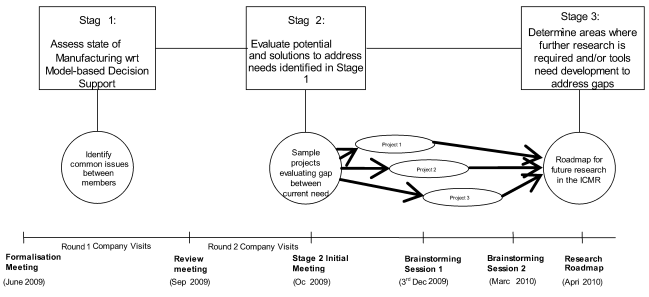


Figure 2: Research Methodology

Stage 1: Assessment of MNEs experiences

Stage 1 of the study was initiated by requesting the MNEs to complete a questionnaire to establish the baseline for the study objectives and determine the similarities and differences in experiences and approaches adopted by the MNEs with respect model-based decision support.

In order to follow-up on insights from the questionnaire a series of site visits were initiated with the MNEs to conduct interviews with key personnel from various engineering, planning and management groups within the companies. This process required several visits to each MNE and was conducted over a three month period. The manufacturing operations of MNEs involved in the research programme ranged from manually-intensive to highly automated. However, using the V,A,I,T paradigm for classifying plant structures (V – one to many, A – many to one, I – serial production, T – serial production with some customisation) it was possible to characterise the MNEs plant structures in a manner that could communicate similarities and differences between the MNEs. Three MNEs were classified as V-Plants, two as I-Plants, and one each as A and T plants.

The MNEs were at different stages of managing solutions to problems arising in the direction of tactical decision making. They had varying experience of model-based analysis and, as Figure 3 illustrates, the MNEs had a wide variety of priority areas where they felt model-based analysis could provide insights and improvements in the solutions deployed. However, one commonality was that all MNEs were highly dynamic environments with continually changing priorities. As such a key challenge for them was to adopt tools, technologies and methodologies that would not only enable them to address current (this week, month, year) questions but that would also allow them to address future questions as priorities change and business environments evolve.

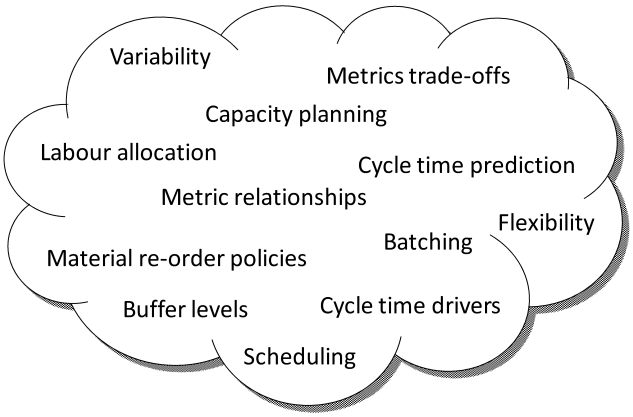


Figure 3: The Research Topics of Interest to MNEs

Stage 2: Identification of potential solutions and barriers

The second stage of the research was aimed at identifying potential tools, technologies and methodologies that could deliver on-going support for tactical decision making in such challenging and dynamic environments. The initial phase of this stage of the research was conducted through interviews and brainstorming sessions with the MNEs.

The findings of the interviews and brainstorming sessions revealed that three of the MNEs had prior experience with

deploying simulation modelling to address both strategic and tactical decision making. Additionally, three of the companies had recently invested in simulation software but had not initiated simulation projects. However, none of these companies reported the sustained use of simulation as a decision support tool for tactical decision making. The underlying factors which these industrial partners believe lead to the collapse of model-based analysis were:

- Results not implemented as potential benefit not successfully sold to person with final decision making responsibility.
- "Babysitter phenomenon" – analyst no longer available after the initial project and in-house capabilities not developed enough to continue/develop analysis further. Therefore, a period of learning/training would be required for a new analyst delaying the delivery of subsequent projects.
- Credibility of results – trade-off between level of detail and build/run time.
- Time to solution – faster turnaround required – data readiness and model build time.
- Maintenance requirement – refreshing input data and updating model structure.
- Dedication of company personnel – use and maintenance of analysis system was not included in specific job description(s).
- Feasibility – cannot implement due to complexity of problem.

The second phase of this stage of the research involved conducting case-study projects with some of the MNEs and disseminating some learnings to the other MNEs. One of the case studies involved a MNE that had recently invested in a simulation software tool. The aim was to assist them to develop their first model of one of their production lines. The feedback from this case study was that the MNE:

- Could more clearly see the potential benefits that could be realised from developing in-house simulation capabilities.
- Had a clearer understanding of the true Cost-Of-Ownership (COO) of simulation based decision support, including expertise requirements, model development, validation and experimentation time, and model maintenance requirements.
- Now realised that simulation is not a plug-and-play solution.
- Had not fully appreciated that a simulation model is only a representation of system in one state, it does not optimise the system unless run in conjunction with an optimisation engine or an experimental design.

Stage 3: Development of the research roadmap

The third stage of the research was to develop the research roadmap for ICMR for the next 3 to 5 years. This stage of the research required detailed consideration of the information collated in the previous two stages of the research in conjunction with state-of-the-art from the literature. The principle aims of stage three of the research was to:

- Develop a formal understanding of the core competencies that ICMR should seek to develop to support MNEs in Ireland,
- Identify the key research objectives for the ICMR and the specific issues of interest to the member companies, and
- Identify the potential performance metrics that could be improved within each MNE as a direct result of ICMR research activity.

Figure 4 delineates the output of this stage of the research. The breadth of MNE interest in deploying model-based decision support for tactical decision making is highlighted by the amount of grey boxes in the diagram (where each of the seven MNEs is represented by an id A through G). Central to this research roadmap, is the requirement for the development of a framework for sustaining the use of simulation based decision support for tactical decision making and reducing the associated COO for an organisation. This will require collaborative research between industry and academia adopting an embedded research approach to ensure transfer of knowledge and ownership to the MNEs.

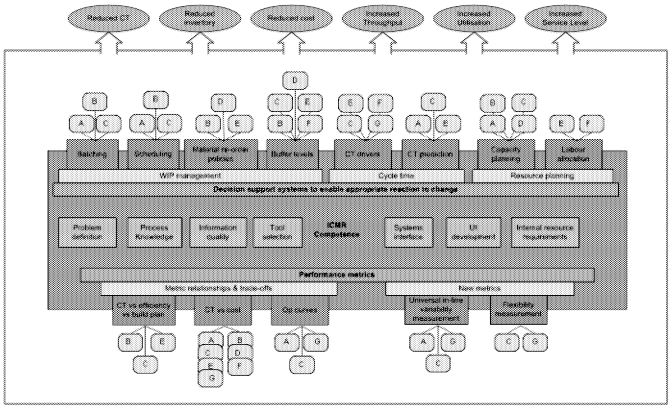


Figure 4: Research Focus for MNEs

CONCLUSIONS

Modern manufacturing is rapidly changing in Ireland as is the case in many parts of the world. Simulation is a tool that has been successfully used to harness and review such change. Traditionally simulation was primarily used for longer term strategical decision making in either existing manufacturing systems or in the design of new systems. Essentially, simulation addressed the "what if" questions. Although there is still a need to use simulation for traditional strategic purposes, there is a growing need to use simulation in the "what now" mode. That is the need to use simulation as one of the key cornerstones in tactical decision support.

This study has found that all participant MNEs feel that there is a worth to using simulation in terms of both "what if" and "what now", but in both cases challenges currently

exist. These challenges span the entire analysis process from effective requirements gathering through to selling the results.

Some solutions have been proposed to address these challenges but do so with varying levels of success. For instance, simulation model build effort is known to be considerable and automated techniques have been put forward. However, use of such automated models can not fully address the long model execution times which may severely restrict the application of DES. Therefore, model simplification methodologies can be incorporated in the automated model construction. However, there can be potential flaws with such combined approaches in terms of expertise requirements and generalisation of the solution model.

This study represents a collaborative effort between seven MNEs and Irish academics with research interests in manufacturing operations excellence to develop a roadmap for research into sustaining the use of model-based analysis for tactical decision support. All MNEs strongly believe that the survival and renewal of the Irish facilities is dependent on their ability to demonstrate capabilities in generating new efficiencies in their operations. Additionally, the MNEs believe that model-based analysis for tactical decision making would be extremely beneficial in achieving the next level of efficiencies. However, none of the MNEs could report a sustained used of simulation to this end. The study has highlighted the barriers that exist to achieving this goal and endorses the development of a framework to enable MNEs to overcome these barriers through collaborative, embedded research between industry and academia.

REFERENCES

Arisha, A., Young, P. and Baradie, M.E., 2004, A simulation model to characterize the photolithography process of a semiconductor wafer fabrication, Journal of Materials Processing Technology 155–156, pp2071–2079.

Byrne, P.J., Heavey, C. and Kabak, E, 2007, An Analysis of Tool Capabilities in the Photolithography Area of an ASIC Fab, WSC '07: Proceedings of the 39th Winter Simulation Conference, Washington, DC,, USA: IEEE Press.

Chance, F., Robinson, J. and Fowler, J.W., 1996, Supporting manufacturing with simulation: Model design, development, and deployment. In Proceedings of the 1996 Winter Simulation Conference, San Diego, CA, USA: IEEE Press.

Forfas ABSEI, 2008, Annual Business Survey of Economic Impact - 2008 Appendix, http://www.forfas.ie/media/forfas100120-annual-business-survey-economic-impact-2008-appendix.pdf, Accessed April 6th 2010.

Forfas, 2008, The Report of the High Level Group on Manufacturing, http://www.forfas.ie/media/forfas080402_manufacturing_report.pdf, Accessed April 6th 2010.

Forfas, 2010, Maximising the Environment for Company Research and Development, http://www.forfas.ie/media /100322asc-maximising-environment-for-company-r-and-d.pdf, Accessed April 06 2010.

Fowler, J.W. and Rose, O., 2004, Grand Challenges in Modelling and Simulation of Complex Manufacturing Systems. Simulation. 80 (9), pp469-476.

Goldsman, D., 2007, Introduction to simulation. WSC '07: Proceedings of the 39th Winter Simulation Conference, Washington, DC,, USA: IEEE Press.

Gupta, A.K., Sivakumar, A.I. and Sarawgi, S., 2002, Shop floor scheduling with simulation based proactive decision support. Proceedings of the 2002 Winter Simulation Conference, San Diego, CA, USA: IEEE Press.

Hanisch, A., Tolujew, J. and Schulze, T., 2005, Initialization of online simulation models. WSC '05: Proceedings of the 37th Winter Simulation Conference, Piscataway, NJ, USA: IEEE Press.

Hollocks, B.W., 1992, Simulation: a well kept secret?, OR Insight, Vol. 5 No.4, pp.12-17.

Hlupic, V., 1999, Discrete-event simulation software: what the users want, SIMULATION 73(6), pp. 362-370.

Ingalls, R.G., 2001, Introduction to simulation. WSC '01: Proceedings of the 33nd Winter Simulation Conference, Washington, DC, USA: IEEE Press.

Jahangirian, M., Eldabi, T., Naseera, A., Stergioulasa, L.K., and Young, T., 2010, Simulation in manufacturing and business: A review, European Journal of Operational Research, 203 (1), pp1-13.

Kuljis, J. and Paul, R.J., 2001, An appraisal of web based simulation: whither we wander? Journal of Simulation Practice and Theory, 9(1-2): p. 37-54,

Pfeiffer, A., Kadar, B., Monostori, L. and Karnok, D., 2008, Simulation as one of the core technologies for digital enterprises: Assessment of hybrid rescheduling methods. International Journal of Computer Integrated Manufacturing. 21 (2), pp206-214.

Shannon, R. E., 1998, Introduction to the art and science of simulation, Proceedings of 1998 Winter Simulation Conference, Washington, DC, USA, IEEE Press.

ACKNOWLEDGEMENTS

We gratefully acknowledge the efforts of the team, both academic researchers (Paul Liston and Stephen Murray) and industrial engineers who undertook this research. This work was funded under the National Development Plan (2007-2013) by IDA and Enterprise Ireland under contract CC-2008-1008.

AIDATrace - Interoperation Platform for Active Monitoring in Healthcare Environments

José Machado[1], Miguel Miranda[1], Pedro Gonçalves[1], António Abelha[1], José Neves[1] and Alberto Marques[2]
[1]Universidade do Minho, Departamento de Informática, Braga, Portugal
email: jmac,miranda,pgoncalves,abelha,jneves@di.uminho.pt
[2]Centro Hospitalar do Tâmega e Sousa, Penafiel, Portugal
email: marquestris@chts.min-saude.pt

KEYWORDS
Interoperability, Healthcare, HL7, RFID

ABSTRACT

The introduction of monitoring systems may have a great potential to introduce Ambient Intelligence based monitoring techniques in healthcare environments. On the other hand, current research being sponsored by the European Union presents interoperability issues as a considerable obstacle to implement and fully explore the capabilities of such a technology. Therefore, and in order to contribute to overcome this drawback, in this paper we address the different methodologies put into operation in the healthcare sector, supported by a putative architecture which has been used in different healthcare institutions to suport RFID monitoring systems.

Introduction

In large-scale healthcare environments, control and spatial monitoring of patients, visits, staff, service providers, medication and equipments are extremely complex and generally do not respond to the needs, nor offer information on time for decision support. These barriers create a demand for new methodologies and technologies for problem solving that have to be studied, analyzed and implemented, in order to improve Quality-of-Service (QoS) and facilitate clinician and other staff work. The unequivocal positioning and identification of an entity (person or object) in a given moment and over time can provide a base for the development of several solutions for critical problems within the healthcare system. Recent academic research and publications are concerned with Radio Frequency IDentification (RFID) in the healthcare industry, however much study must still be performed in terms of the integration of such system with existing informations systems. Recent studies directed by the European Union, defining some of the possible and undergoing applications of position identification and monitoring using RFID, state to be possible to orientate this technology towards the needs of control and optimization of medication, lab tests, equipment, patients and staff (Ngai et al. 2009) (Chen et al. 2010) The different projects mentioned in these studies demonstrate the utility of this technology, unleashing some of its potential but also its limitations. From advantages and niches of opportunities of these projects, key limitation elements such as interoperation, integration, security, ubiquity, data quality, price and overload of existing infrastructure (e.g. networks) emerge. On the other hand, the degrees of confidence and excellence are mandatory to healthcare activity, and these environments of lesser forbearance can not flaw, equipments with slower time of response and greater reliability are needed, and intelligent data processing expertise may bring decision support.

In this paper we describe and supporte a putative architecture which has been used in different healthcare institutions to support RFID based monitoring systems. The architecture core is based on a Multi-Agent (MA) and WebServices approach that uses HL7 and other proprietary syntaxes to enable interoperation among the applications in the Healthcare Information System (HIS). Indeed, in healthcare, several standards have been defined for interoperability, being Health Level Seven (HL7) the most uniformly used by healthcare information system providers (HISP).

Interoperation and integration

Interoperation and systems integration are among the major limitations generally appointed to RFID and are relevant fields of study for information systems (IS) in healthcare environments. Theses features are also among the major concerns stated in EU studies, as a secluded tracking and identification architecture limits the flow of information within the institution and threatens independence from service providers, perpetuating lock-ins. Studies using similar middleware and agent-based techniques have been previously purposed, however the devised architecture was implemented together with a interoperable platform specifically oriented towards healthcare, using its standard norms (Kim et al. 2008) (Chen et al. 2010). The flow of information generated and required by the RFID system is greatly enhanced by the existing interoperable platform, which increases the potential of these systems. Other studies,

although in quasi-real setting also mention the importance of systems integration, and find it to be one of the key difficulties in modeling a global institutional RFID architecture in an HIS (Ngai et al. 2009).
Our research group has created partnerships and is involved in research activities with three of the major Portuguese hospital centers, namely the Centro Hospitalar do Porto (CHP, EPE), the Centro Hospitalar do Tâmega e Sousa (CHTS, EPE), the Centro Hospitalar do Alto Ave (CHAA, EPE) and the Unidade Local de Saúde do Norte Alentejano (ULSNA, EPE).

AIDA Platform

The AIDA (an Agency for the Integration, Difusion and Arquive of information) platform was developed in order to support the diffusion and integration of information generated in the healthcare environment. This platform imbues many different integration paradigms, using mainly Service Oriented Architectures (SOA) and MAS to implement interoperation in a distributed, specific and standardized manner with all the service providers within an healthcare institution. Using this synergy it is maintained the independence and modularity of SOA and the intelligence and autonomy associated to MAS and Artificial Intelligence (Machado et al. 2008) (Machado et al. 2007).

Interoperability Architecture

Being MAS a field of research in Distributed Artificial Intelligence, this technology is intrinsically connected with distributed problem solving, while being distinct in the intrinsic definition of an agent versus the properties of the general middle-wares of the architecture called in its support. Indeed, under this approach a MAS subsumes a distributed architecture. The MAS is able to manage through the agent life cycle, the availability of the modules of the healthcare system and the HIS as a whole, while keeping all the agents that constitute the MAS freely distributed. In fact, new agents with the same characteristics and objectives can be created on-demand by the MAS, according to the necessities of the system they are a part of. The structuring of these agents and of the MAS can be developed according to the services they provide and the logical functionality of the systems they interoperate with, allowing for a conceptual form to take advantages of the SOA.
The SOA core has as a key principle the division of large and complex problems into simpler and modular ones. However, contrary to the common methodologies to address a great number of complex problem solving algorithms, this architecture aims to contribute to be achieved complete independence between these units. This methodology searches distinct areas of logic automation in order to unitarily make available a service that is part of a vaster and more complex service. It

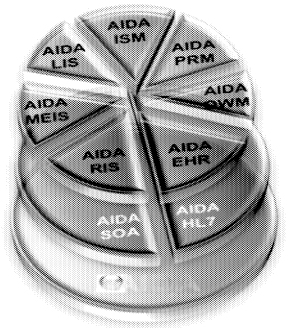


Figure 1: AIDA Architecture

is ensured that in this way each of these units can be replaced by any other unit which performs the same service without concern with other services. These base units can be distributed, providing a service in a distributed way, independently and disassociated of the underlaying global services structure (Erl 2005). Although, at first glance it may be extremely similar to the distributed architecture; the difference is in the dependency between distinct distributed units, something that is common in distributed architectures, but that goes against the ideals of a truly SOA. SOA is a concept not bound to one specific technology, it can be based on web-services, agents or any other technology that follows its basic rules. Through the SOA paradigm, a system will not be dependent of its core units. It means that services can be easily replaced and updated by others which provide the same services, enabling modularity, scalability and independence to the infrastructures that use the architecture.

As demonstrated in Figure 1 from (Marques et al. 2010), the agency's top layer of abstraction conceptually consists of 7 (seven) multi-agent based subsystems: AIDA-RIS - Radiological Information System; AIDA-MEIS - Medical Exams Information System; AIDA-LIS - Laboratories Information System; AIDA-ISM - Information System for Monitoring (e.g., vital signals monitoring); AIDA-PRM - Patient Relationship Management (including communication using SMS); AiDA-OWM - Organisation and Work Management (Including agenda, scheduling, planning and resource management; and AIDA-EHR- Electronic Health Records.

AIDA's devised architecture supports intelligent agents that acting as distributed entities on a healthcare environment, gather all the data, transform the underlying information, correct information incoherence and disseminate it thought-out the HIS. The introduction of data validation allows to improve the overall quality of information extracted and avoid the spread of inconsistencies over all involved systems.The quality of the gathered information is important to guarantee that the decisions made in these environments is based on sound principles and are not led astray by incoherent or inex-

istent information.
This platform was engineered under the perspective of a centralized repository for all the significant data in a healthcare institution. Under this happening, the underlying information must be retrieved from and disseminated towards the different service providers in the healthcare institution. Henceforth, the gathered information must be processed and corrected, validating the execution of existing workflows and ensuring the Quality-of-Information (QoI) disseminated to other systems. The presence of a global information and knowledge repository that is oriented towards the whole of the healthcare institution, that has the extensibility to adapt to the heterogeneity within its environment, is an important tool for information validation and knowledge discovery. For this purpose the AIDA platform uses an ORACLE RAC database, structuring most of its clinical and management information in highly compact but well structured XML syntax. The use of XML grants the necessary malleability to adapt the repository configuration to the needs of a specific institution, service or external provider. Clinical reports and other information gathered by the platform are therefore structured and processed in this repository and validated against existing knowledge or information.
An intelligent behaviour is necessary to predict and validate, either semantically or syntactically, the data and information gathered according to the knowledge representation and reasoning techniques used in each middleware agent responsible for systems interoperability. Each of the agents is embedded with the particular behaviours from one another in order to guarantee a good response of a particular agent having in mind the particularities of the service provided. The scalability and modularity of this architecture is essential not only to the selection of new solutions but specially when developing Decision Support Systems (DSS). The multitude and intricacy of services that must be performed by DSS or Group Decision Support (GDSS), require such a platform or otherwise would be inefficiently intertwined with other essential solutions such as the Electronic Health Record (EHR).

Relevance of the Interoperability Platform

Considering the definition of an HIS, AIDA is crucial in the implementation and functioning of an HIS. Its capabilities allow for communication among different and otherwise secluded systems, avoiding centralization and dissemination of end-to-end connections, that may restrict the growth of the infrastructure associated to the HIS. The non-modularity of services adds complexity to updates and improvements extremely complex, increasing the global costs of the information systems (Aier and Schönherr 2006). Therefore, it is understandable the present concern demonstrated by organs and institutions, responsible for financing and regulating the purchase and development projects for new HIS, requiring flexibility, interoperation and integration of heterogeneous systems (Berg 2004). Furthermore, the existence of such a platform enables one to concentrate efforts in systems integration, making needless the everlasting arduous complication of interoperation processes among different providers and the complexity of adding a new one. A global knowledge repository such as AIDA, also provides independence for the healthcare institution relatively to external service providers and legacy applications.

Interoperation between RFID and other HIS Services

Within an HIS, several services must be fully prepared for interoperation with the RFID system, one of the cornerstone of implementing such a system and one of the main current concern of healthcare institution envisioning such a step. The flow of information must be consistent within the existing services. Just to name a few services, one may mention human resources, patient management system, electronic health record and radiological information systems. The AIDA platform already disseminates information among all these services according to previously defined workflows, inherent to the policies of a general healthcare institution and the institution specificities. Therefore, the main concern of integrating an RFID system into an HIS is the communication of this system with AIDA, as the platform already ensures interoperation. The implemented architecture (Figure 2) is centered on key events of the RFID system, which are received, transformed and processed by the AIDA platform. The events are generated when a state is updated or comes from others services of the HIS. To understand the events one must mention the underlying abstract concepts of the RFID implementation.
In this RFID system the main functional units are zones, profiles, tasks and alarms. The areas of the hospital are divided into zones, which are linked to the profile of the patient, staff and visits. The access of a patient to one or more zones is delimited by a task, in which entities and a time frame are worked out. The movement detections between zones of tags associated to profiles generates alarms which may be of different types, depending of certain factors, namely the permission to enter a particular zone.
According to this concept, the events can be divided into demographic communication and active monitoring. Demographic communication events are concerned with the insertion, update and removal of profile information and tag association: Admission, update and discharge of patient profile; Admission, update and removal of equipment profile; Admission, update and removal of staff profile; and Association and dissociation of a tag to a profile. The active monitoring events are centered

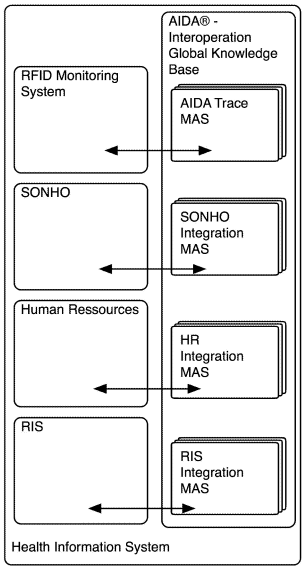


Figure 2: Interoperation Architecture

on the detection events and the association of tasks to active profiles: Managing alarm types; Detection of an alarm event; and Insertion, update and removal of a task. Detectors have been placed in each zone where patients, staff and visits have necessarily to cross when changing zones, as well as in some central areas and crossing points to control their presence. This procedure is adequate in some specific services (e.g. admitted patients and emergency care), but particular administrative procedures have to be set in other ones due to a high number of incoming users (some thousand of RFID tags delivered each day).

Although the detection events and admissions (patient and profile) occur only in one direction, i.e. only the monitoring system detects and sends those events, the other events are bi-directional. As an example of an unidirectional event, the admission of a patient can be performed by the administrative staff using the patient management system (in most of portuguese healthcare intuitions it is used the SONHO application made available by the ACSS) and the AIDA platform would disseminate this information among all defined services for this event, including the monitoring system. In opposition a task associated to an existing patient can be created, altered or deleted in both monitoring and the AIDA platform. The integration architecture uses MAS that communicates using HL7 messages. If the monitoring provider does not make available an HL7 system, the AIDA platform can use specialized agents to transform events into HL7 messages that fit into the according interoperation semantics. The use of HL7 is essential for interoperability given the relevance and dissemination of this standard in the healthcare industry. In fact this is one of the key points of this implementation, as the transformation of such events in HL7 messages grants the modularity and the generalization of the integration processes. Although some projects such as from (Chen et al. 2010), (Kim et al. 2008) or (Ngai et al. 2009) consider the problem of interoperability, the potential and complications of this process are not yet well defined, and that is where the existing HIS exchange standards as HL7 must be considered.

The modeling of these events in HL7 messages comes naturally, considering the different type of messages for admissions, transfers and other regular processes in an healthcare institution. Moreover, to model and exchange it in any other proprietary syntax and semantic would be a considerable complication if it was not the effort towards the communication process. Upon receiving an event message, the MAS does not register the data, but it extracts the information facing the existing information of the overall HIS. According to this information, one may infer the existence of inconsistency and proactively correct and unify the available information within the HIS. In fact, such situations are common in a naturally complex and heterogeneous environment such as the healthcare one, where information inconsistencies are an active concern of clinicians and administrative staff. The use of such behaviours in the integration agent system improves the results of the RFID architecture, as it enables procedures to correct misreadings or unexpected/error events that may occur in any complex system.

The introduction of intelligent behaviours on information systems in healthcare may, however, be a complex theme for legal and ethical discussion. As defended in (Machado et al. 2009), the requirements of software applications for the healthcare arena, although being rather similar to those of other areas, develops in a completely different dimension due to the value inherent to the legal abd moral good, i.e. the health condition of a human being. Considering the introduction of a learning intelligent MAS in healthcare, within this technology different intelligent agents, autonomously and adaptively, defend individually or by means of cooperation their interests and objectives. They concentrate vital functions of the healthcare unit, improving the overall quality of service. These agents are autonomous from each other, having the capacity to interact with its environment and evolve, acquiring new methodologies and information to improve their own qualities and competence, i.e. to solve different problems according to its duties. However, the legal and ethical complexity inherent to most medical decisions carries doubts considering the capability of these agents to act and support the decision making of contexts which carry a great legal and ethical dilemma (Andrade et al. 2004) (Machado et al. 2009). Although the use of intelligent agents for integration of systems may not seem to hold a great deal of ethical significance as these tools improve the security and functionality of the medical information as a whole, the consequences of the loss or adulteration of clinical

information or the permissiveness towards this sort of actions, carries a unmeasurable ethical and moral value. Henceforth, in the case of RFID systems, not only the moral nuances of an individuals right to not be monitored against it's will and if monitored what to do with this information are to important to take under consideration. The legal and ethical limitations and complications associated to the introduction of intelligent agents in both interoperability and decision support based on these RFID systems must be evaluated in order guarantee a proper control over the exiting systems.

Conclusion

The AIDA platform is an example of an application based on AI techniques, namely MAS, to overcome key limitations in the healthcare environment. In this paper, we present an architecture and methodologies for implementing interoperation between ambient monitoring systems and the HIS in a scalable, manageable and productive manner in healthcare. The implementation of SOA using MAS enables HL7 communication using modules and ensuring compliance to the defined specification, even when the monitoring service provider fails due to technological limitations. Furthermore, the scalability of the system enables rearrangements within the MAS parallelization tasks. The introduction of intelligent behaviours in data processing through the modeling of inconsistencies and incompleteness in the existing information available within all solutions that are part of the HIS, has the potential to improve the results of the RFID system itself. Moreover, it can improve the overall quality of information within the HIS, using the information exchange between systems to validate theirs completeness and detect flaws in workflows and exceptional cases. The development of an RFID monitoring systems has been performed by different other systems such as those developed by (Chen et al. 2010), (Kim et al. 2008) or (Ngai et al. 2009), however the insights in the devised interoperability architecture presents another perspective to improve the behaviour and effectiveness of this genre of tool. Furthermore it strengthens the usefulness of RFID systems by increasing the amount of information that can be extracted from the data events they generate.
As future work, while the system is in a production phase, AI techniques such as evolutionary algorithms, artificial neural networks or decision-trees could be further studied to improve the behaviour of data validation, information consistency and knowledge extraction within the system. The use of interoperation platforms such as AIDA addresses and resolves many of the concerns on existing studies required by the European Union. However future research must be held regarding the improvement of exiting mis-detections and false alarms developed by these monitoring systems, which increase the entropy and error within the HIS.

REFERENCES

Aier S. and Schönherr M., 2006. *Evaluating Integration Architectures –A Scenario-Based Evaluation of Integration Technologies. Trends in Enterprise Application Architecture*, 2–14.

Andrade F.; Neves J.; Novais P.; and Machado J., 2004. *Software Agents As Legal Persons.* In *Virtual Enterprises and Collaborative Networks.* 123–132.

Berg M., 2004. *Health Information Management: Integrating Information Technology in Health Care Work.* Routledge.

Chen M.; Gonzalez S.; Leung V.; Zhang Q.; and Li M., 2010. *A 2G-RFID-BASED E-HEALTHCARE SYSTEM. IEEE WIRELESS COMMUNICATIONS*, 17, no. 1, 37–43.

Erl T., 2005. *Service-Oriented Architecture: Concepts, Technology, and Design.* Prentice Hall PTR.

Kim Y.B.; Kim M.; and Lee Y.J., 2008. *COSMOS: a middleware platform for sensor networks and a u-healthcare service.* In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing.* ACM, 512–513.

Machado J.; Abelha A.; Novais P.; and Neves J., 2008. *Quality of Service in Healthcare Units. European Simulation and Modelling Conference 2008*, 291–298. Bertelle, C Ayesh, A European Simulation and Modelling Conference OCT 27-29, 2008 European Technol Inst, Havre, FRANCE.

Machado J.; Alves V.; Abelha A.; and Neves J., 2007. *Ambient intelligence via multiagent systems in the medical arena. ENGINEERING INTELLIGENT SYSTEMS FOR ELECTRICAL ENGINEERING AND COMMUNICATIONS*, 15, no. 3, 151–157. ISSN 1472-8915.

Machado J.; Miranda M.F.M.; Abelha A.; Neves J.; and Neves J., 2009. *Modeling Medical Ethics through Intelligent Agents.* In C. Godart; N. Gronau; S.K. Sharma; and G. Canals (Eds.), *I3E.* Springer, *IFIP*, vol. 305. ISBN 978-3-642-04279-9, 112–122.

Marques A.; Correia A.; Cerqueira L.; Machado J.; and Neves J., 2010. *Archetype-based Semantic Interoperability in Healthcare.* In J. Filipe; A.L.N. Fred; and B. Sharp (Eds.), *ICAART (2).* INSTICC Press. ISBN 978-989-674-022-1, 305–308.

Ngai E.; Poon J.; Suk F.; and Ng C., 2009. *Design of an RFID-based Healthcare Management System using an Information System Design Theory. Information Systems Frontiers*, 11, no. 4, 405–417.

MANUFACTURING PROCESS SIMULATION

A COMPARISON OF MODIFIED JUST-IN-TIME PRODUCTION SYSTEM WITH CONWIP

Ali Ardalan
College of Business and Public Administration
aardalan@odu.edu
Rafael Diaz
Virginia Modeling, Analysis, and Simulation Center
Old Dominion University
Norfolk, Virginia, 23529 USA
rdiaz@odu.edu

KEYWORDS
Production, Discrete simulation, Monte Carlo

ABSTRACT

This paper compares performance of a modified kanban-controlled JIT system with Constant Work-In-Process system. The conventional JIT system is modified by using the information about customer waiting lines. Results of simulation experiments show that the modified JIT system performs better than CONWIP with respect to customer wait-time and inventory when system utilization is high.

INTRODUCTION

The late 70's and early 80's witnessed the rapid spread of successful implementations of Just-In-Time (JIT) concepts in mass production. Since then practitioners and researchers have been trying to deploy these concepts in other types of production system. One of the major successful approaches for applying JIT concepts in flowshops is Constant Work-In-Process (CONWIP) that was reported by Spearman and Zazanis (1988), and Spearman, et al. (1990). They also discuss the superiority of CONWIP over JIT in flowshops that have an unstable master production schedule and a bottleneck work-station.

The last three decades have also witnessed significant advancements in information and production technologies and creative approaches in management of operations. These improvements provide grounds for deployment of JIT concepts in non-repetitive production environments. One such application makes information about the status of customer orders available to each work-station in a flowshop. Operators use this information and the information provided by kanbans to determine priority of each job in their work-centers. The purpose of this paper is to use simulation to compare the performance of CONWIP with this modified JIT system in a flowshop. Results show that the modified JIT system outperforms CONWIP with respect to both customer wait-time and inventory. In addition, this paper

BACKGROUND AND REVIEW OF LITERATURE

Just-In-Time

Each work-center in a conventional JIT system has an input stock-point to hold raw materials and an output stock-point for finished components of that work-center. The dual-kanban JIT system has two types of kanban in each work-center. One, a production kanban always remains in the work-center, and its presence on the kanban post signals the need for production of the item. The other type is withdrawal kanban. A work-center in need of raw materials moves a withdrawal kanban to the output stock-point of its upstream work-center to withdraw finished components from that center to replenish its input stock-point inventory. A more detailed discussion of the operation of just-in-time systems is presented by several authors (Karmarkar 1986; Kimura and Terada 1981; Lee 1987; Monden 1983; Sarkar and Fitzsimmons 1989; Schonberger 1982; and Spearman and Zazanis 1992).

The very limited number of published reports on the use of JIT in non-repetitive environments discuss ways of modifying JIT operations to better suit the special characteristics of these production environments. Handfield (1992) presents a case study of JIT in discrete batch processes. Ardalan (1997) and Diaz and Ardalan

(2010) compare performance of priority rules in JIT systems with kanban in intermittent production systems. Yavuz and Tufekci (2006) discuss application of JIT in non-synchronized assembly lines. Production systems, including JIT, that are deployed for mass customization may also be used to produce goods in small batches (Aigbedo 2007). Build-to-order production systems that use JIT are ideal for improving volume flexibility, configuration, and complexity in build-to-order supply chains (Christensen et al. 2005). Abdul-Nour et al. (1998) report on the use of JIT concepts with kanban in a make-to-order system where product demand is very unpredictable and uncertain. Kesen and Baykoc (2007) study automated guided vehicles in a JIT job-shop. Handfield (2003) studies the effectiveness of JIT in non-repetitive processes and shows that average set-up time per operation and average lot size are significantly smaller when JIT is used to produce goods. He confirms that production of small lots can be economical in JIT controlled non-repetitive production systems. These studies show that although JIT is being used in such environments, relatively few published articles deal with the application of JIT in intermittent systems.

One major difference between mass production and non-repetitive processes is the level of stability of their master production schedules. In mass production, the master schedule includes a small number of products with highly stable demand for each. Once the system is setup for production, it makes the product for a relatively long time. Determining production priority at each stage of the process is relatively simple and stable because of the stability of the master schedule, and the relatively long production runs. In this environment, work-center operators use the simple signals provided by kanbans or cards, or other types of communication media to control production of items in their center. Operators also use this information for controlling the flow of products between their center and their succeeding and preceding work-centers. However, in some flowshops, the master production schedule includes a relatively large number of products with variable demand for each. In addition, the master schedule is updated frequently to reflect changes in demand. Production runs are relatively short, and production priorities change frequently. In their current form, kanbans cannot provide the information about production priority that is needed for effective operation of flowshops.

The modified JIT uses the flexibility that has become possible by improvements in information and production technologies and innovative approaches in operations planning and control to be considered as a strong contender for management of operations in flowshops.

Constant Work-In-Process (CONWIP)

CONWIP is a push/pull system that aims at achieving JIT's inventory control in flowshops by starting production of a new unit only when a unit is made. A production kanban is released to the starting station in the flowshop only when a unit is completed exits the system. The push system is used to move units between stations as they are being made. No kanbans are used to coordinate the movement of units within and between stations. Hence, while the total number of units in the system is held constant, the number of jobs in each station may vary.

Spearman and Zazanis (1988) and Spearman et al. (1990) introduced CONWIP as a production system that preserved some of the strengths of JIT, yet could deal with several operational features that the conventional JIT could not address well. These features include:

long set-up times
production of a relatively large number of products in small lots
unpredictable variations in demand for each product, and
presence of a bottleneck station.

They compared Kanban with CONWIP to demonstrate that Kanban is applicable to a limited number of production environments while CONWIP can be applied to more types. Also, Spearman et al. (1990) discuss similarities of CONWIP and Drum-Buffer-Rope (Goldratt and Fox 1986) when there is a bottleneck station present in the flowshop.

Spearman et al. (1989) present a detailed description of a hierarchical control system for CONWIP production systems. They indicate that CONWIP has the advantages of JIT systems and in addition it accommodates changing product mix, can be used when setups are longer and production runs are shorter, and is perdictable. They also state that proper operation of CONWIP production system requires establishing a production quota and the number of kanbans. However, they do not show how to determine these policy variables.

Duenyas et al. (1993) argue that most often processing times do not follow exponential distribution. They characterize the output process of a CONWIP system in a one station, one line product production unit when processing time is deterministic, but machine failure and repair times follow exponential distribution. They derive the optimum production quota and number of kanbans for this system. In addition, they use simulation to determine these policy variables and show that the simulation results are very close to the optimum values of these variables.

Yang (2000) compares the performance of a single kanban, dual kanban, and CONWIP system. He shows that although the differences in performance of these systems with respect to customer wait-time and inventory are not statistically different, both customer wait-time and inventory are smaller for CONWIP than the two kanban systems.

While the aforementioned studies conclude the superiority of CONWIP over JIT, some reports suggest similar performance for JIT and CONWIP. Gstettner and Kuhn (1996) compare the performance of JIT and CONWIP and indicate that JIT is more flexible than CONWIP as it allows for variable distribution of cards to work-centers. In the optimal condition for saturated line (infinite demand) the optimal JIT performs better than CONWIP with respect to WIP. Geraghty and Heavey (2004) define several types of Hybrid Push/Pull and CONWIP/Pull systems. They compare the performance of a Hybrid Push/Pull system with CONWIP/Pull and show that under conditions of optimum inventory and safety stock the systems are equivalent with respect to WIP and gain – total inventory holding and shortage cost.

This study compares the performance of the modified JIT with CONWIP in a flowshop and shows that the modified JIT performs better with respect to both customer wait-time and total inventory.

METHODOLOGY

The Flowshops

Two flowshops were simulated in this study. Both were comprised of four stations that made four products. One of the flowshops had a well balanced station processing times. The other flowshop had a bottleneck station. The bottleneck station was the third station in the process. The bottleneck station was included to test the performance of CONWIP and modified JIT in a flowshop with bottleneck. The products had equal demand. Demand interarrival time followed exponential distribution. In the balanced model the difference in processing times for the four products in each station and among the four stations was very small. In the other model the bottleneck station's processing time was about ten percent larger than processing time in other stations. Processing times followed exponential distribution.

The Experiments

The major policy variables in CONWIP and modified JIT are different. Production quota and the number of kanbans are the major policy variables in CONWIP. Number of kanbans is the main policy variable in modified JIT. While it was possible to compare the effect of a variety of levels of CONWIP and modified JIT's policy variables on the operations of the two flowshops, we compared their effect at the same utilization levels of the flowshops. We believe that this will provide a more accurate comparison of the CONWIP and modified JIT. Three utilization levels of 80, 85, and 90 percent were used. The flowshops were saturated at all of these levels – there were almost always customers in line for at least one of the four products.

Performance Measures

The measures of performance were total inventory and customer wait-time. These two measures are usually used to provide information about of cost of customer service. Both CONWIP and JIT have been very successful in reducing both of these measures simultaneously.

Simulations

The Arena simulation software was used to simulate the two models. Both simulation models were validated by manually tracing the operations of the production systems. Also, the interarrival time and processing times were tested extensively to ensure that all waiting lines in the model reached steady state conditions.

The study used a total of 6 treatments – three levels of flowshop utilization and two levels production control – CONWIP and modified JIT. Each treatment was simulated for 10,000 hours for each of the 50 replications to provide a higher than 99 percent confidence level in the results. In addition, the same set of random numbers was used for simulation of all of the treatments to minimize any variations caused by factors external to the production system.

RESULTS

Full factorial ANOVA was used to analyze the effect of CONWIP and modified JIT on total inventory and customer wait-time in the two flowshops.

Balanced Flowshop

ANOVA results in the balanced flowshop showed statistically significant differences in total inventory for CONWIP and modified JIT ($p < 0.01$) at all three utilization levels. Modified JIT had smaller inventory than CONWIP at all three levels, Table 1.

Table 1: Inventory for the Balanced Flowshop

System Utilization %	CONWIP	Modified JIT
90	10.75	4.47
85	7.82	2.13
80	6.98	1.58

However, there was a statistically significant difference in customer wait-time ($p < 0.01$) between CONWIP and modified JIT at 90 percent utilization level. Modified JIT had a smaller customer wait-time than CONWIP, Table 2.

Table 2: Customer Wait-time for Balanced Shop

System Utilization %	CONWIP	Modified JIT
90	188.72	134.71

Flowshop with Bottleneck

ANOVA results for data in this flowshop showed statistically significant differences in total inventory between CONWIP and modified JIT ($p < 0.01$) at all three levels of system utilization. Modified JIT had a smaller total inventory at all three levels, Table 3.

Table 3: Inventory for the Bottleneck Flowshop

System Utilization %	CONWIP	Modified JIT
90	11.76	5.27
85	8.11	2.33
80	7.15	1.58

ANOVA results showed a statistically significant difference in customer wait-time ($p < 0.01$) between the two systems. Modified JIT had a smaller customer wait-time than CONWIP, Table 4.

Table 4: Customer Wait-time for Bottleneck shop

System Utilization %	CONWIP	Modified JIT
90	251.82	194.71

CONCLUSIONS

Since its inception, JIT concepts have been expanded to make them applicable to non-repetitive production environments. This study compared a modification of JIT that includes information about customer line in sequencing jobs in each station with CONWIP. Because CONWIP was developed to deal with features of a production system that did not lend them to application of JIT such as bottlenecks, we used two systems - one without a bottleneck and one that included a bottleneck.

Statistical results of simulation experiments showed that modified JIT resulted in higher performance of both types of flowshops with respect to the two measures of performance.

Another important concept of JIT is simplification. The modified JIT is very easy to implement. However, implementing CONWIP requires proper values for production quota and the number of kanbans that are relatively difficult to determine. As such, the modified JIT is superior to CONWIP in the flowshops considered both in performance and ease of application. However, the results of this study may not be generalized and more research is required to better understand the characteristics of the two systems.

REFERENCES

Abdul-Nour, G., S. Lambert, J. Drolet. 1998. Adaptation of JIT philosophy and Kanban technique to a small-sized manufacturing firm; A project management approach. *Computers and Industrial Engineering* 35(3-4) 419-422.

Aigbedo, H. 2007. An assessment of the effect of mass customization on suppliers' inventory levels in a JIT supply chain. *European Journal of Operational Research* 181(2) 704-715.

Ardalan, A. 1997. Analysis of local decision rules in a dual-kanban flow shop. *Decision Sciences* 28(1) 195-211.

Christensen, W.J., Germain, R., Birou, L. 2005. Build-to-order and just-in-time as predictors of applied supply chain knowledge and market performance. *Journal of Operations Management* 23(5) 470-481.

Diaz, R., A. Ardalan. 2010. An analysis of dual-kanban just-in-time systems in a non-repetitive environment.

Production and Operations Management. 19(2), pp. 233-245.

Duenyas, I., Hopp, W.J., and Spearman, M.L., 1993, Characterizing the output process of a CONWIP line with deterministic processing and random outages, *Management Science*, 39(8), pp. 975-988.

Geraghty, J., and Heavey, C., 2004, A comparison of Hybrid Push/Pull and CONWIP/Pull production inventory control policies, *International Journal of Production Economics*, 91, pp. 75-90.

Goldratt, E.M., and Fox, R.E., 1986, The Race, New York, North River Press, Croton-on-Hudson.

Gstettner, S., and Kuhn, H., 1996, Analysis of production control system kanban and CONWIP, *International Journal of Production Research*, 34(11), pp. 3253-3273.

Handfield, R. 1992. Amadas industries: JIT in a discrete batch manufacturing plant. *Unpublished case study,* Michigan State University.

Handfield, R. 2003. Distinguishing features of just-in-time systems in the make-to-order/assemble-to-order environment. *Decision Sciences* 24(3) 581-602.

Karmarkar, U.S. 1986. *Push, pull and hybrid control systems*. Working paper series no. QM8614, Graduate School of Business Administration, University of Rochester.

Kesen, S. E., Baykoc, O.F. 2007. Simulation of automated guided vehicle (AGV) systems based on just-in-time philosophy in a job-shop environment. *Simulation Modeling Practice and Theory* 15(3) 272-284.

Kimura, O., H. Terada. 1981. Design and analysis of pull system: A method of multistage production control. *International Journal of Production Research* 19(3) 241-253.

Lee, L.C. 1987. Parametric appraisal of the JIT system. *International Journal of Production Research.* 25(10) 1415-1429.

Monden, Y. 1983. *Toyota production system: Practical approach to production management*. Atlanta, GA: Industrial Engineering and Management Press.

Sarkar, B.R., J.A. Fitzsimmons. 1989. The performance of push and pull systems: A simulation and comparative study. *International Journal of Production Research*, 27(10) 1715-1731.

Schonberger, R.J. 1982. Japanese manufacturing techniques. New York: Free Press.

Spearman, M.L., and Zazanis, M.A., 1988, Push and pull production systems: Issues and comparisons, Technical Report, 88-24, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL.

Spearman, M.L., Hopp, W.J., and Woodruff, D.L., 1989, A hierarchical control architecture for Constant Work in Process (CONWIP) production systems, *Journal of Manufacturing and Operations Management*, 2, pp. 147-171.

Spearman, M.L., M.A. Zazanis. 1992. Push and pull production systems: Issues and comparisons. *Operations Research* 40(3) 521-532.

Yang, K.K., 2000, Managing a flow line with single-Kanban, Dual-Kanban or CONWIP, *Production and Operations Management*, 9(4), pp. 349-366.

Yavuz, M., Tufekci, S. 2006. A bounded dynamic programming solution to the batching problem in mixed-model just-in-time manufacturing systems. *International Journal of Production Economics* 103(2), 841-862.

BPM METHOD EXTENSION FOR AUTOMATIC PROCESS SIMULATION

Štěpán Kuchař
Jan Kožusznik
Department of Computer Science
VŠB – Technical University of Ostrava
17. listopadu 15/2172
70833 Ostrava-Poruba, Czech Republic
E-mail: stepan.kuchar@vsb.cz, jan.kozusznik@vsb.cz

KEYWORDS

BPM method, business processes, business process modeling, business process simulation, Petri nets

ABSTRACT

This paper extends the BPM (Business Process Modeling) method with the ability to perform simulation experiments on business processes described by this method. This is accomplished by introducing probability distributions to the activity coordination model. These distributions describe variable aspects of the process like choosing activity scenarios, defining execution and waiting times of individual activities and continual entry of customers to the process. This final behavioral model of the process is then converted to a timed Petri net with variable transition times that is used for performing the simulation.

INTRODUCTION

Ideas of business processes and business modeling are very up-to-date recently and not only in the field of computer science. Business modeling doesn't take part only in the software process but it is also used for analysis, verification and planning of activities that are performed by all employees in the company. A description of business processes in the company can bring advantages to the company's management to plan available resources more effectively or to optimize steps necessary to fulfill all requirements from company's customers. This description can also reveal possible problems or bottlenecks in the process.

A method for verification and simulation of processes is crucial for such analysis. Verification can be done by modeling methods based on formal mathematic approaches such as Petri Nets, State charts or Pi-calculus. On the other hand, the simulation isn't possible without simulation tools created exclusively for the applied method and without additional parameters that aren't important for the basic description of processes.

BPM Method (Business Process Modeling) created by Prof. Ivo Vondrák (see chapter BPM Method, for more information see (Vondrák et al. 1999a), (Vondrák et al. 1999b), (Vondrák et al. 2006), (Vondrák 2002)) is such a method for modeling business processes that is supported by a software tool called BP Studio (see (Vondrák 2000)). This modeling tool also provides means for manual simulation that can be used to manual validation of the process and can help a user to understand to the process more easily. But manual simulation isn't able to answer questions about performance parameters of the process with limited resources for multiple concurrent customers. This article strives to extend the BPM Method with parameters and procedures to allow automatic simulation of processes based on given stochastic behavior of the process.

BPM METHOD

BPM method is a visual tool for business process modeling based on the Petri Nets formalism (see (Peterson 1981)). BPM makes it possible to create models that describe various aspects of processes like their structure, functions, services and their dynamic behavior. Together with the formal approach and simulation tools the BPM Method is a strong alternative for modeling and verification of processes.

BPM offers three basic views on the business process – Functional View, Object View and Coordination View.

Functional View shows the architecture of the process and his inputs and outputs. Main subprocesses and services, their relations, customers, products and responsible workers are described by this view.

Object view describes static structure of the process. Its main purpose is to depict all important workers and artifacts and their mutual relations and interactions.

Coordination view is the last but the most important model of the BPM Method. It specifies the behavior of the process as a sequence of activities, what resources the activities demand and which artifacts are produced. Alternative flow and concurrency is also enabled by specific modeling elements. This is the view that is based on the Petri Nets formalism and is directed by its rules.

Because the coordination view is the most important view for the extension proposed in this paper, an example is shown on the Figure 1. This example shows a simple process of hiring new employee in the human resources agency based on my master's thesis (see (Kuchař 2007)).

The process in the example starts with a Demand that is based on the company's demand for new employees. HR Worker takes the Demand and searches for suitable applicants. The HR Worker then contacts the chosen Applicant and the Applicant chooses if he is interested in the job. If he isn't interested the process ends without further activities, but if the Applicant is interested he then

sends his CV to the agency. HR Consultant from the agency then arranges an interview with the Applicant and the interview takes place in the agency. The final product of the process is the interview report made by the HR Consultant.

Another possible scenario of the process is when the Applicant contacts the agency on his own initiative when he tries to find a job himself. When this happens, the process starts with already interested Applicant that sends the CV to the agency. Then the interview is arranged and performed in the same way as in the first described process scenario.

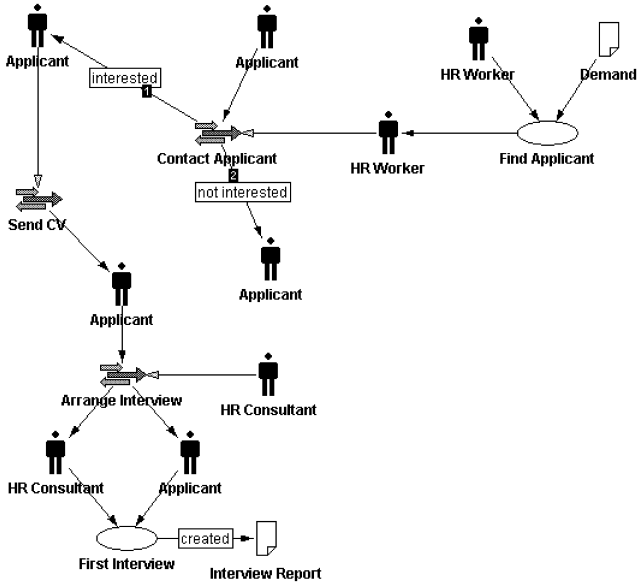


Figure 1: Coordination View

Subprocesses *Finding new applicant* and *First interview* shown on the diagram should be specified by other diagrams but let's replace them with simple activities in this example. This simplification won't influence the proposed extension because any complex subprocesses could be processed as parts of the final simulated process. When the simulation encounters any subprocesses, it will insert the subprocess definition to the standard process flow and create one huge Petri Net.

STOCHASTIC EXTENSION OF THE BPM METHOD

The main purpose of this paper isn't the description of different views of the BPM Method, but to extend these views with parameters and functions needed for automatic simulation of processes. This will be ensured by introducing several stochastic aspects to the models that could be used to describe the performance behavior of processes in the reality, so that weak spots and bottlenecks in the process can be detected.

According to the definition of the BPM Method in prior chapters, it is clear that the whole behavior of the process can be described by the Coordination View. And so all proposed extensions will be introduced to this view, mainly to individual activities in the model - the duration of activities can be varying based on external and internal factors, some activities have to wait for external events or services and some activities can have alternative scenarios that have to be performed with given probability. Entry of new customers to the process has to be introduced to the models for analyzing performance statistics in addition to the single activities' stochastic aspects.

The extended coordination model will then be converted to the underlying Petri Net and the stochastic extension will be encoded into the stochastically timed transitions. The final result will therefore be a Generally Distributed Timed Transitions Stochastic Petri Net (see (Wang 1998), (Rozenberg and Reisig 1998)).

VARYING DURATION OF ACTIVITIES

Actual version of the BPM Method is able to define specific duration for every activity, but in real processes the activities' duration depends on many factors, both external and internal. Therefore it is useful to define the duration as an interval in which the most probable durations lie. Interval definition is also more easily specified by managers (who will have to specify these parameters because they have the best understanding of analyzed processes) than standard probabilistic distribution parameters.

Stochastic properties

A normal distribution is probably the most relevant probability distribution to be used to define varying duration of activities because the most probable values are situated around the mean value (see (Longnecker 2008)). It is also useful to place the specified interval of possible values around the mean value so that the mean value will lie in the center of the interval.

If the duration interval is defined as (*min*, *max*) then the mean value μ of the distribution $X \rightarrow N(\mu,\sigma^2)$ will be evaluated as

$$\mu = (min + max) / 2$$

The variance σ^2 of this distribution will be a bit more difficult to evaluate because one of the properties of normal distribution is its non-zero probability of every value. Even though risk situations can prolong the duration of activity out of the specified interval, the most probable values should always lie in the interval. And this requirement will help us to find out the ideal variance.

Variance σ^2 is the square of the standard deviation σ. According to the commonly used and known Empirical Rule (for more information see e.g.(Longnecker 2008)), 99.7% of the values from the normal distribution lie in the interval

$$(\mu-3\sigma, \mu+3\sigma)$$

And so the variance could be evaluated as

$$\sigma^2 = ((\mu - min) / k)^2 \qquad \text{where } k = 3.$$

Chosen constant $k = 3$ actually corresponds with the amount of risk that the duration will be situated out of the specified interval. Therefore if the constant is smaller then the probability that the evaluated duration will be out of the interval is greater and vice versa. Value $k = 3$ is a standard value that covers the greatest portion of the interval, but

other value could be chosen for each activity to better describe the appropriate risk for prolonging the duration.

The resulting duration value can't be allowed to be negative, but normal distribution can return negative numbers. When the negative value is computed the simulation algorithm should provide another value or try to use the normal distribution a few more times to get positive number.

Conversion to Petri Net

An activity with specified duration will be converted to the timed Petri Net transition with stochastic duration defined by the normal distribution evaluated from given interval (as described earlier). This transition will have the same input and output places as the original activity and his timer will be in the Enabling Memory state (the timer will be restarted when the transition is not enabled) (see (Rozenberg and Reisig 1998)).

An example of such conversion is shown on Figure 2. Activity called “Find applicant” is in this example set to last for 20-40 minutes.

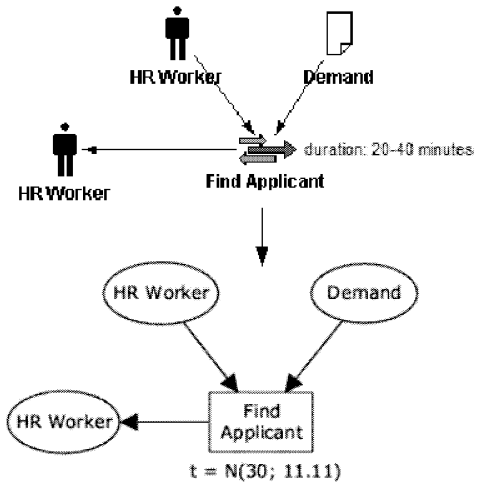


Figure 2: Conversion of varying duration

Necessary extension of the model

No new modeling elements have to be added to the models to extend the Coordination View in such manner. Every activity should only provide means to define two parameters – minimal and maximal duration of activity, two values defining the duration interval. These parameters aren't even important in the modeling of the process but only during the simulation and so they don't have to be included in the standard modeling diagrams.

VARYING WAITING DURATION OF ACTIVITIES

Waiting duration of activity is very similar to the normal duration but with on very important difference. The activity has to wait for some external factor to be able to start performing the activity, but no resources are used while waiting. Both the normal duration of activity and waiting duration are inseparable properties of the process but their difference has great impact on the resource availability. Therefore these two durations are described separately and one activity can utilize both durations.

Stochastic properties

Variable waiting duration has the same stochastic properties as normal duration of activity (see chapter *Variable duration of activity*).

Conversion to Petri Net

Even though the stochastic properties are the same, conversion to Petri Net is different from the normal duration conversion. Activity with waiting duration will be divided into two transitions connected with one common place. The first transition will have only one input place and one output place and this transition will model the waiting duration of activity. The second transition's input will be connected to the output place of the first transition. In addition to this place the second transition will also have all input and output places of the converted activity. This transition himself will therefore describe the performance of the activity and its normal duration.

An example of such conversion can be seen on Figure 3. Waiting duration of the activity “First interview” is set to 1- 3 days and the normal duration is 15-30 minutes. (All duration parameters are converted to minutes for the final probability distributions.)

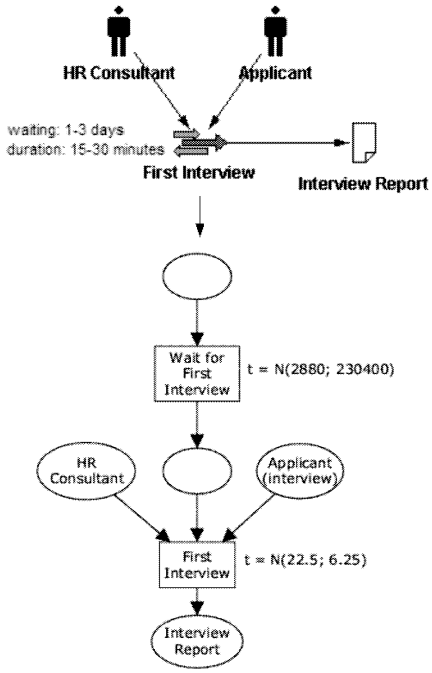


Figure 3: Conversion of waiting duration

Necessary extension of the model

Because the stochastic properties of waiting duration are the same as for the normal duration, the added elements will be the same. Therefore the minimal and maximal waiting duration will be added to every activity as simulation properties.

PROBABILITY OF ACTIVITY SCENARIOS

The BPM Method uses activity scenarios to represent the alternative flow in the Coordination View. These scenarios describe various routes through the activity that can lead to different outputs. That's why every activity scenario can have different duration but all share one common waiting duration.

Activity scenarios are the only possible way to define alternative flow in the Coordination Model. If the model is constructed incorrectly and conflicting transitions are present in the net, the resulting Petri Net will fire the transition with the lowest duration or one of the conflicting immediate transitions will be chosen in the nondeterministic manner.

Stochastic properties

It is necessary to decide which of the activity's scenarios should be performed during each cycle of the simulation. And so every possible scenario will be extended by the percent probability that they will be performed during this run. The cumulative probability of all scenarios for one activity has to be 100% (no undefined states are allowed).

Every scenario will have its own duration according to the *Varying duration of activities* chapter but waiting duration of all activity's scenarios will be the same.

Conversion to Petri Net

Activity with *n* scenarios will be converted to *n* Petri Net transitions with the same input places as the underlying activity and the probability of firing these conflicting transitions will be set according to the percent probabilities of individual scenarios. Every one of these transitions will feature only one output place that will connect them to the additional transition for each scenario. These second transitions will depict the actual realization of each scenario with the proper duration and proper output places for each scenario (as defined in the Coordination View).

Every scenario is split into two transitions because of the Timed Petri Nets standard behavior. When two conflicting transitions in the Petri Net are enabled, the one with shorter duration is fired. If this was the case, the faster scenario would always fire and the slower scenario wouldn't have a chance to be performed. The first row of immediate transitions with probabilistic firing ensures that every scenario is realized with given probability.

Figure 4 depicts the conversion of activity “Contact applicant” to the Petri Net. Scenario 1 is 80% probable and its duration is 5-8 minutes. Scenario 2 is 20% probable and lasts 3-6 minutes.

Necessary extension of the model

Each scenario will be extended by the percent probability for his activating and with minimal and maximal duration of this scenario. Both these properties will be important only for the simulation view, not for the modeling view.

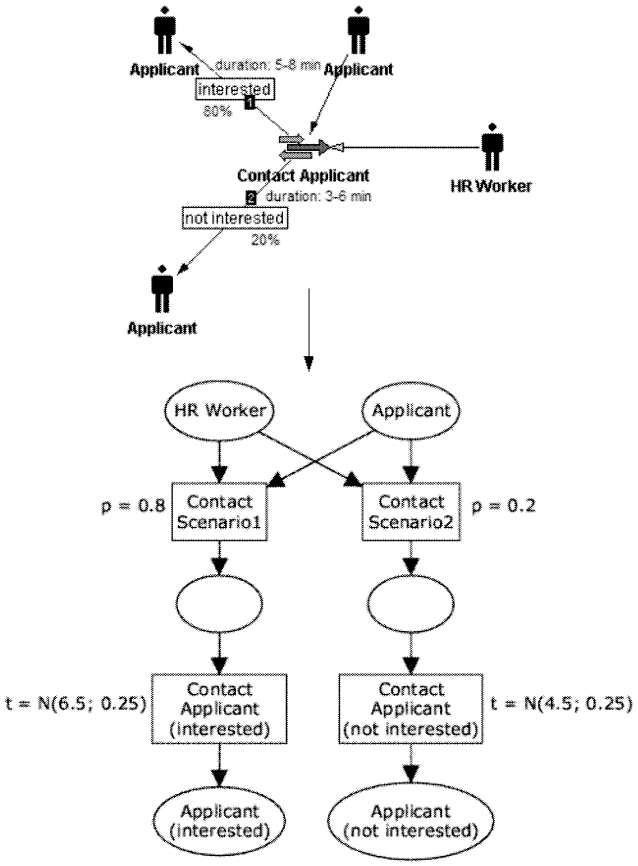


Figure 4: Conversion of activity scenarios

ENTRY OF NEW CUSTOMERS TO THE PROCESS

The only stochastic aspect that isn't directly connected with activities is the delay between an entry of new customers to the process. With every new customer that enters the process, new instance of the process will be created and simulated individually. The only interaction between two instances is the sharing of resources.

Stochastic properties

Entry of new customers to the system is described by the queuing theory as a Poisson's process and the delay between individual contracts is characterized by exponential distribution (see (Gross and Harris 1998)). This generalization isn't very precise because Poisson's process expects that the intensity of his occurrences is constant and each event is independent from the others. Acquisition of new customers doesn't have such characteristics in the long-term view. But the exponential distribution is sufficient to describe short-term simulation and measurement of performance parameters because the intensity of new contracts can be calibrated to correspond with standard scenarios. Thanks to this calibration the process can be analyzed for periods with lower customer acquisition and then compared to the analysis of higher customer acquisition behavior. Long-term simulations and quantity predictions aren't suitable for this distribution, but this extension of the BPM Method isn't created to support these.

The delay between individual customers T will therefore be simulated by exponential distribution $T \rightarrow E(\lambda)$ that will

be defined by its mean value *E(T)* for simplicity's sake. This value describes mean delay between acquisition of new contracts. The intensity of Poisson's process λ is then evaluated from the mean value as

$$\lambda = 1 / E(T)$$

Intensity is the only needed parameter of the exponential distribution. One additional parameter is needed though and that is the number of customer entries. This parameter makes it possible to finish the simulation after defined number of customers. No new simulation instances of the process will be created after the number reaches the prescribed limit.

Conversion to Petri Net

Two possibilities appear to be of any value in the conversion of this stochastic aspect. The first possibility is to extend the final Petri Net with additional input timed transitions that will generate tokens to the net based on the exponential distribution. This method will significantly change the reachable states of the Petri Net because it will contain any number of concurrent customers.

The second possibility is to create new instance of the net every time new customer enters the process. More instances of the same Petri Net will coexist, but every one of them will contain only one customer. The availability set of every instance will be the same and the properties of the system will be preserved. The greatest feedback of this method is that it doesn't support the dependencies between two customers' processes and the only allowed interaction is by sharing common resources.

The second technique is more suitable for the domain of the BPM Method and will be used for this extension. Business processes are mainly based on the independent satisfaction of individual customers.

The resulting Petri Net won't need to be extended by any new elements due to the choice of the second method. New instance of the Petri Net will be created for every occurrence of the Poisson's process and a token will be generated into the proper input place.

Figure 5 shows the possible resulting form of the Petri Net converted from the Coordination View in Figure 1. Places called “Demand” and “Applicant” are two of process' input places with mean values 1 day and 5 hours and limits of 20 and 50 customers respectively. (All duration parameters are converted to minutes for the final probability distributions.)

All remaining input places of the process characterize available external resources or internal resources that should be filled by tokens in accordance to the availability of such resources. The checking of availability and qualification of these resources is beyond the scope of this paper and will be studied in further papers. In this moment the method presumes that all internal resources are available and so the new instances will be created with one token in these input places.

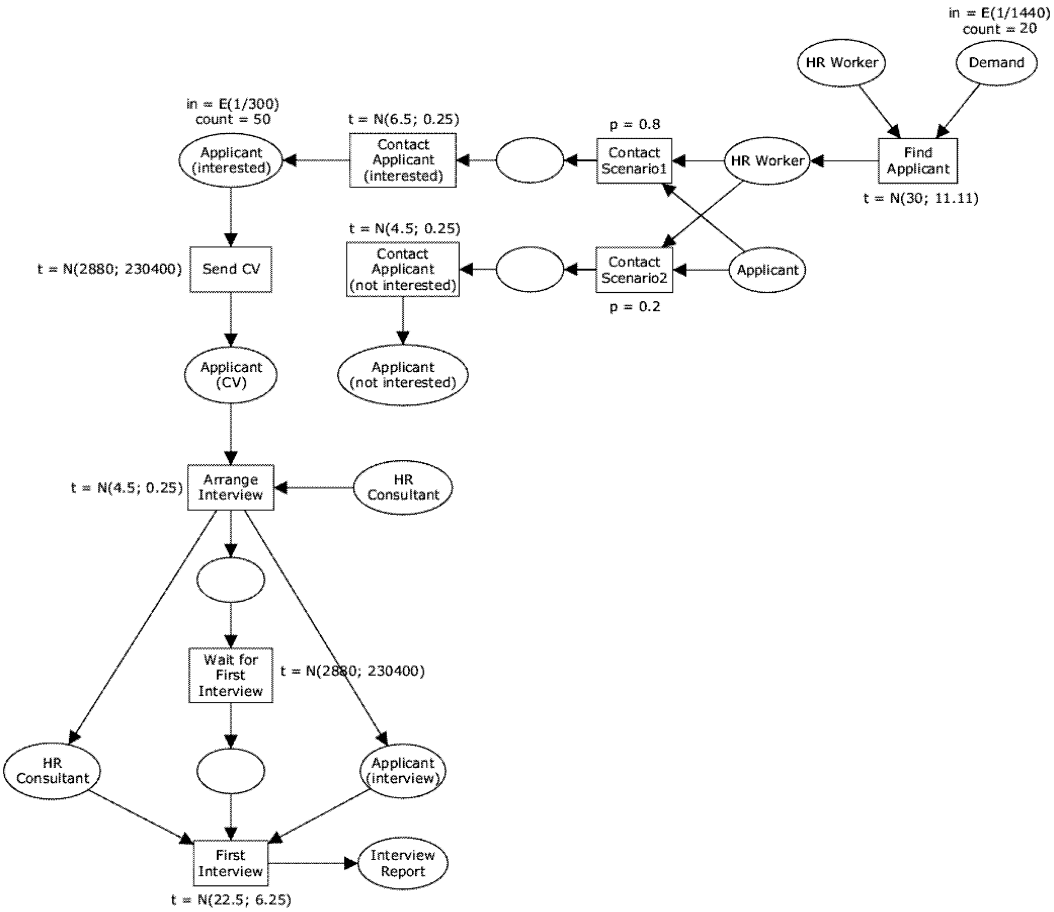


Figure 5: Final Petri Net converted from the Coordination View from Figure 1

Necessary extension to the model

Only the input places where the customers will be introduced to the process will gain two new parameters – mean value of the exponential distribution and limit number of acquired contracts. Both parameters will only serve as simulation parameters.

SIMULATION EXPERIMENT

After defining these extensions and implementing them to the BP Studio modeling software (mentioned in the introduction to this paper), we also performed a simulation experiment based on the example process that was presented in this article. This example is only demonstrative and it is necessary to create a model of real process comparable with the reality.

Table 1 and 2 contain statistical results for activities performed in the process where the input was produced in the Demand place. In results in Table 1 the chosen Contact Applicant activity scenario was number 1 (Applicant is interested).

Table 1: Results for Demand Input and Interested Scenario

Activity	Count	Minimum duration	Average duration	Maximum duration
Find Applicant	16	21 minutes	28.81 minutes	34 minutes
Contact Applicant	16	5 minutes	5.31 minutes	6 minutes
Send CV	16	1 day, 10 hours, 55 minutes	2 days, 47 minutes	2 days, 12 hours, 31 minutes
Arrange Interview	16	3 minutes	3.38 minutes	4 minutes
Waiting time for the First Interview	16	1 day, 10 hours, 21 minutes	1 day, 23 hours, 6 minutes	2 days, 10 hours, 39 minutes
First Interview	16	17 minutes	22.19 minutes	26 minutes
Whole Process		3 days, 2 hours, 43 minutes	4 days, 53 minutes	4 days, 19 hours, 5 minutes

Table 2 shows the results for scenario number 2 (Applicant isn't interested).

Table 2: Results for Demand and Not Interested Scenario

Activity	Count	Minimum duration	Average duration	Maximum duration
Find Applicant	4	27 minutes	30 minutes	33 minutes
Contact Applicant	4	3 minutes	4 minutes	5 minutes
Whole Process		31 minutes	34 minutes	37 minutes

Table 3 contains results for activities performed in the process with entries produced in the input Applicant place.

Table 3: Results for Applicant Input

Activity	Count	Minimum duration	Average duration	Maximum duration
Send CV	50	1 day, 5 hours, 30 minutes	2 days, 55 minutes	2 days, 19 hours, 7 minutes
Arrange Interview	50	3 minutes	3.98 minutes	5 minutes
Waiting time for the First Interview	50	1 day, 6 hours, 27 minutes	1 day, 23 hours, 24 minutes	3 days, 2 hours, 31 minutes
First Interview	50	18 minutes	22.1 minutes	27 minutes
Whole Process		2 days, 18 hours, 51 minutes	4 days, 46 minutes	5 days, 3 hours, 19 minutes

CONCLUSION

Extended mechanisms of BPM's Coordination View suggested in this paper can be used for running basic simulation experiments on business processes modeled by this method. Coordination View is converted into the Petri Net that will serve as a pattern for creating instances of the process for individual customers. Exponential distribution is used for defining the entry of new customers to the process but its properties aren't very good for long-term predictions and simulations when the intensity of customers fluctuates.

Another significant limitation of this extension is sharing of the resources between more instances of the process. All internal resources are always available and this contradicts the reality. Our future work will concern this issue and we will try to describe the abilities and skills of various resources and develop a way to find the best suitable available resource to be used in the simulated activity.

Our next effort will be dedicated to finding other suitable distributions for defining activity durations that remove the problem with negative values produced by the distribution.

REFERENCES

Gross D., Harris C.M. 1998. *Fundamentals of Queueing Theory*. Wiley Publishers, New York.

Kuchař Š. 2007. *Analysis and Design of Human Resources System*. M.S.thesis. VŠB – Technical University of Ostrava.

Longnecker M. 2008. *An Introduction to Statistical Methods and Data Analysis*. Cengage Learning.

Peterson J.L. 1981. *Petri Net Theory and the Modelling of Systems*. Prentice Hall.

Rozenberg G., Reisig W. 1998. *Lectures on Petri nets*. Springer.

Vondrák I., Szturc R., Kružel M. 1999. "Company Driven by Process Models", *European Concurrent Engineering Conference ECEC '99*, SCS, Erlangen-Nuremberg, Germany, 188-193.

Vondrák I., Szturc R., Kružel M. 1999. "BPM – OO Method for Business Process Modeling", *ISM '99*, CSSS, Rožnov pod Radhoštěm, 155-163

Vondrák I. 2000. *Business Process Studio, version 3.0*. Electronic manual, VŠB – Technical University of Ostrava.

Vondrák I. 2002. *Introduction to Software Engineering*. VŠB – Technical University of Ostrava.

Vondrák I., Fedorčák D., Kožuszník J. 2006. Business processes. *Informatics for geoinformatics Conference 2006*, Ostrava.

Wang J. 1998. *Timed Petri Nets: Theory and Application.* Springer.

Acknowledgements

This research has been supported by internal grant agency of VSB-TU of Ostrava - SP/2010214 Modeling, simulation and verification of software processes II.

MANUFACTURING OPTIMIZATION

A LEGO FACTORY FOR TEACHING SIMULATION-BASED PRODUCTION OPTIMIZATION

Anna Syberfeldt
Virtual Systems Research Center
University of Skövde
541 48 Skövde,
Sweden
E-mail: anna.syberfeldt@his.se

KEYWORDS
Simulation, Optimization, Production, Education.

ABSTRACT

To analyze a production system and find its best configuration, the technique of simulation-based optimization is often utilized. This paper describes a practical exercise used at the University of Skövde, Sweden to teach simulation-based optimization to students using a physical Lego factory. The Lego factory provides a setting that is close to reality, and optimizing its production assists students to deepen their understandings of simulation-based optimization.

INTRODUCTION

Simulation-based optimization is an important technique in production industry to improve efficiency. Simulation-based optimization is the process of finding the best parameter values for a system, in which the performance is evaluated on the basis of output from a simulated model of the system (Swisher et al., 2000). The process is iterative; an optimization algorithm generates a set of input parameter values (referred to as a solution) and feeds them to a simulation model, which computes one or multiple performance measures of the system (called objective values or fitness) (Figure 1). In a production system, for example, the input parameters might be buffer sizes and the performance measures might be throughput of the system (i.e. production capacity) and work-in-process (i.e. partially completed crankshafts that are somewhere in the manufacturing process and not ready for delivery). Based on the evaluation feedback obtained from the simulation model, the optimization algorithm generates a new set of parameter values and the generation-evaluation process continues until a user-defined stopping criterion is satisfied. Such a criterion may, for example, be that a certain amount of time has passed, or that specific objective values have been achieved.

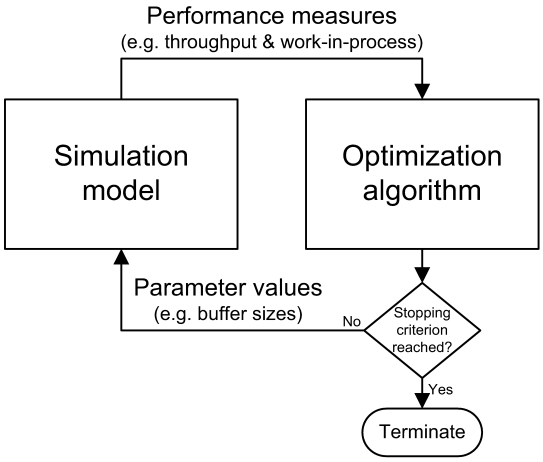


Figure 1: The principle of simulation-based optimization.

When a new production system is to be set-up, or an existing one is to be modified, a thorough analysis with the use of simulation-based optimization is often carried out to find the best configuration. For this task, an expert in simulation-based optimization is needed in order to utilize the technique to its full potential. With the aim of producing such experts, courses in simulation and optimization have been offered for some years at the University of Skövde, Sweden (see de Vin and Jägstam, 2001). This paper describes an innovative exercise used in one of these courses to teach simulation-based optimization in a practical manner.

The exercise takes place upon a physical mini factory built using Lego and Lego Technic (www.lego.com). The mini factory provides a setting that is close to reality, and optimizing its production assists students to deepen their understandings of simulation-based optimisation. We also believe that experiential learning opportunities enhance and enrich the learning experience.

Figure 2 shows a picture of the factory. The factory is made up of three production stations, and the base of each station is about 40x40 centimeters. In total the factory is about 1.5 meters in height and 1 meter in width. The stations are placed above each other in a spiral and are connected with PVC tubes, allowing gravity to transport the bubble gums between stations. A conveyor belt transports the bubble gums from the last station to the first, making the system a closed loop and removing the need to feed new raw material

to the system at runtime. The system can thus run indefinitely without manual intervention.

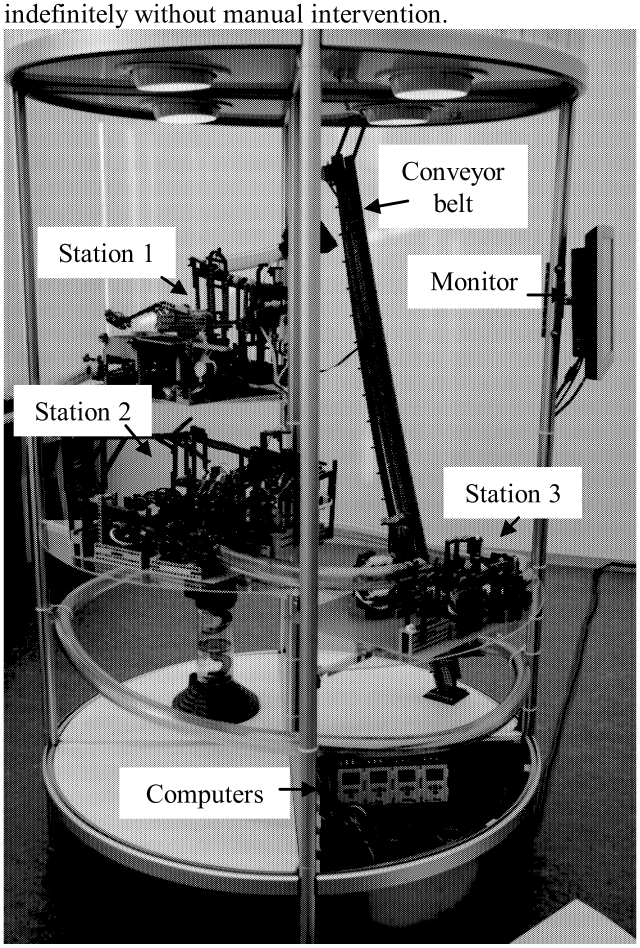


Figure 2: Lego factory.

In the next section, details about the production system of the factory are provided.

DESCRIPTION OF PRODUCTION SYSTEM

The mini factory simulates refinement of bubble gums using three production stations. These are described in the following section, and thereafter the production process is presented.

Production Stations

The three production stations are controlled using Lego Mindstorm NXT 2.0 computers. The NXT computer has a 32-bits micro processor, four input sensors, and three motor outputs. These sensor inputs and motor outputs are used to control the production stations.

The general design is the same for all three stations. When a bubble gum arrives at the station, a light sensor reads the color of the gum (blue, red, green, or yellow). If the specific color should be processed in the station, it is placed in a buffer in front of the station – otherwise it passes by the station. The buffer holds bubble gums that await processing, similarly to a real production process. The space of a buffer (how many gums it can hold) is variable and can be changed by the "production manager". Each buffer has a minimum size of 1 and a maximum size of 5.

The first production station simulates flavoring. In this station, the bubble gum is placed in a spoon-like construction and dropped into a container, which is then covered with a lid for a few seconds. When the lid opens, the spoon lifts up the bubble gum and drops it into the tube. The bubble gum then proceeds to the second station.

The second station simulates drying. In this station, a bubble gum is placed at the end of an arm that spins quickly in order to air dry the bubble gum. The bubble gum is then put into the tube and rolls to the third station.

The third station simulates polishing. In this station, the bubble gum is placed under a rotating brush. After polishing, the bubble gum rolls to the conveyor belt and is transported back to the first station.

Production Process

The production system handles four different types of bubble gums: blue, red, green, and yellow. The various types are processed differently, meaning, for example, that a blue bubble gum should pass through certain stations while skipping others. A red bubble gum may follow different rules. The production process is further complicated by the fact that for a specific type of bubble gum there is a set of different legal process configurations, as shown in Table I. A blue bubble gum, for example, could go through the stations for flavoring and drying, or through the stations for flavoring and polishing, or through only the polishing station. Only one of these three configurations can be used, and which one to use up to the "production manager" to decide. In the assignment, the students are appointed "production managers" for the Lego factory and their task is to find the best configuration possible. The next section describes this task in further detail.

TABLE I
POSSIBLE CONFIGURATIONS OF THE PRODUCTION FLOW

Blue	Red	Green	Yellow
Flavor & dry **OR** Flavor & polish	Flavor **OR** Dry **OR** Polish	Flavor & dry **OR** Dry & polish **OR** Flavor & polish	Flavor & polish **OR** Dry & polish

DESCRIPTION OF EXERCISE

In the exercise, the task of the students is to find the best configuration possible for the production process in order to maximize profit. The students can change two aspects of the production process; the production flow, and the buffer sizes. To make production efficient, the production flow and the buffers should be configured in a way that minimizes work-in-process and maximizes throughput. Work-in-

process is the number of bubble gums currently being processed, while throughput is the number of bubble gums that the system outputs per minute.
Work-in-process and throughput are two important objectives in real production scenarios. The amount of work-in-process in the production system should be kept as small as possible in order to avoid capital being tied up in partially completed products. At the same time, the throughput rate should be kept as high as possible since the higher the throughput rate, the better the productivity and profitability.

Achieving a low work-in-process and a high throughput rate is not trivial, neither in reality nor in the bubble gum factory. With large buffers, the throughput increases since machine starvation is avoided and the impact of the interference caused by variability in processing times decreases. However, with large buffers the work-in-process also increases since a large number of bubble gums are kept in the system. Conversely, small buffers decrease both the work-in-process and the throughput. Further complexity is added by the fact that it is not only the buffer sizes that are variable, but also the process flow (Table 1). Predicting the effect of various configurations of the buffers and the process flow is not possible, but each of them must be tested in order to find best one. Testing all 4500 possible configurations is impossible in practice. One of the goals of the exercise is to make the player realize the infeasibility of exhaustive testing and the benefits of simulation-based optimization.

For solving the exercise, the students must complete two parts; simulation and optimization. These two parts are described in the following subsections.

Part 1: Simulation

In order to evaluate different configurations and identify the best one, a simulation model that imitates the factory is needed. The simulation model is a black-box function evaluator in the sense that the relationships between its input parameters and output values are of a closed form (April et al., 2001). As a consequence, there must be a complete separation between the model that represents the system and the algorithm that is used to solve the optimization problem. An advantage of this separation is that the model and the optimization algorithm are independent of each other and can be developed separately (Laguna and Marti, 2003).

There are many types of simulation methodologies possible for production systems. Among these, discrete-event simulation (DES) is probably the most common one. Using this methodology in the exercise is, however, considered inappropriate. This is since it would take too long time for the students to construct a sufficient DES model for the factory. Instead, the technique of Artificial Neural Networks (ANNs) is used for simulating the factory. Once historical data has been collected, an ANN is very quick to construct. The fundamentals of this technique are described in the next section.

Artificial Neural Networks

In general terms, an ANN is a non-linear statistical data modeling method used to model complex relationships between inputs and outputs (Mehrotra et al., 1996). The inspiration for the technique originated from the area of neuroscience and the study of the neurons as information processing elements in the central nervous system. Essentially, artificial neural networks are simple mathematical models defining a function $f:X \rightarrow Y$. The function f(x) is defined as a composition of a number of other functions $g_i(x)$, which can also be defined as a composition of a number of other functions, altogether represented as a network structure. A variety of different types of ANN architectures exists. The one most commonly used is the feedforward artificial neural network, in which the information only moves forward from the input nodes, through the hidden nodes and to the output nodes via a series of weights (Villaseñor et al., 2006). The sum of the products of the weights and the inputs is calculated in each node, and if the value in an output node is above some threshold the node is activated. The activation is determined by an activation function, such as the sigmoid function or the binary threshold function. An example of a feedforward ANN is shown in Figure 3.

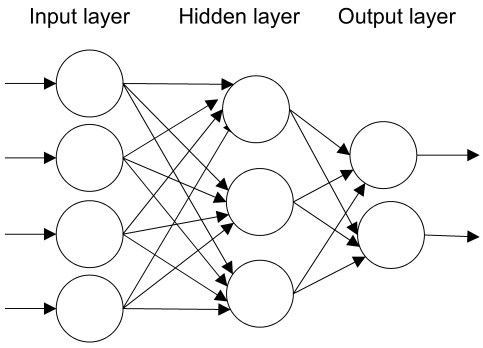


Figure 3: Feedforward ANN.

The students may implement any type of ANN, but they must construct the ANN in a way that makes it possible to use it as fitness function in the evolutionary optimization algorithm that they are to develop in the next step. For not misguiding the evolutionary optimization algorithm, it is important that the ANN represents the real factory as closely as possible.

The students are not given any instruction on *how* to construct the ANN, but need to find this out themselves. They are allowed to use a code library for solving the task, for example FANN[1]. For training the ANN, the students provided a data set of 100 historical samples. The samples are collected by running the Lego factory with different configurations and collect their corresponding outcomes.

[1] Fast Artificial Neural Network library, http://leenissen.dk/fann

When designing the ANN, the students must consider aspects such as network topology, data representation, learning parameters, training parameters, and training termination. The students are asked to select a number of parameters that they think have the greatest influence on the ANN's performance, and perform experiments on their settings. They should figure out which configuration that leads to the best performance of the network and try to find an explanation why. They are asked to save their best ANN to be used as fitness function in the evolutionary algorithm.

After constructing the ANN, the students are expected to submit a short report answering the following questions:

1. Describe (in text and graphically) your selected network topology and how you represent factory configurations in the input layer.
2. Present a graph of the RMSE[2] as a function over time for both the training and test sets. Mark at what time you terminate the training.
3. Describe and motivate your choice of training method.
4. Why is it important to verify the network against a test set?
5. Describe and motivate your training termination conditions.
6. Why is a RMSE of ~0 on the training set undesirable in a stochastic environment? What happens to the RMSE of the test set as the RMSE of the training set approaches ~0 and why?

When the students have completed the report, they continue with the next part of the exercise; optimization.

Part 2: Optimization

In the second part of the exercise, the students should construct an AI-based algorithm for optimizing the buffers and the production flow with respect to work-in-process and throughput. As fitness function for the algorithm, the ANN they developed in the previous step must be used. Many different AI techniques are possible in production optimization, and for the assignment an evolutionary algorithm should be used.

The idea behind evolutionary algorithms (EAs) is to use computational models of evolutionary processes in the design and implementation of problem solving applications. Based on Darwin's theory of "survival of the fittest", candidate solutions to a problem are iteratively refined. Generally, an EA consists of a genetic representation of solutions, a population-based solution approach, an iterative evolutionary process, and a guided random search.

In evolving a population of solutions, EAs apply biologically inspired operations for selection, crossover and mutation. The operators are applied in a loop, and an iteration of the loop is called a generation. The solutions in the initial population are usually generated randomly. During each generation, a proportion of the solutions in the population is selected to breed offspring for the next generation of the population. Solutions are selected based on their fitness, representing a quantification of their optimality. Typically, solutions with high fitness have a higher probability o be selected, but to prevent premature convergence, it is common that a small proportion of solutions with worse fitness are also selected. From the solutions selected, new solutions are created to form the next generation of the population. For the creation of each new solution, two parent solutions are chosen and through mating (called crossover) an offspring is produced. Occasionally, the new solution can undergo a small mutation in order to keep the diversity of the population large and avoid local minima. A mutation usually involves changing an arbitrary part of a solution with a certain probability. When the new population is formed, the average fitness will have generally increased. The process of evolving generations continues until a user-defined termination criterion has been fulfilled, for example that the best solutions in the last n generations have not changed.

As with the ANN, the students are not given any instruction on *how* to construct the EA, but need to find this out themselves. In this case they are **not** allowed to use a code library for solving the task, but must develop the EA from scratch.

In the exercise, there are two optimization objectives that must be considered by the EA: work-in-process, and throughput. The difficulty with this problem is that there is no single optimal solution with respect to both the objectives, as improving the performance of one objective means decreasing the performance of the other.

With large buffers, the throughput increases since machine starvation is avoided and the impact of the interference caused by variability in processing times decreases. However, with large buffers the amount of work-inprocess also increases since a large number of crankshafts are kept in the system. Conversely, small buffers decrease both the work-in-process and the throughput. This trade-off problem between throughput and work-in-process is illustrated with an example in Figure 4. In this example, there are five solutions to the problem, none of which is superior to any other when considering both objectives at the same time.

[2] Root Mean Square Error

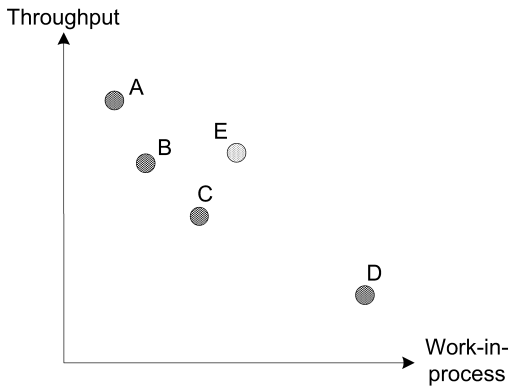


Figure 4: Trade-off problem.

Instead of a single optimum for the problem, there is a set of optimal trade-offs between the conflicting objectives, which is usually called Pareto-optimal (Deb, 2004). For Pareto-optimal solutions there is no other solution that is superior in one objective without being worse in another. In order to manage multiple objectives, specific multi-objective EAs have been suggested. Instead of only seeking a single optimum, these algorithms maintain a set of Pareto-optimal solutions. In the exercise, the students must implement one of these algorithms.

After completing the implementation, the students are expected to submit a short report describing and discussing the design of the EA and its results. They should also answer the following questions:

1. Describe and motivate your choice of EA. Also present a graph of your population after 0, 10 and 100 generations.
2. Briefly describe the two implemented selection schemes and motivate why you chose one over the other.
3. Briefly describe the two implemented recombination operations and motivate your choice of one over the other.
4. Briefly describe the two implemented mutation operations and motivate why you chose one over the other.
5. From the evolved Pareto front, draw conclusions about how different configurations affect the optimization goals and briefly discuss three of your conclusions.

When the report has been submitted, the exercise is corrected by the examiner and the results are gone through with the students. The students are also asked to evaluate the exercise, as described in the next section.

EVALUATION OF EXERCISE

Until today, the exercise has run once with 13 participating students. After the course had ended, the students were asked to share their opinions about the exercise by answering three questions in writing. The first questions asked about positive aspects, the second one about negative aspects, and the third one about possible improvements of the exercise.

On the positive side, a majority of the students mentioned the experience of working with a physical system, and several of the students expressed that this was their first time doing so. They liked the opportunity to experiment and put the theories taught into practice. Furthermore, the students pointed out the practical elements as encouraging and motivating in solving the exercise. All of the students believed that the assignment had given them useful knowledge when it comes to simulation-based production optimization.

On the negative side, some of the students mentioned the open-ended nature of the assignment. As described in the previous section, the students were not provided any instructions on *how* to solve the task, but they needed to find this out themselves (problem-based learning). However, having no clear path to go was experienced as frustrating to some of the students.

SUMMARY

This paper describes an practical exercise used at the University of Skövde, Sweden to teach simulation-based optimization. The exercise uses a physical Lego factory as platform for experiments. The Lego factory provides a setting that is close to reality, and optimizing its bubble gum production assists students to deepen their understandings of simulation-based optimization. The experiential learning opportunity also enhances and enriches the learning experience.

In the exercise, the students should implement simulation-based optimization for improving the bubble gum production with respect to work-in-process and throughput. Two aspects are possible to configure in the optimization process; buffers and production process. The students are not given any instruction on how to complete the exercise, but need to find this out themselves (so called problem-based learning).

Evaluation of the assignment shows that the students enjoy working with a physical system and put theories into practice. However, several of them expressed that they were unused to the open-ended nature of the assignment, and experienced that is was frustrating to have no clear path to go. To address this feedback, some general guidelines on how to tackle the exercise should probably be provided to the students in the future.

REFERENCES

April, J., Glover, F., Kelly, J. and Laguna, M. (2001) Simulation/optimization using "real-world" applications, Proceedings of the 2001 Winter Simulation Conference, Arlington, VA, pp. 134–138.

Deb, K. (2004) Multi-Objective Optimization using Evolutionary Algorithms, second edn, JohnWiley & Sons Ltd.

Mehrotra, K., Mohan, C. and Ranka, S. (1996) Elements of Artificial Neural Networks, MIT Press ISBN 0-262-13328-8.

Swisher, J., Hyden, P., Jacobson, S. and Schruben, L. (2000) A survey of simulation optimization techniques and procedures, Proceedings of the 2000 Winter Simulation Conference, Orlando, Florida, pp. 119–129.

de Vin, L. and Jägstam, M. (2001) Why We Need to Offer a Modeling and Simulation Engineering Curriculum. Proceedings of the 33nd Winter Simulation Conference. Arlington, Virginia pp. 1599 – 1604.

Villaseñor, D., Morales-Menéndez, R., Rodríguez, C. and Alique, J. R. (2006) Neural networks and statistical basedmodels for surface roughness prediction, Proceedings of the IASTED International Conference on Modeling, Identification, and Control, Lanzarote, Spain, pp. 326–331.

AUTOMATIC ANALYSIS OF MANUFACTURING SIMULATION OUTPUT

Habtom Mebrahtu, Rob Walker
Department of Computing and Technology
Anglia Ruskin University
Bishop Hall Lane, Chelmsford
Essex CM1 1SQ, UK
Habtom.Mebrahtu@anglia.ac.uk, Rob.Walker@anglia.ac.uk

KEYWORDS
Manufacturing Simulation, Manufacturing Analysis, Simulation Optimisation, Rule-based System

ABSTRACT

This paper demonstrates the need for a consistent mechanism that can be used to assist users in interpreting and analysing simulation results without a special knowledge requiring expertise. It discusses and demonstrates the use of an expert mechanism that reads simulation output results, interprets and analyses performance, and then decides on the changes to be made before the next run. Most of the published and commercially applied optimisation techniques depend on mathematical methods and search engines. These techniques, as effective as they are, do not make use of proven manufacturing performance enhancing techniques such as the Theory of Constraints. In this paper the necessity of integrating optimisation and manufacturing simulation technologies is discussed, some of the existing simulation optimisation methods briefly looked into, and the main methodology in integrating simulation with optimisation method described. A representative model in the context of a manufacturing simulation integrated with an expert mechanism for automatic performance enhancement is demonstrated and the results discussed. The paper also discusses the need for checking statistical accuracy of the model, using a Cusum chart, before considering the model as being stable.

1 INTRODUCTION

The pace of change in manufacturing industry is growing fast. Product life cycles have been decreasing and companies are facing tougher competition as customers are becoming more demanding (Randel et al 1999). Simulation, unlike spreadsheets, treats the whole workshop as a system and is able to model system relationships, independencies and interactions. Simulation is one of the most powerful analysis tools for the design and operation of complex processes and systems (Law and Kelton 2000, Spall 2003).

Simulation modelling as an effective analysis tool used to be a niche management property that could only be afforded by a few because of the costs and the expertise required. This is no longer the case thanks to the ever increasing computer power and affordable price. The advancement in programming and software engineering also means that very clever simulation packages have hit the market, with highly configurable user features and powerful animation.

Despite the power and flexibility of simulation in modelling existing or proposed systems, the performance of a simulation model mostly depends on a large number of parameters or factors that act and interact in a complex manner (Alam et al 2002). With simulation modelling, the relationships between the design parameters and their resulting performance are not explicitly known. Therefore, simulation modelling becomes a trial-and-error process in which a set of input factors is used to generate output performance measures. Unfortunately, the repetitive nature of this approach is often inefficient in terms of time and computing cost as well as in interpretation and prediction of results.

Although their prime objective is to provide decision support, virtually all simulation models are input-output models. They can only help in describing the system by yielding outputs for given inputs but cannot generate optimal solutions on their own. The process of interpreting and assessing simulation outputs to improve performance has to be done by the user.

Significant investigation on the need for simulation optimisation has been carried out. Among the methods described in the literature include Response Surface Methodology (Nicolai et al 2004, Box and Draper 1987), Simulation Meta-models (Kilmer et al 1997, Alam et al 2002), Expert Systems (Agarwal and Babu 1994, Lyu and Gunasekaran 1997) and Metaheuristics.

Commercially used optimisation methods fall mainly under Metaheuristics. Examples are Simulated Annealing (Barretto et al 1999, Debuse et al 1999, Avello et al 2004), Scatter Search and Tabu Search (Glover and Laguna 1997), and Genetic Algorithms (Baccouche et al 2003, Brusha and Franek 2003). All of these methods are based on analytical optimization techniques.

2 SIMULATION OPTIMISATION

Simulation optimisation is described as a process where the objective function, constraints or both are responses that can only be evaluated by computer simulation. Suppose a simulation model M, has *n* input variables (x_1, x_2,… x_n) and *m* output variables (y_1, y_2,…, y_m), the objective of the simulation optimisation is to find the optimum values (x_1*, x_2*,…, x_n*) for the input variables (x_1, x_2,… x_n) that optimise the output variable(s). That is, it is the combination of an optimisation method with a simulation model to determine the input variable settings that maximise the performance of the simulated system (Azadivar 1992). In order to solve a simulation optimisation problem both the operational simulation model and an optimisation method or procedure are needed.

Most of the optimisation methods mentioned above have not been properly applied in industries due to their demand of high level of sophistication on the part of the user. April et al (2003), while describing the amount of attention given to simulation optimisation from the research community, expressed that the analytical techniques (such as response surface methodology) require considerable technical depth that they have not been favoured by industrial users. The most commercially applied method, metaheuristics, performs some sort of search for optimal values of input parameters. Yet, the procedure usually needs to run a large number of alternative models before deciding on the best solution, and it often requires a substantial amount of computer power and time.

An alternative method, as demonstrated in this paper is to employ an expert mechanism that will run a simulation model, identify the influencing factor(s) such as the bottleneck, and make effective changes in the next run that will result in better performance.

3 EXPERT MECHANISM

3.1 Definition

In this paper an expert mechanism is a rear-end addition to a manufacturing simulator that could assist in interpreting simulation results, assessing performance and automatically adjusting input data of the model for the next run, to improve model performance. The mechanism contains rule-based knowledge that reads simulation output, analyses the results, assesses the level of performance, identifies the main influencing factor(s) and decides on actions to be taken in the next simulation run.

3.2 Methodology

One way of assuring compatibility between the simulator and the expert mechanism is by introducing an object linked embedding (OLE) automation. The simulator used is Witness and can be controlled by other programs, that is, Witness is an OLE automation server. Visual Basic, as an OLE automation controller, is one of the programs that can control Witness (Lanner Group 2008).

Relevant input/output values and the running of Witness can be controlled by Visual Basic with minor support from Excel. Visual Basic being able to control both Witness and Excel is an ideal language to use for developing the expert mechanism. A simplified diagram depicting the system is shown in figure 1.

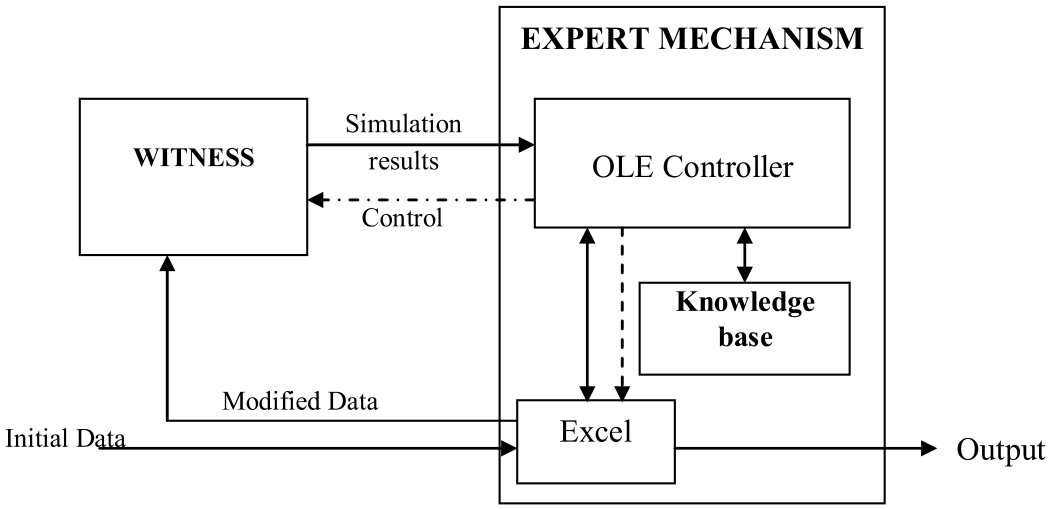


Figure 1. Integrated Simulator-Expert Mechanism Model (Mebrahtu et al, 2009)

Input data for cycle time, set-up time, machine break down, buffer size, conveyor details, etc can be displayed on Excel but controlled by Visual Basic. The simulator uses these input data, runs the model and generates simulated output. The expert mechanism receives the relevant output data from the simulator, manipulates the data (with some help from Excel), assesses model performance and generates recommended changes on the next input data to the simulator. The iteration goes on until a limiting factor is reached, at which point the result would be output.

4. EXPERIMENTAL FRAMEWORK

4.1 Case Study and Objectives

Possibilities of improving throughput, reducing operating costs, reducing work in progress, etc of an existing system can be deduced from analysis of simulation outputs. When the resources are limited, however, one of the common ways in industry of meeting extra demands is through overtime work. The following case study is used to demonstrate the concept of introducing an expert mechanism to improve throughput and operating cost.

The Case Study

The model is a flywheel and starter ring gear manufacturing line that the authors re-planned and installed the extra machines required to increase capacity. The objective was to increase throughput from 1250 to close to 4000 flywheels per week based on two shift working (4680 minutes per week). It was accepted that some seasonality in demand would occur and overtime would be worked to accommodate these variations. The main changes to the line were: a pair of new rough turning machines, a replacement heat treat and assembly machine, a new fine turning machine and an extra gear hobbing machine on the starter ring line. The diesel flywheel was almost identical to the petrol except that it was approximately 1kg heavier and needed a different starter ring gear that was bought in finished machined. The manufacturing times for the diesel flywheel were identical to the petrol, although there was a change over time required on the rough turning. The castings for both flywheels and the forging for the ring gear for the petrol were bought in.

The line was initially planned using a flow chart, using work standard and maintenance records to establish the capacity. A simplified version of the flow chart is shown in figure 3.

4.2 Model Validation and Consistency Check

Cusum Method

A cusum chart is recognized as the best statistical process control (SPC) chart for detecting small changes in performance that would not be detected by a Shewhart chart (Owen 1989).

The cusum chart in figure 2 requires two pieces of information to initiate the system; they are the target value and the sample standard deviation. It is suggested that the mean of the initial results be the target value and the sample standard deviation be the s value used to calculate the mask limits. A minimum of 25 readings should be used to calculate these variables.

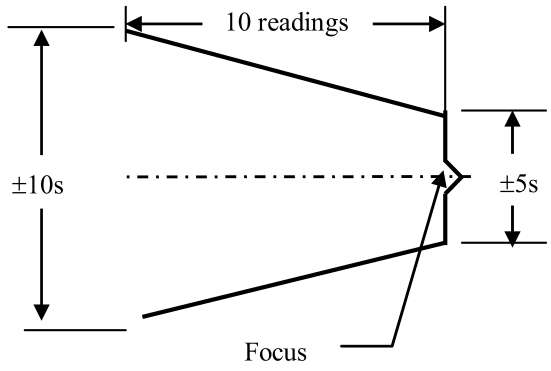


Figure 2. Typical Mask Design

The semi-parabolic mask that is recommended by the authors is equivalent to a 2.65σ action limit and 1.65σ warning limit (BS5703 1981). Other mask designs are available; see BS 5703 (1981) for further details.

The purpose of the stability check is to ensure that the model is valid and that changing the random number stream will not significantly affect the model. If it did then the model would be unstable and the model results cannot be expected to give a solution that would be realistic.

The case study model was checked and proved stable.

4.3 Optimisation and Results

For automatic performance enhancement on the simulation system, bypass balancing backed by theory of constraints within the existing shop floor system was used. Bypass balancing improves throughput and unit operating costs without requiring extra resources to balance the system. Only those resources acting as bottlenecks at the time of the run would be allowed to work overtime. For simplicity, in every adjusted run, the system increments an over-time of 2 hours to the bottleneck workstation. As the company recommends work up to a maximum of two shifts, the maximum allowable overtime work hours is 40 (one additional shift).

In other words, the optimising mechanism which is written in Visual Basic receives the relevant simulation results from Witness, transfers the data to excel, arranges the data; then it assesses their values for performance and takes adjusting actions on the input data for the next run.

The throughput, modified work hours and calculated costs for each run are shown in table 2. The modified costs included a 50% mark up of overtime labour rate. It is assumed that the effect of work in progress due to over-time work is negligible and the variable overhead costs are proportional to the work hours.

The operating cost (Cop) was calculated using the following formula

$$Cop = C_f + (C_{ov} + C_{la})*(W_h/W_{ho}) + C_{ma} * X$$

Where

C_f – Fixed cost per week
C_{ov} – Variable overhead cost per week
C_{la} – Labour cost per week
W_h – Total Work hour
W_{ho}- Initial total work-hour
C_{ma} – Material cost per unit
X – Total parts made

Twenty one simulation-optimisation runs were conducted on the existing resource, as shown in table 2 and figure 4. At the end of runs where one workstation had to work to an over-time of 40 hours per week, the total throughput increased by 70% and the operating cost per flywheel was reduced by 21%. There was an average increase of 17 flywheels in every two-hour over-time work as compared to under 9 flywheels in the existing system.

As there often is a fluctuation in demand, the table (and the graph) was intended to be used for selecting the best combination that would fit the production schedule at hand.

It is worth mentioning that a similar expert mechanism was applied to a more complex real system of a reputable industry (under confidential agreement) and resulted in the same optimum output as the Optimiser of Lanner Group but with under 10% of the time taken by Optimizer. The application dealt with optimising throughput and operating expenses under constraints of resources and delivery time.

5 CONCLUSION

The paper has demonstrated the basic concept of how an expert mechanism can be used to assist in automatic result interpretation and system optimisation of simulation processes in manufacturing. It also demonstrated that the model could be checked for statistical accuracy automatically. A simple model was used and the results showed good consistency. A more complex model would definitely require a more comprehensive knowledge base.

The more comprehensive the knowledge base in the expert mechanism is, the more robust it would be and the more complex systems it could handle. Further work on the project will include incorporating more shop floor manufacturing performance measures by building a more comprehensive knowledge base.

REFERENCES

Agarwal and Babu (1994), A Simulation Study of Performance of MRP based Systems under Interaction Effects (Taylor & Francis).

Alam, F.M, McNaught,K.R., and Ringrose, T.J., 2002, Investigating Appropriate Experimental Designs for Neural Network Simulation Metamodels, Operations Research Simulation Study Group - Workshop Proceedings, Birmingham, 20-21 march (Operational Research Society), pp.45-54.

April, J., Glover, F., Kelly, J.P., Laguna, M., 2003, Practical Introduction to Simulation Optimization, Proceedings of the 2003 Winter Simulation Conference, New Orleans, Louisiana, 7-10 December, www.informs-cs.org/wsc03papers.010.pdf, pp.71-78.

Avello, E A, Baesler F F, Moraga R J, A, 2004, Metaheuristic Based Simulation Annealing for Solving Multiple-Objective Problems in Simulation Optimisation, Proceedings of the 2004 Winter Simulation Conference, December 5-8, Washington DC, USA, pp 508-513

Azadivar, F., 1992, Tutorial on Simulation Optimisation , Proceedings of the 1992 Winter Simulation Conference, Arlington, eds Swain J, Glodman D, Crain R C, Wilson J R, (New Jersey: IEEE), pp.198-204

Baccouche, M.,Boukachour, J., and Benabdelhafid, A., 2003, Genetic Algorithm for Scheduling Aircraft Landing, , Proceedings of Industrial Simulation Conference 2003, Valencia, Spain (Belgium:Eurosis Publication), pp.418-422

Barretto, M.R.P., Chwif, L., Eldabi, T., Paul, J.R., 1999, Simulation Optimisation with the Linear move and Exchange move Optimisation Algorithm, Proceedings of the 1999 winter Simulation Conference, Phoenix, Arizona, 4-8 December, pp. 806-811.

Box, G.E.P., and Draper, N.R., 1987, Emprical Model Building and Response Surfaces (New York: John Wiley & Sons)

British Standards, 1981, BS 5703 Part 3, Data analysis and quality control using cusum techniques.

Brusha & Franek, 2003, Inducing Parameters of decision tree for expert system shell MCESE by Genetic Algorithm, Proceedings of Industrial Simulation Conference (ISC2003), Valencia, Spain (Belgium:Eurosis publication), pp.48-52

Debuse, J.C.W, Rayward-Smith, V.J., Smith, J.D., 1999, Parameter Optimisation for a Discrete Event Simulation, Journal of Computers and Industrial Engineering, 37, 181-184.

Glover & Laguna, 1997, Tabu Search (Boston: Kluwer Academic Publishers)

Kilmer, R.A., Smith, A.E., Shuman, L.J., 1997, An Emergency Department Simulation and a Neural Network Metamodel, Journal of the Society for Health Systems, 5, 63-79.

Lanner Group, 2008, OLE Automation in Witness, Witness 2004 on-line, Lanner Group.

Law, M.A., & Kelton, D.W., 2000, Simulation Modelling and Analysis (London: McGraw Hill).

Lyu & Gunasekaran, 1997, An intelligent Simulation Model to Evaluate Scheduling Strategies in a steel Company, International Journal of Systems Science, 28(6), 611-616.

Mebrahtu H, Walker R W, Dionysopoulos T, Mileham T, 2009, Manufacturing Performance Optimization: The Simulation -Expert Mechanism Approach, Proc. IMechE, Part B: J. Engineering Manufacture, 2009, 223(B12), 1625-1634

Nicolai R P, Dekker R, Piersma N, Van-Oortmarssen G J, 2004, Automated Response Surface Methodology for Stochastic Optimisation Models with Unknown Variance, Proceedings of the 2004 Winter Simulation Conference, December 5-8, Washington DC, USA, pp 491-499

Owen M, 1989, SPC and continuous improvement (Bedford:IFS).

Randel, L., Holst, L., Bolmsjo, G., 1999, Concurrent Development of Large Discrete-Event Simulation Models, World Manufacturing Congress, ICSC.

Spall, J. C., 2003, Introduction to stochastic search and optimization (New Jersey: John Wiley & Sons).

AUTHOR BIOGRAPHIES

Dr HABTOM MEBRAHTU is a Principal Lecturer and Programme Leader of Design & Engineering in the Faculty of Science and Technology at the Anglia Ruskin University, UK. He received his PhD degree from the University of Bath, Mechanical Engineering Department, in the field of Manufacturing Simulation and Optimisation. He is a Chartered Engineer, Fellow member of the HEA and a member of the IET. He has recently worked with the simulation team of the Ford powertrain operations, Manufacturing Engineering division at Dunton, UK where he modelled engine assembly lines. His areas of interest include performance optimisation with simulation, and web-based simulation. He is a referee for EUROSIS, Manufacturing Simulation area.

DR ROBERT W WALKER C Eng MIET is Principal Lecturer and Deputy Head of the Department of Computing and Technology in the Faculty of Science and Technology at Anglia Ruskin University. His research field is the novel application of control charts for the analysis of experimental data. This includes the testing for stability the results from discrete event simulations, finite element analysis results in the field of bioengineering as well as industrial experiments. He has a number of publications including refereed conferences and journals. He has extensive industrial experience including Quality Control Engineer at the Ford Motor Company. He is referee for the European Simulation Conference EUROSIS, Automotive Pathway.

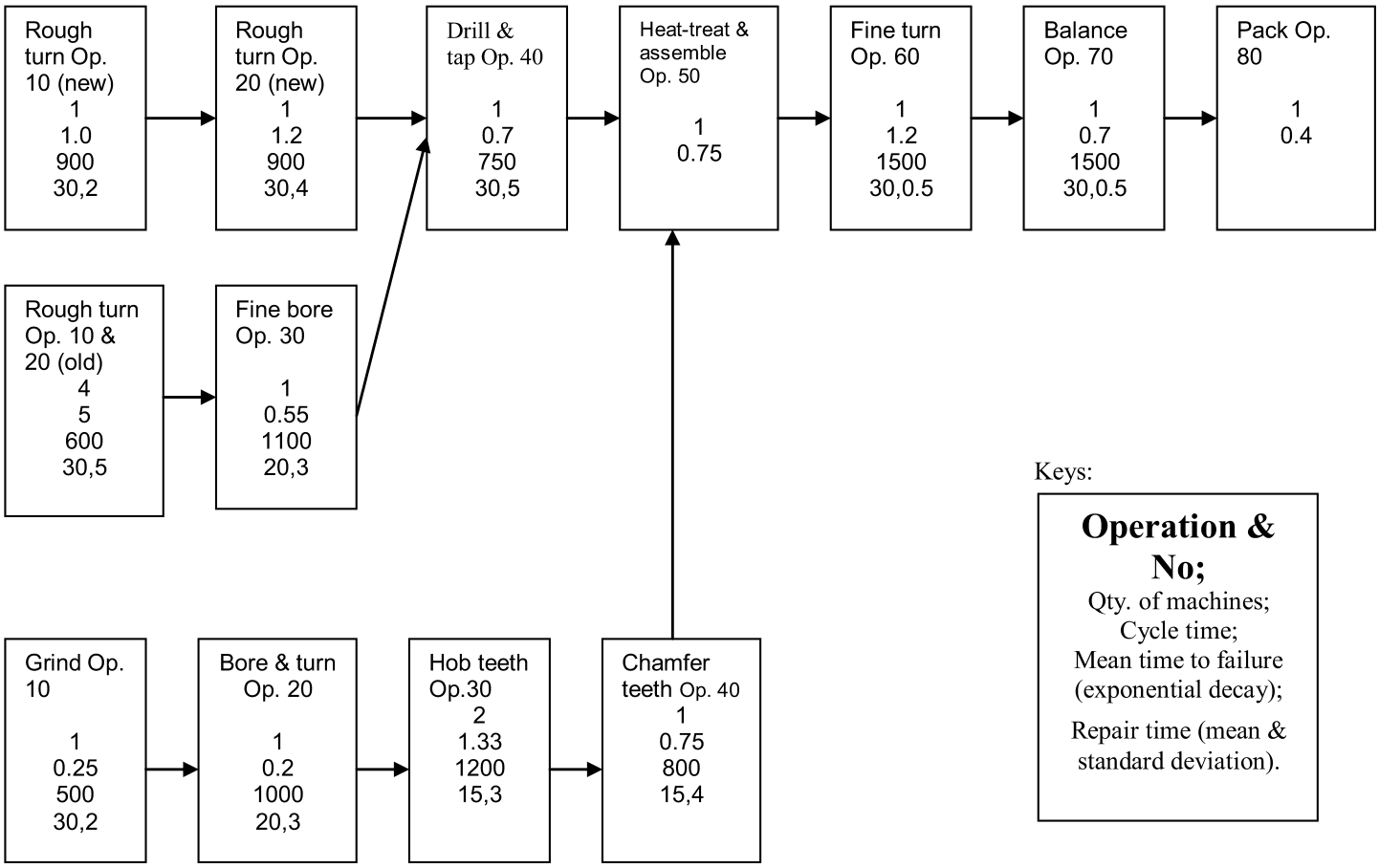


Figure 3. Flow Chart for the Flywheel & Starter Ring Manufacture

Total runs	21																					
Run No.		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Parts made, X		2248	2334	2427	2484	2538	2607	2743	2822	2913	3005	3098	3206	3294	3291	3396	3423	3549	3667	3713	3792	3824
	Modified Data					Work hours																
RoughturnOld	5.00	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
RoughturnNew	0.48	40	40	42	44	46	48	50	52	54	56	58	60	60	62	64	64	66	68	70	70	70
R_turnNew2	0.45	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	76	78
Finebore	0.55	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Drilltap	0.25	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80
Hassy	0.48	40	40	40	40	42	42	44	46	48	48	50	52	52	52	54	54	56	56	58	58	58
Grind	0.25	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
BoreTurn	1.20	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Hob	16.24	40	40	40	40	40	40	42	42	44	46	46	48	50	50	50	50	52	54	54	56	56
Chamfer	0.46	40	40	40	40	42	42	44	46	48	48	50	52	52	52	54	54	56	58	58	58	60
Fineturn	0.94	40	40	40	40	40	40	40	40	40	40	42	42	44	44	44	46	46	48	48	50	50
Balance	0.47	40	40	40	40	40	40	42	44	44	46	48	48	50	50	52	52	52	54	54	56	56
Pack	0.40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Total work_hour/week		520	524	530	536	546	552	566	578	590	600	614	626	636	642	654	660	672	686	694	702	708
Op_cost-flat (£)		21144	21444	21785	22018	22284	22553	23107	23468	23866	24246	24670	25119	25487	25540	25980	26123	26625	27125	27346	27666	27824
Cost/part (£)		9.4	9.2	9.0	8.9	8.8	8.7	8.4	8.3	8.2	8.1	8.0	7.8	7.7	7.8	7.7	7.6	7.5	7.4	7.4	7.3	7.3
Op_cost-mod (£)		21144	21450	21802	22046	22329	22609	23186	23569	23987	24384	24833	25302	25687	25751	26211	26365	26889	27412	27647	27981	28150
Cost/part (£)		9.4	9.2	9.0	8.9	8.8	8.7	8.5	8.4	8.2	8.1	8.0	7.9	7.8	7.8	7.7	7.7	7.6	7.5	7.4	7.4	7.4

Fixed cost/week, Cf (£)	9000
Overhead cost/week, Cov (£)	3600
Labour cost/week, Cla (£)	1800
Material cost/part, Cma (£)	3

Operating Cost(flat), Cop = Cf + (Cov + Cla)*(Wh/Who) + Cma * X

Operating Cost(mod), Cop = Cf + (Cov + Cla)*(Wh/Who) + Cma * X + (0.5*Cla*(Wh - Who)/Who)

Table 2. Throughput, Modified Data, and Operating Costs

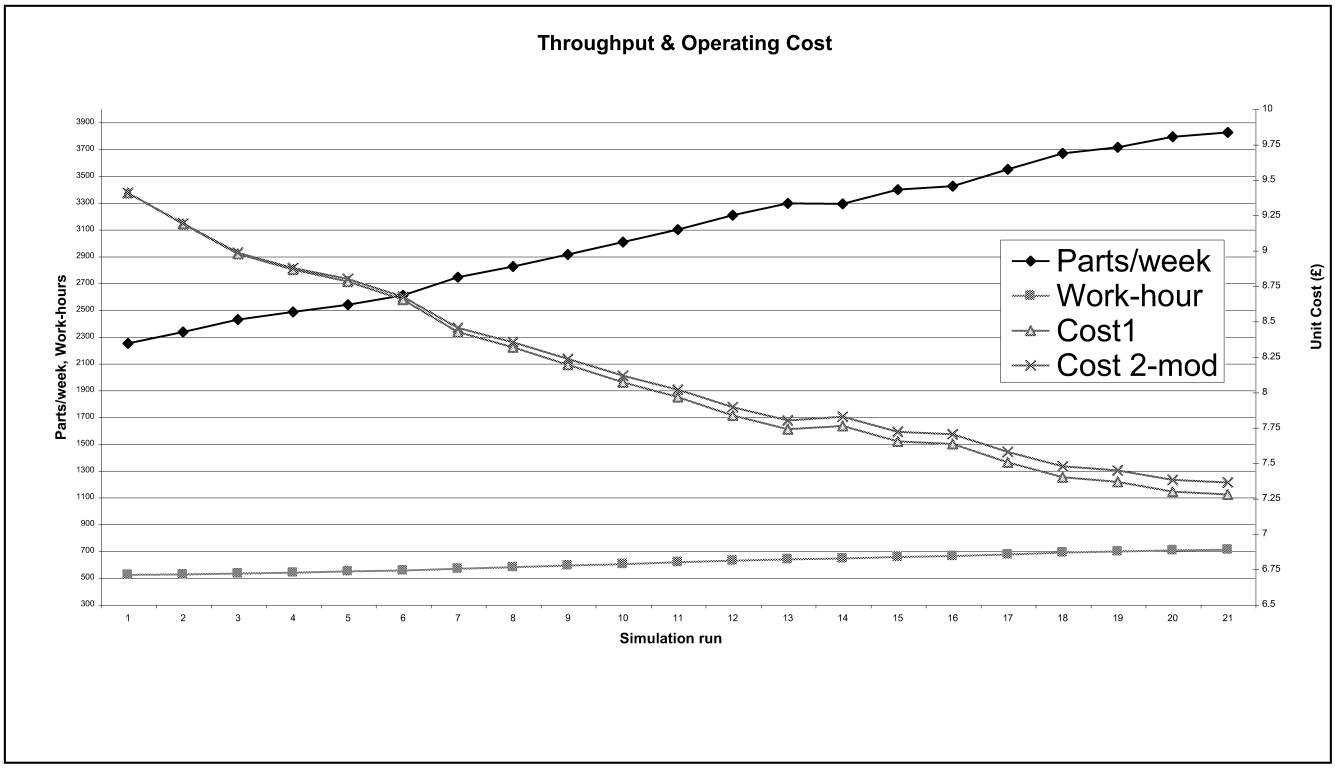


Figure 4. Throughput and Operating Costs.

A DESIGN METHOD FOR PARTS PICKING ZONES IN A MANUFACTURING ENVIRONMENT

Klaas Peerlinck, Tim Govaert and Hendrik Van Landeghem
University of Ghent
Department of Industrial Management
Technologiepark 903
B-9052 Zwijnaarde, Belgium
E-mail: Tim.Govaert@ugent.be

KEYWORDS

Order picking, Order picking system (OPS) design, supermarket, kitting

ABSTRACT

This paper describes a method for the design of an order picking system in a manufacturing environment. Unlike order picking systems in warehouses, there is almost no literature available concerning order picking systems in a manufacturing environment. We start by defining the needed input parameters, followed by a parts classification method. This leads to the calculation of order specifications (order lines, volume, weight,…). The needed throughput, available floor space and associated costs then define the most appropriate order picking system under the given circumstances.

INTRODUCTION

Order picking is the process of retrieving products from a predesignated storage medium for multiple customers (Choe and Sharp, 1991). Each customer order is characterized by a number of order lines and the amount needed per order line. The customers, in a manufacturing environment, are the different workstations across the factory floor. The design of an order picking area requires attention, as the order picking process is considered to be the most labour-intensive and costly activity (de Koster et al., 2007; Ten Hompel and Schmidt, 2005). Unfortunately there is little literature to be found concerning the overall design procedure of an order picking area. The literature that has been written focuses almost always on distribution warehouses (Baker et al., 2009; de Koster et al., 2007; Goetschalckx et al., 2001; Gu et al., 2007; Van Den Berg et al, 1999) and not on manufacturing environments and their specific needs. In manufacturing, there is a move to smaller lot-sizes, point-of-use delivery, order and product customisation, and cycle time reductions (de Koster et al., 2007). To allow this change, parts picking supermarkets are currently seen as the answer. In this paper we will present a method for the design of an order picking system in a manufacturing environment.

LITERATURE REVIEW

According to Dallari et al. (2008) most order picking systems are designed on the basis of insights, experience and, sometimes, on a detailed simulation. Most of the literature considers a specific order picking system in which they try to optimise some order picking policies like storage assignment, routing, batching and zoning.
Yoon and Sharp (1996) were the first to consider the interdependent relationships among different functional areas in an order picking system.
According to Yoon and Sharp (1996) the general structure of an OPS consists of eight functional areas:

- receiving area
- pallet reserve area
- case pick area
- item pick area
- sorting area A and B
- unitizing area
- shipping area

Although the model presented by Yoon and Sharp is constructed for distribution warehouses, several of its characteristics can be implemented in manufacturing systems. The OPS design procedure developed by Yoon and Sharp is subdivided in three main phases:

- input stage
- selection stage
- evaluation stage

Dallari et al. (2008) added a fourth phase (detail stage) and made some adjustments in the previous stages but the principles remain the same.
There has been a lot of literature on order picking methods and operational policies. Figure 1 (de Koster et al., 2007) gives an overview of the different order-picking methods and distinguishes between order-picking methods executed by humans or by machines. In manufacturing environments the more limited scope (volume, number of parts etc.) as well as the more stringent cost controls make the automated methods rather the exception. Many of the parts-to-picker systems became the exception as well and it is an open question whether this has any factual base. This is an inspiring issue for this research.

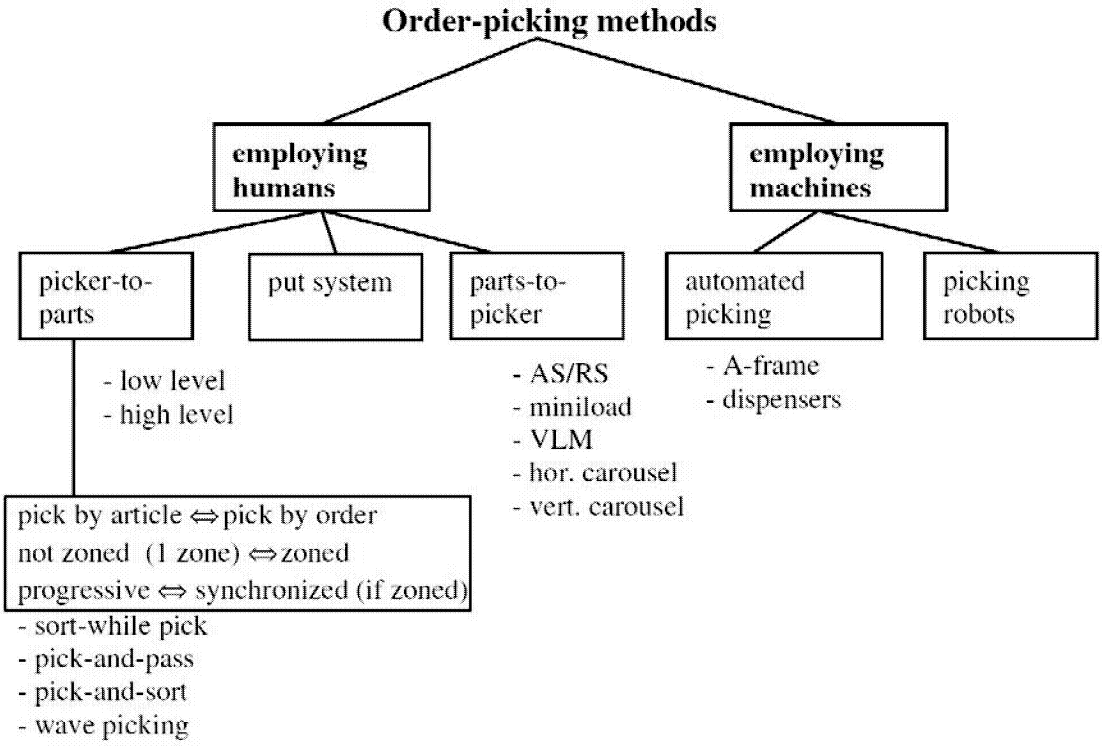


Figure 1: Classification of order-picking methods (de Koster et al., 2005)

DESCRIPTION OF THE DESIGN METHOD

The main goal of our method is to design an order picking system for a manufacturing environment. In our method we will only consider the design of the case/item pick area. Automated picking (picking by employing machines) will not be considered as it has little use in a manufacturing system: automated picking is mostly used for regular shaped boxes with high volumes (f.e. pharmaceutical industry). In a manufacturing environment all parts need to be unpacked to allow assembly and thus handled in their net irregular shape. By using the method one can render the performance of the existing order-picking systems that employ humans.

In figure 2, a flowchart is given of the different steps in the method. In the following paragraphs this will be explained more clearly. The whole method has been implemented in Excel.

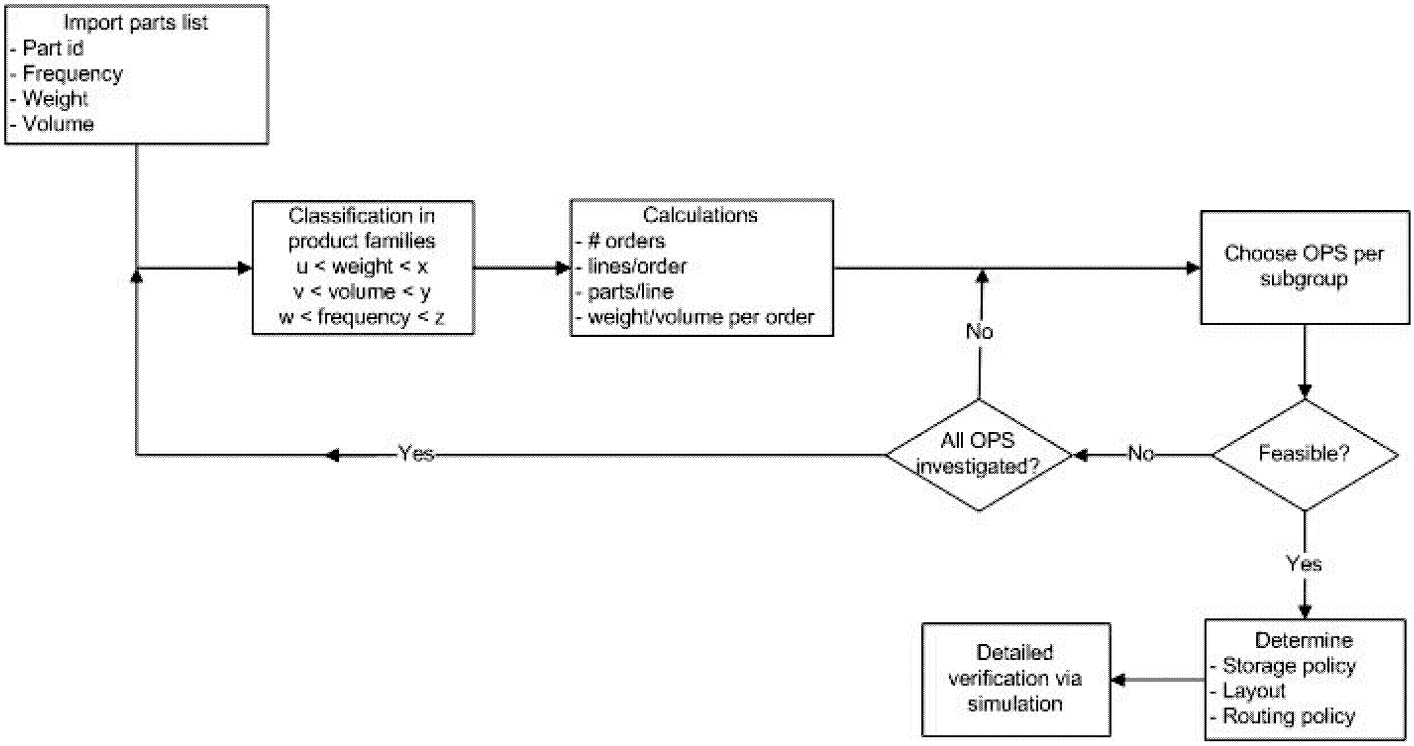


Figure 2: Flowchart of the OPS design method

Step 1: Input

The first step is to import a parts list. The following product parameters need to be known:

- Part-id
- Frequency
- Parts per product
- Weight
- Volume

A part-id is needed to identify unique Stock Keeping Unit's (SKU's) throughout the design procedure. With frequency we mean the percentage of finished products that contain this specific part. Parts per product is the amount of a specific part that is needed for the assembly of one product. The weight and volume of each individual part is needed to group parts in homogeneous product families. This classification can be extended with additional attributes (safety requirements, bulky parts indicators etc.) that enhance the classification process.
In our case study, situated in the automotive industry, some parts had to be handled differently because of safety regulations and sensitivity to damage.

Step 2: Classification

Due to the high amount and variety of parts, there are very different needs. There is a tremendous difference between how big parts are manipulated versus the handling of small parts. Also the weight plays an important role as parts with a certain weight cannot be handled without a hoist. Therefore we need product groups.
The classification structure implemented in the method makes it easy to group parts with common characteristics. Four classes have been introduced to classify the weight, ranging from heavy to light parts. Volume is also divided into four classes ranging from very big to small parts (nuts, bolts etc.). The ranges can be adjusted by the user and it is possible to add or remove classes. Frequency can be used to determine which parts are common, variant and rare. The limits must be set by the user. The user of the method can now select the range of parts one wants to investigate, taking into account the weight, volume and frequency.
In our case study, one of the weight cut-offs was set at 12kg, as this is a limit for manual handling. The volume restrictions were defined by the standard emballages the company was using (euro boxes, small boxes etc)

Step 3: Order specifications

Based on the classification made in step 2 and the order interval determined by the user, the following order characteristics per customer will be calculated:

- Maximum and average number of order lines/order
- Number of picks
- Volume per order
- Weight per order

A customer can be seen as a workstation or anything that orders parts for the assembly of a product.
In the automotive case study we defined a customer as a workstation in the same way they exist today at the assembly line. As an extension it is also possible to combine several workstation and treat them as one customer. In this way we can pick parts for several workstation at once.

Step 4: Choosing an Order Picking System (OPS)

There exist multiple order-picking systems. Picker-to-parts and parts-to-picker are the two major groups. The choice between a picker-to-parts or parts-to-picker OPS is very complex and depends on many factors. Major issues affecting the design of an order picking system are material properties, economic constraints, environmental constraints, system requirements, operating strategies and transaction data (Yoon and Sharp, 1996). The information obtained in step 3 and the product characteristics (weight, volume and frequency) will be used to identify a feasible OPS for each product family. If there is no feasible solution for a specific product family other options have to be considered (f.e. outsourcing in order to save space) or adjustments must be made in the classification step.

3.4.1 Picker-to-parts

We can distinguish between low- and high-level order picking. In high-level order picking high storage racks are employed. Parts are picked by using a lifting order-pick truck or crane. In low-level order-picking all parts are reachable from the floor level. For low level order picking several types of storage racks are reviewed:

- Shelve racks
- Shelve racks with floors
- Gravity racks
- Moveable racks

3.4.2 Parts-to-picker

For parts-to-picker order picking systems we consider the use of horizontal and vertical carousels, a vertical lift module (VLM) and an Automated Storage & Retrieval System (AS/RS).
R.D. Meller et al. (2006) developed a calculation tool for horizontal and vertical carousels and a VLM. In this method the same equipment parameters are used.
The information needed to calculate the results concerns capacity characteristics, operational characteristics, order info and costs. Table 1 gives an overview of the needed information for a horizontal carousel.

Table 1: Equipment parameters based on Meller (2006)

Horizontal Carousel	
# Shelves	2200
Shelves/bin	6
Shelf depth	0,6 m
Shelf width	0,6 m
Shelf height	0,45 m
Speed	30 m/min
Walking time btw units	2 sec
Start/Stop time	2 sec
Pick time	5 sec
Orders/batch	14
Lines/order	8
Items/line	5
Setup Time/batch	1 min
# Carousels/pod	4
# Pods (pickers)	1
Costs	
Price per carousel	€
Price per m²	€

RESULTS

Based on the characteristics given by the suppliers of each order picking system, the following variables are calculated:

- throughput,
- required floor space
- associated costs

Based on these figures it is possible to determine for each product family what the most interesting order picking system is. This solution will be optimal only for the predefined product families. Several possible ranges of the classification parameters, resulting in different product families, will need to be investigated in order to achieve the overall optimal solution.

FUTURE RESEARCH

The method introduced in this paper is only a first step into the design of an order picking system in a manufacturing environment. The final goal of the research is to set up a classification diagram that can immediately tell you the optimal OPS for a certain range of parts. This range is defined by the weight, volume and frequency. In this way these diagrams can be of high value for the vast amount of companies that are currently designing their parts picking areas.

There is also research needed in the field of order picking systems and the parameters that influence the decisions.
After the selection of the order picking systems for all product families a next step can be implemented namely the analysis of what is the best storage and routing policy based on the chosen order picking systems and classifications. There is extensive literature on this subject.
As a next step in the method, an immersive 3D environment will be created, using standard building blocks in order to rapidly simulate and evaluate the designed order picking systems.
Validation of the method can be achieved by running the model with data from multiple manufacturing facilities and comparing their solutions with the calculated solution. By using these results we can further optimize the method.

REFERENCES

P. Baker and M. Canessa, "Warehouse design: A structured approach", *European Journal of Operational Research*, Vol. 193, 2009, pp. 425–436.

J.J. Bartholdi III and S.T. Hackman, *Warehouse and distribution science*, v0.90, The Supply Chain and Logistics Institute School of Industrial and Systems Engineering Georgia Institute of Technology Atlanta, 2009, 292 pp.

K. Choe, G.P. Sharp, Small parts order picking: design and operation, School of Industrial and Systems Engineering Georgia Institute of Technology, 1991, pp. 41

F. Dallari; G. Marchet and M. Melacini, "Design of order picking system", *International Journal of Advanced Manufacturing Technology*, Vol. 42, June 2008, pp. 1-12.

R. de Koster, T. Le-Duc and K.J. Roodbergen, "Design and control of warehouse order picking: A literature review", *European Journal of Operational Research*, Vol. 182, August 2005, pp. 481-501.

M. Goetschalckx et al.., Development of a design methodology for warehousing systems: hierarchical framework, School of Industrial and Systems Engineering Georgia Institute of Technology Atlanta, 2001, pp. 6.

J. Gu, M. Goetschalckx and L.F. McGinnis, "Research on warehouse operation: A comprehensive review", *European Journal of Operational Research*, Vol. 177, 2007, pp. 1-21.

R.D. Meller, J.F. Klote and E. Lilaj, Design of carousel order picking, MS Excel spreadsheet, VirginiaTech, 2006

J.P. Van Den Berg, W.H.M. Zijm, Models for warehouse management: Classification and examples, International Journal of Production Economics, Vol. 59, 1999, pp. 519-528.

C.S. Yoon and G.P. Sharp, "A structured procedure for analysis and design of order pick systems", *IIE Transactions*, Vol. 28, No. 5, 1996, pp. 379-389.

VIRTUAL MANUFACTURING

A COMPREHENSIVE 3-D SIMULATION SYSTEM FOR THE VIRTUAL PRODUCTION

Juergen Rossmann
Institute of Man-Machine Interaction
RWTH Aachen University
Ahornstr. 55
52074 Aachen, Germany
rossmann@mmi.rwth-aachen.de

Roland Wischnewski, Oliver Stern
RIF e.V.
Department Robot Technology
Joseph-von-Fraunhofer-Str. 20
44227 Dortmund, Germany
{wischnewski,stern}@rt.rif-ev.de

KEYWORDS
Digital Factory, Virtual Production, 3-D Simulation, Simulation System, Robot Simulation, PLC Simulation

ABSTRACT

Today, many industrial companies use the virtual production to plan and optimize their automated manufacturing plants. In this paper, we describe a 3-D simulation system that can be used throughout the whole lifecycle of a plant including mechanical construction, electrical construction, controller programming, and commissioning. We present the integration of powerful modeling and simulation concepts and expand these with new usability methods to gain a software tool that integrates into modern industrial engineering processes. Selected applications prove the practical use of the simulation system even in virtual reality scenarios.

INTRODUCTION

During the last decades, the life cycle of products of any kind has changed a lot. More and more, customers demand high quality up-to-date products at low prices. These demands result in new requirements for the employed manufacturing plants. Engineers in industry try to cope with these requirements by increasing the degree of automation of the plants. In most cases, this strategy seems to work but also results in an ever growing complexity of the plants. To reduce the risk of non-functioning plants at the start-of-production, many companies apply means of the Digital Factory in the planning stage. The traditional course of the engineering process is accompanied and supported by computer based methods like the virtual production.

VIRTUAL PRODUCTION

In virtual production, a digital computer model of a real manufacturing plant is created to allow for technical experiments without the need to allocate or even have the real plant. Figure 1 shows the most important stages of the typical course of the virtual production for automated manufacturing plants (Rossmann et al. 2007). In every stage, data of the previous steps is used and new data is added to the model. At first, mechanical engineers "invent" a process to solve the given production task, think about cycle times, and choose components from manufacturer's catalogues. These components are assembled in a 3-D simulation system and result in a first functional model. After this, electrical engineers define wiring diagrams and thus add information to the model. Now, robot and controller programmers can start their work early in the planning process without actually having a real plant. All their work is done controlling just the digital model of the plant. Resulting programs are used to control the real plant later without any changes. This can potentially save a lot of time up to the start-of-production. However, most problems in setting up a manufacturing plant are not revealed until the commissioning stage. To minimize the problems in this stage, virtual production offers the possibility to execute a simulation-based virtual commissioning using all original controller programs. Every problem that is identified and solved in this stage will shorten the time for the real commissioning later: engineers get better plants earlier.

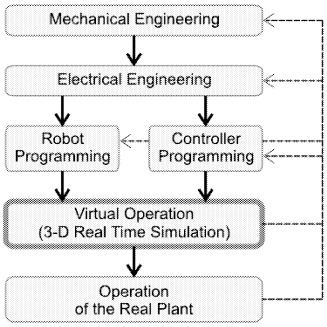


Figure 1: Typical course of the virtual production for automated manufacturing plants

The described course of the virtual production imposes many requirements upon the software tool used. Each relevant aspect of the real manufacturing plant has to be modeled. The simulation system must support robots, sensors, transport devices, and arbitrary kinematic devices. Every component must look and behave like in reality and must have a reality compliant electrical interface to be able to work with original controller programs. In this paper, we will present a 3-D simulation system that supports all stages of the virtual production.

THE 3-D SIMULATION SYSTEM

Some of the concepts making up the 3-D simulation system date from the early 1990s but are designed sufficiently

general that they are still suitable for the integration with new modern methods. In the following chapters, we will describe the most important concepts of the 3-D simulation system.

Model Representation

The simulation model used must comprise both geometric and functional aspects. To describe all environment data, we use the Systematic Environment Decomposition Data (SEDD) for the model representation and the XML-like Geometric Modeling Language (GEMOL) to store models in a file system. Both concepts are first presented in (Rossmann 1993). Figure 2 shows how a 3-D simulation *environment* is hierarchically decomposed into the SEDD elements *objects*, *sections*, *hulls*, *vertices*, and *facets*.

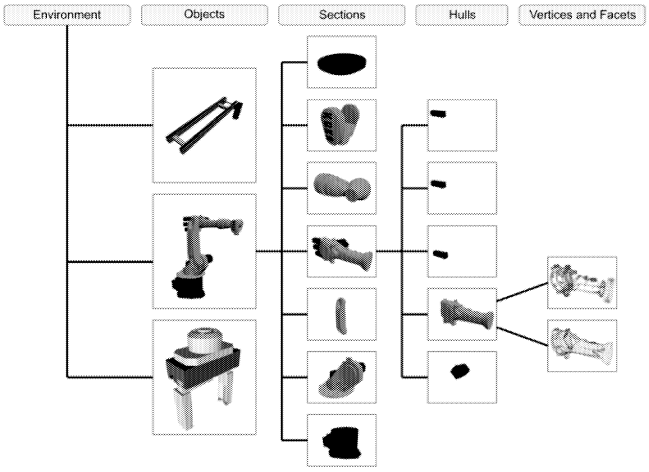


Figure 2: Hierarchical decomposition of 3-D simulation models using the SEDD

An *object* is a logical unit of the real world like a belt conveyor, a robot, or a robot gripper. Such an object is further decomposed into *sections* which represent rigid bodies. There may be a degree-of-freedom, i. e. a linear or a radial axis, between sections but not within one section. For a robot, the sections correspond to the separate axis bodies. A section itself consists of any number of hulls. A *hull* is a purely geometric element. Its surface is described in a 2½-D fashion by planar *facets* in between *vertices*. Moreover, *materials* with optical, electric, magnetic, and acoustic parameters can be assigned to each hull. Every SEDD element's pose in 3-D space is given by a 6DOF homogenous transformation matrix (a frame) relative to the superordinate element.

Functional properties are added to the model on several levels (Rossmann 1993). Every object has a unique type denoting its function e. g. behaving like a robot, a robot gripper, a part magazine, a clickable button etc. Electrical inputs and outputs (I/Os) – digital and analog – are also defined at object level. Electrical connections can be established directly between any output and any input. For the definition of arbitrary kinematic chains, a dependent section's *grip point* can be attached to another section's *gripper point*. If the latter moves, the dependent section and its whole object will move accordingly. Furthermore, the SEDD allows for the integration of whole environment models (*libraries*) into a superordinate environment.

Simulation System Architecture

Figure 3 shows the architecture of the simulation system (Hypki 2008). The central coordinating instance, the *kernel*, keeps and handles the SEDD-based environment model, updates I/O connections, and provides basic mechanisms. Moreover, the kernel offers the fundamental user interface of the simulation system and the multi instance 3-D render engine. The renderer will even be capable of creating stereo images of a scene if the present graphics adapter supports quad buffered OpenGL as, for example, the NVIDIA Quadro series does. Concerning simulation, the kernel employs time-based simulation with a fixed time step. Opposed to many other simulation systems, the visualization of the scene can be updated after every single time step to achieve an "online visualization" during simulation – always showing the current model state and allowing for user interaction like e. g. pressing buttons.

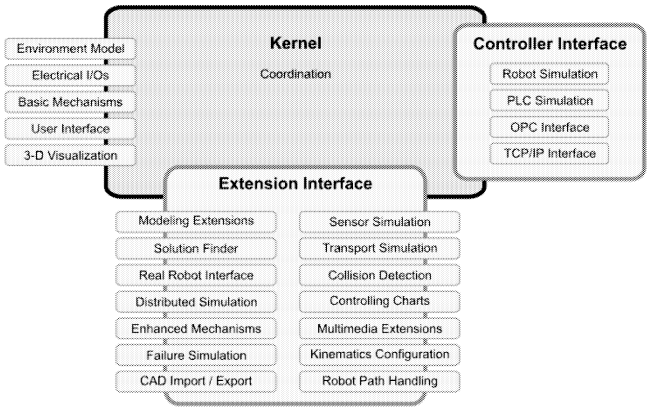


Figure 3: Architecture of the 3-D simulation system

The relation between simulated time and real time is handled by a controller-based real time adjustment (Freund et al. 2002). Simulation time can either elapse synchronously to real time or as fast as possible. Even time scaling is possible to make the simulation run faster or slower than real time by a fixed factor.

The system's architecture provides two concepts to add functionality to the system – the controller interface and the extension interface (Hypki 2008).

The *controller interface* allows for the internal integration and the external coupling of controllers, e. g. robot controllers or programmable logic controllers (PLCs). By doing so, a controller determines the behavior of an SEDD object.

The controller interface is based on a client / server-architecture in which the kernel as the one and only client requests data from different controllers which act as servers. The controller interface is kept as small as possible

to remain capable of communicating with the controllers in real time.
The *extension interface* is a more comprehensive interface allowing for adding any kind of functionality to the simulation kernel. The interface is based on two features. First, the kernel sends cyclic event notifications to all extension modules to keep them updated about the current SEDD state. Second, the kernel and / or the extension modules can communicate via extension messages with variable parameters at any time. To keep the handling overhead low for many extension modules, the modules can register for certain messages and thus remain unaffected by the others. In general, extension modules can extend the environment model, extend the time-based simulation calculations, add to the 3-D visualization, and extend the user interface.
Finally, the controller interface and the extension interface offer the possibility to add additional modules dynamically. The separation of enhanced functionality from the kernel supports a fast and stable software development process and keeps the maintenance effort low. In the following, we will present some modules that have been integrated using the two interfaces.

CONTROLLER MODULES

Many automated manufacturing plants employ robots. Because of this, it is crucial for a simulation system that robot controllers can be integrated. The controller interface allows both the coupling of external robot controller hardware and the incorporation of internal robot controller software. As robot manufacturers do not reveal their original controller algorithms, the independent PC-based robot controller PCROB (Van der Valk 1995) has been chosen and integrated into the simulation system via the controller interface (Freund et al. 1993, Keibel 2003). The robot controller executes all kinematic calculations and is programmed in the standardized low-level language IRDATA (Industrial Robot Data).
To ease programming, compilers have been added to translate the high-level languages IRL (Industrial Robot Language), Mitsubishi MRL, and Puma VAL-II (Freund 1991) into IRDATA code. To unify the creation of new compilers, a compiler framework (Freund et al. 2001b) has been designed and used to translate the languages KUKA KRL, ABB RAPID, Mitsubishi Melfa-Basic-IV, Adept V+, and Stäubli VAL3 into IRDATA code. Figure 4 shows how robot programs written in native robot languages can be used to control either a real or a simulated robot. The dotted arrows indicate that the robot controller PCROB can even calculate axis values to be sent to real robots. The native language programs can be written within the simulation system – syntax highlighting supports the programmer's work. For example, figure 5 down right shows a program window with syntax highlighting for a Melfa-Basic program.

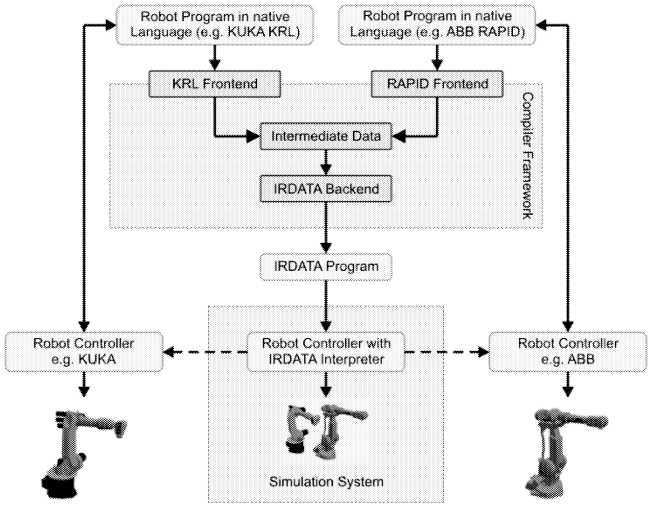


Figure 4: The use of robot programs written in native robot languages to control real and simulated robots

Besides robot controllers, the controller interface is also used to couple OPC servers and clients (Bauer 2003), PLC program interpreters for Siemens STEP5 and STEP7 (Freund et al. 2001a), the PLC simulator Siemens PLCSIM, and the Stäubli CS8 Emulator. Any controller coupled via the controller interface can influence the mechanical and electrical state of corresponding SEDD objects during simulation.

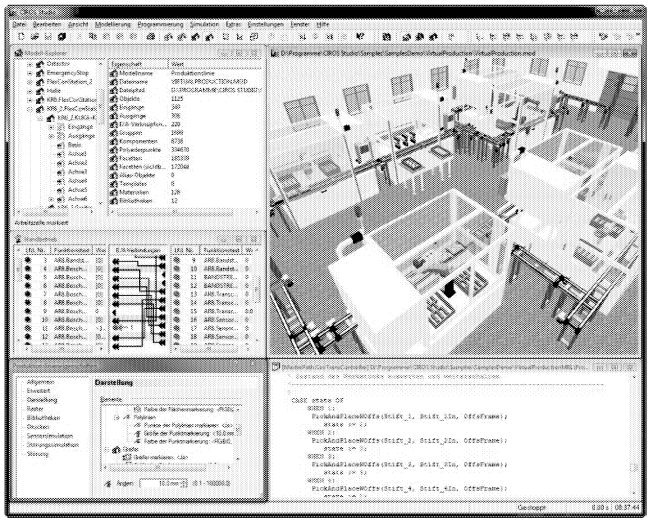


Figure 5: User interface of the simulation system showing Model Explorer, wiring assistant, property editor, scene view and program window (top left to down right)

EXTENSION MODULES

Several extension modules have been added to the simulation system via the extension interface. (Pensky 2004) describes modeling extensions for comfortable access to the environment model. The *Model Explorer* is an editable tree structure that maps the hierarchy of the SEDD elements. A property editor allows modification of

currently selected elements. Figure 5 shows both Model Explorer and property editor. Furthermore, the modeling extensions allow for the creation of model libraries which enable the reuse of approved models.

Sensor Simulation

As almost every automated manufacturing plant contains lots of different sensors, a simulation system for the virtual production must also consider these. The sensor simulation integrated in the simulation system is described in (Uthoff 1998). Sensors are classified considering the measuring principle, the type of the electrical outputs, the measuring range, and the physical measuring method where applicable. Concerning measuring principles, the concept distinguishes between tactile, inductive, capacitive, optical (light barrier, laser scanner, color detector, barcode reader, camera), and acoustic methods. The state of a sensor is output via the electrical digital and / or analog outputs of the corresponding SEDD sensor object. For reasons of calculation performance, a modeled sensor's measuring range is never identical to its real complex shape but an approximation. Such approximation is achieved by a point, a line, a fan of lines (circle-sector-shaped or rectangle-shaped), or a cone. Measuring ranges can be parameterized concerning size and the density of the approximating lines. Figure 6 shows models of a camera, a light barrier, and a laser scanner. Visualization of the measuring ranges helps the engineer to comprehend the model. If a sensor detects an object, the approximating lines that touch the object will switch their color to support detailed analysis. The sensor simulation also considers physical parameters e. g. optical sensors only detect objects which have a material with certain optical properties assigned. During simulation, a hierarchical system of different bounding volumes is used to speed up geometric intersection algorithms. Additionally, (Wischnewski 2007) describes concepts to consider transceiver / transponder-systems.

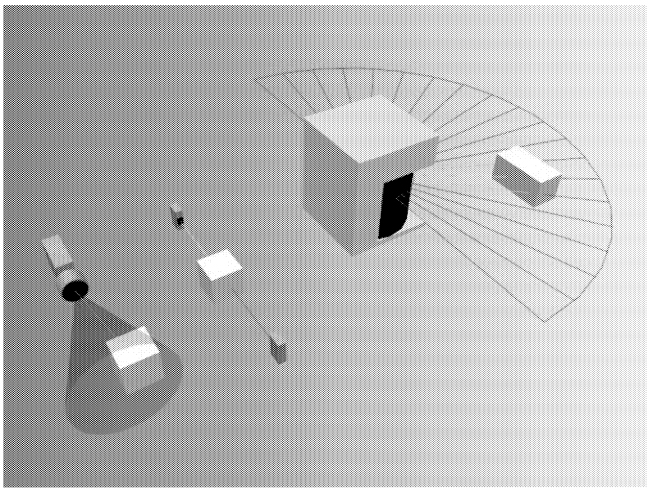


Figure 6: Examples of different sensors – a camera for object tracking, a light barrier, and a sector laser scanner

Transport Simulation

Another extension to the simulation system is the transport simulation for carrier-based and track-bound transport systems presented in (Wischnewski 2007). This approach is not intended to execute event-based time lapse analysis but to allow for simulating transport devices with original controller software in real time. For example, figure 7 shows the model of a branching point of a belt conveyor system. The tracks on which carriers move are defined by *nodes* and *segments* between these nodes. Stoppers and other devices are modeled by nodes that are attached to segments. A carrier is represented by an *anchor* that moves along the segments. *Connections* of segments make up entire tracks. Moreover, nodes and segments may be attached to any other SEDD elements and thus may move – e. g. up and down like the lift-traverse unit in figure 7. All transport devices can be modeled with a reality compliant electrical interface to use original controller programs. Some physical effects, like gravity and sliding friction, are also considered. Due to the time-based simulation concept, interaction – e. g. to remove a moving carrier from the belt – is always possible.

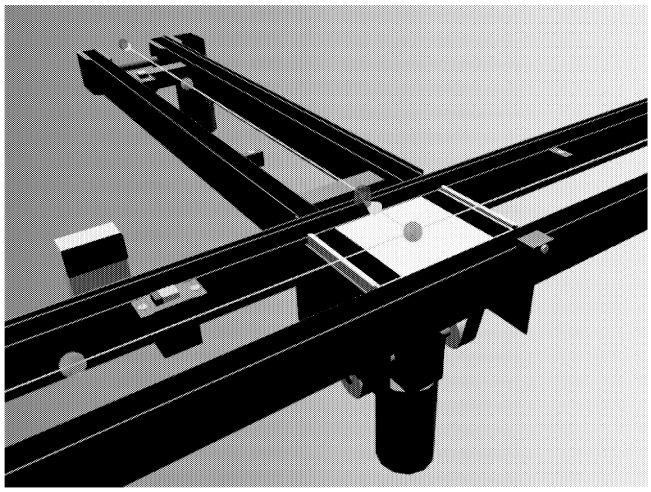


Figure 7: Part of a transport system model – a branching point with conveyor belts, lift-traverse unit, upstream stopper and transceiver, and different sensors

Non-Distortive Component Scaling

When using component models from libraries, the size of the models sometimes must be adjusted in one or more dimensions. As the simulation system represents geometry by facets describing the surface, a simple stretching of the models can distort the model geometry. To cope with this, (Wischnewski 2007) has proposed a concept for intuitive non-distortive scaling. Separate parts of the component can be configured with rules to translate scaling commands like "length 1200 mm". These commands are mapped onto exact parameters for the geometric and functional elements of the part. Figure 8 shows two examples for the scaling. If the belt conveyor on the right is scaled in length, the motor

will not be stretched but moved to its new position. Also the length of the functional transport segment will be adjusted. If the ladder on the left is scaled, rungs will be added or removed from the geometry to conserve an equidistant arrangement of the rungs.

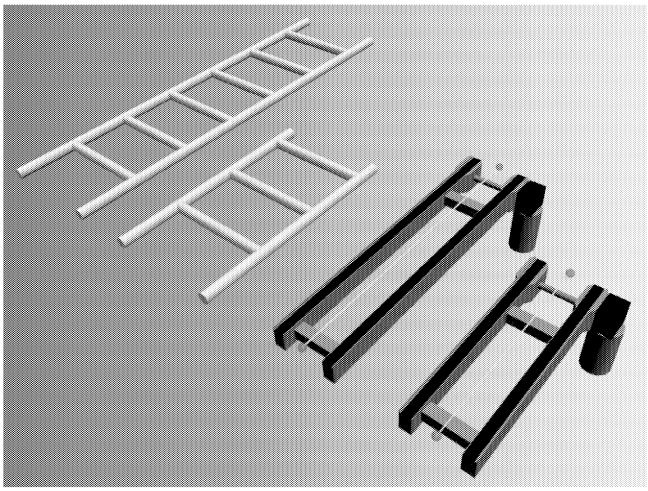


Figure 8: Non-distortive scaling of a simple ladder (left) and a belt conveyor (right)

Interface Adaptation

While modeling manufacturing plants, it happens quite often that the plant requires a certain component which has a close match in the component model libraries of the simulation system. Required and available component are often similar in such a way that geometric and mechanical characteristics are identical but the electric interface differs slightly. To enable the use of such similar models, (Wischnewski 2004) has introduced "Logic Controllers" to adapt the electrical interface. Figure 9 shows how the required electric inputs E'_i are converted to the available inputs E_i using the functions f_i. These functions can be Boolean or arithmetic. Even cascades of multiple logic controllers can be used.

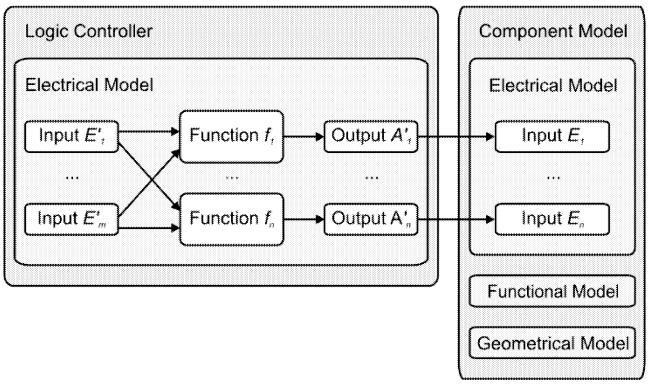


Figure 9: Logic Controller to adapt the electrical interface of a component model from a library

Solution Finder

As mentioned before, the extension interface offers the possibility to access all SEDD elements. Based on this, the extension module "Solution Finder" provides a scripting language including high-level programming language elements and specialized functions for modeling and simulation. Engineers can write scripts to automate the planning and programming of industrial robot applications (Luedemann-Ravit 2005). In particular, the concept supports layout optimization (Rossmann 2009) and the creation of optimal robot trajectories. Figure 10 shows a simple example for the creation of a robot path with the Solution Finder. The script defines a plane which cuts the C-shaped work piece and thus defines the path line the robot shall move along. Moreover, the script automatically creates all positions and a native language robot program to move to all positions. Once created, the same script may be reused for similar tasks in other projects.

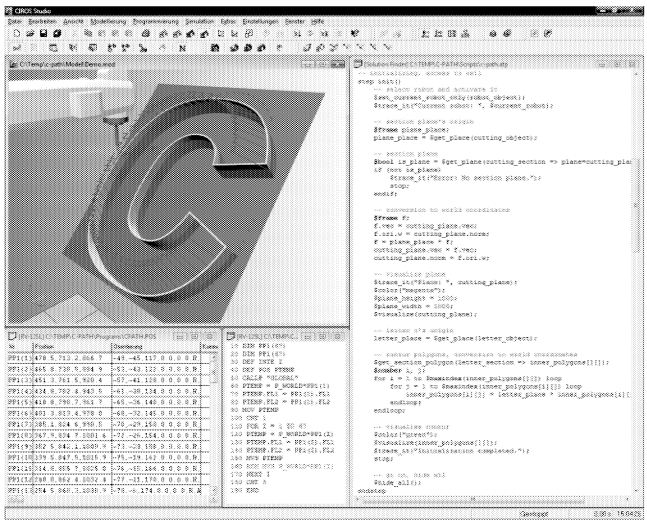


Figure 10: Solution Finder script (right) for the automated creation of a program and positions for a robot path along a C-shaped work piece

Cycle Time Analysis

(Rossmann et al. 2008) show how the integration of a cycle time analysis tool into a 3-D simulation system can contribute to the optimization of robot-based manufacturing plants. Figure 11 presents a typical cycle time diagram. The list to the left shows actions which contribute to the overall process while the Gantt chart to the right shows dependencies of these actions within the process. Engineers can create cycle time diagrams to obtain the overall process time – even in cases with many actions and complex dependencies. The integration of such cycle time diagrams into the simulation system furthermore offers the possibility to control the 3-D model of the plant. The bars of a diagram are interpreted to determine the values of electrical inputs of corresponding model devices which will behave accordingly during simulation. Then, engineers can

intuitively observe in 3-D if the planning of the process makes sense and appears optimal. The other way round, a simulation model controlled e.g. by a PLC during simulation can be "recorded" concerning changing values of the devices' inputs thus producing a cycle time logging for further process analysis. If an engineer is satisfied with a planned cycle time diagram, signals of sensors can be considered in the diagram and sequential flow chart (SFC) diagrams as defined in IEC 61131-3 can be created automatically. These can directly be used on an original PLC if it features SFC import.

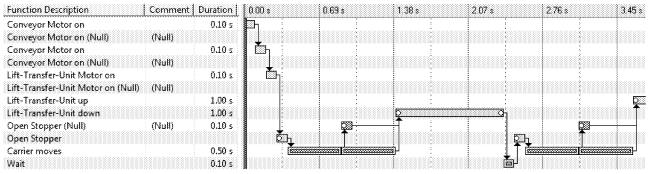


Figure 11: Cycle time diagram with actions and dependencies, also for the control of 3-D simulation models

Other Extension Modules

In addition to the extensions presented, a lot more modules enhancing the capabilities of the simulation system have been integrated. These modules base on concepts for collision detection (Uthoff 1998), distributed simulation (Pensky 2004), an interface to a real robot controller's data (Freund et al. 2004b), enhanced mechanisms for e. g. cardan joints (Freund et al. 2004a), trajectory creation and process simulation for surface treatment processes (Rokossa 1999), and tool change systems (Wischnewski et al. 2009).

USABILITY CONCEPTS

The concepts for the environment model, the system architecture, the controllers, and all the extensions fit together very well. However, the conceptual independence of the modules results in a vast number of usage scenarios. On the one hand, this flexibility does not unnecessarily restrict the applicability of the simulation system to certain problems. On the other hand, the same flexibility prevents a straight course of user action. To cope with this problem, specially designed *wizards* have been created to guide the user through the configuration of typical problems, e. g. the setup of a robotic work cell or the parameterization of real time collision detection.
Besides the wizards, further concepts to support engineers in modeling and simulating have been developed. A method to ease the handling of arbitrary elements in 3-D is the *master coordinate system*. This is a 6DOF coordinate system which can be freely positioned in 3-D space. Positioning can be carried out graphically, relatively to other elements, or by entering absolute coordinate values. Then, the modeling environment allows the execution of many functions for the current position and orientation of the master coordinate system like aligning SEDD elements or moving the current robot. The master coordinate system is also employed in programming robots. The robot programming language RAPID for ABB robots, for example, represents orientations in the very unintuitive quaternion description. If a program line containing such a description is selected, the contained position and orientation can be used to set the master coordinate system in 3-D space. This way, even complex program lines can be understood easily.
Another concept simplifying the use of the simulation system specifies *user defined buttons*. These can be labeled and positioned anywhere in the 2-D view of the 3-D scene. If clicked with the mouse pointer, a corresponding electrical output will be set. Such an output can then be used to control any input in the simulation model. Using this concept, the user can easily create small user interfaces e. g. for repeated experiments.

APPLICATIONS

As mentioned above, the simulation system has a modular structure and thus allows for optimal project specific configurations. This means that a user does not always have to face all features available but only a subset of relevant modules. This way, the overall complexity of the software and thus the time spent for training is kept small.
Currently, the software is used for the simulation of hardware components of the learning system MPS (Modular Production System) by Festo Didactic. Figure 12 shows a catalogue photo of the hardware module "MPS Robot Station" to the left and a screenshot of the fully functional simulation model of the same module to the right. Especially in automation education, a high quality representation of the components is very important. Students learn the course of the virtual production by experimenting with the model concerning all mechatronic devices, the wiring of these, and the programming of the robot in its native programming language. After students have understood the functioning of the model, they apply their new knowledge to the functionally identical hardware components. The use of the simulation software thus allows for training more students at the same time and reduces costs for the purchase of hardware equipment.

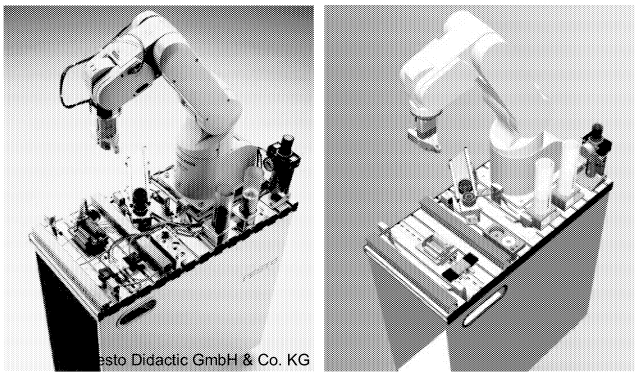


Figure 12: Photo of an automated mechatronic module (l.) and screenshot of the corresponding simulation model (r.)

Figure 13 shows another example from the Festo Didactic learning system. The small manufacturing plant with ten stations integrates a robot, different mechanical devices, a conveyor system, many part magazines, lots of buttons and switches for control, and eight PLCs. Simulation of the model uses the original Melfa-Basic robot program, internally interprets the original Siemens STEP7 PLC programs, and allows all buttons to be pressed interactively during simulation. Every single PLC modeled or even all together can be switched not to be interpreted internally but to communicate with external PLC hardware coupled via OPC thus making up a hardware-in-the-loop scenario.

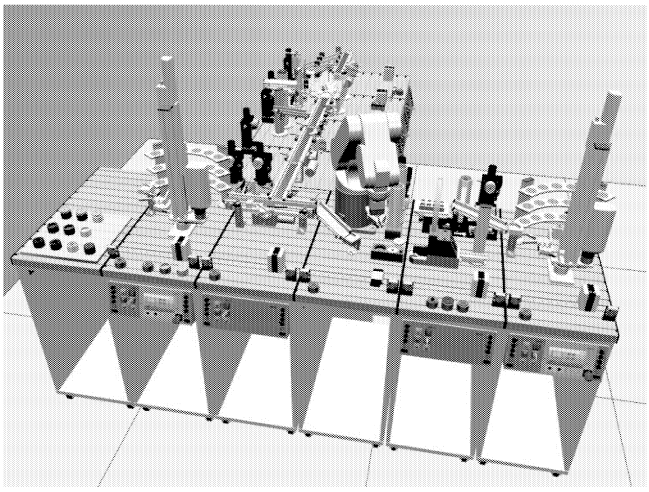


Figure 13: Fully functional 3-D model of a mechatronic plant controlled by original robot and PLC programs

Besides the applications from education, the simulation system is also used for the virtual production in industry. Unfortunately, most of the projects are confidential and must not be published. At least, (Rossmann 2009) presents an industrial application especially of the Solution Finder which is used to optimize industrial robotic work cells. Figure 14 shows a robot which is used to load and unload two presses. The layout of the plant was optimized to achieve a minimum cycle time. All robot positions and paths were verified to be free of collisions. The positions obtained were then directly used on the real robot.

The simulation system also enables the use of virtual reality scenarios. Figure 15 shows a photo taken inside a seven-walled walk-in 3-D projection environment. The model of a manufacturing plant is simulated and engineers can freely interact with the model during simulation. For example, engineers can take transport carriers off the conveyor system to test the controller's reaction during virtual commissioning.

The use of virtual reality environments will be especially useful if members of different departments of a company want to analyze a complex plant together. This is because the high-quality representation makes them immerse into virtual reality which reduces their cognitive load to comprehend the functioning of the model. Thus, more intuitive potential remains to consider the actual problem.

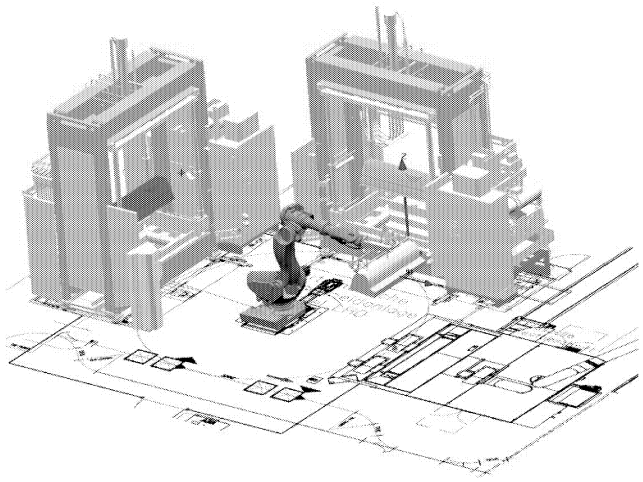


Figure 14: Simulation model of a robot workcell for the optimization of robot programs



Figure 15: Interaction with a 3-D model of a manufacturing plant within a walk-in virtual reality environment

CONCLUSION

In this paper, we have presented a comprehensive 3-D simulation system to be used in industrial virtual production scenarios. The system has an open and extendable architecture and is based on powerful concepts for different fields of modeling and simulation. It integrates a universal environment model to describe all relevant devices together with their geometric, mechanical, and electric properties. At the same point of time, several different controllers can connect to the model and thus control its elements. Extension modules allow for the dynamic adding of features to the simulation systems. The modularity also enables the project specific configuration of the software and thus results in clear, manageable, and operable software. Engineers trained for the use of the software in virtual production scenarios report fast learning success and efficient use. Future work will focus on even more new usability concepts to further improve the efficient use of the simulation system.

REFERENCES

Bauer, R. 2003. *Öffnung einer Arbeitszellensimulation zur Anbindung externer Steuerungen über industrielle Kommunikationssysteme.* Diss., University of Dortmund, Germany. In German.

Freund, E. and H. Weber. 1991. "Entwicklung eines VAL-II-Compilers für die Zielsprache IRDATA." Zeitschrift Robotersysteme. 7 (1991), Springer, p. 139-147. In German.

Freund, E., J. Uthoff, A. Hypki, and U. van der Valk. 1993. "COSIMIR und PCROB: Integration von Zellensimulation und Robotersteuerung auf PCs." In *VDI/VDE-GMA-Fachtagung Intelligente Steuerung und Regelung von Robotern.* VDI Berichte 1094. p. 823-833. In German.

Freund, E., A. Hypki, F. Heinze, and R. Bauer. 2001. "COSIMIR® PLC – 3D Simulation of PLC Programs." In *Proc. 6th IFAC Symposium on Cost Oriented Automation.* Berlin, Germany.

Freund, E., B. Luedemann-Ravit, O. Stern, and T. Koch. 2001. "Creating the Architecture of a Translator Framework for Robot Programming Languages." In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA 2001).* Seoul, Korea.

Freund, E., A. Hypki, R. Bauer, and D. H. Pensky. 2002. "Real-time Coupling of the 3D Workcell Simulation System COSIMIR®." In *Proc. of the Int. Conf. on Information Technology & Applications (ICITA 2002).* Bathurst, Australia.

Freund, E., R. Feist, D. H. Pensky, and R. Wischnewski. 2004. "3-D Graphical Simulation of Complete Manufacturing Systems in Real-Time." In *Proc. of the 6th IASTED Conf. on Control and Applications (CA 2004).* Marina del Rey, CA, USA.

Freund, E., F. Heinze, A. Hypki, and T. Koch. 2004. "User Friendly Operation of Industrial Robots with a New Robot Controller Interface." In *Proc. of the Int. Symposium on Robotics (ISR 2004).* Paris, France.

Hypki, A. 2008. *Beitrag zur Simulation industrieller Automatisierungssysteme.* Diss., TU Dortmund University, Germany. In German.

Keibel, A. 2003. *Konzeption und Realisierung eines integrierten Moduls zur Simulation und Steuerung von Kinematiksystemen.* Diss., University of Dortmund, Germany. In German.

Luedemann-Ravit, B. 2005. *Ein System zur Automatisierung der Planung und Programmierung von industriellen Roboterapplikationen.* Diss., University of Dortmund, Germany. In German.

Pensky, D. H. 2004, *Parallele und verteilte Simulation industrieller Produktionsprozesse.* Diss., University of Dortmund, Germany. In German.

Uthoff, J. 1998. *Offenes, modulares System zur zellenorientierten Robotersimulation.* Diss., University of Dortmund, Germany. In German.

Rokossa, D. 1999. *Prozeßorientierte Offline-Programmierung von Industrierobotern.* Diss., University of Dortmund, Germany. In German.

Rossmann, J. 1993. *Echtzeitfähige, kollisionsvermeidende Bahnplanung für Mehrrobotersysteme.* Diss., University of Dortmund, Germany. In German.

Rossmann, J., O. Stern, and R. Wischnewski. 2007. "Eine Systematik mit einem darauf abgestimmten Softwarewerkzeug zur durchgängigen Virtuellen Inbetriebnahme von Fertigungsanlagen: Von der Planung über die Simulation zum Betrieb." atp 49, Issue 7, p. 52-56. In German.

Rossmann, J., T. Koch, and O. Stern. 2008. "Optimierung von robotergestützten Fertigungsanlagen durch Integration eines Taktzeitanalysewerkzeugs in ein 3-D-Simulationssystem." In *Tagungsband Robotik 2008*. VDI-Bericht 2012. In German.

Rossmann, J., R. Wischnewski, and P. Lenk. 2009. "Optimization of Industrial Robot Work Cells using the Automated Variant Simulation." In *7th Int. Industrial Simulation Conf. (ISC'2009)*. EUROSIS-ETI. Loughborough, UK.

Van der Valk, U. 1995. *Konzeption und Realisierung einer PC-basierten Robotersteuerung.* Diss., University of Dortmund, Germany. In German.

Wischnewski, R. and E. Freund. 2004. "COSIMIR® Transport: Modeling, Simulation and Emulation of Modular Carrier Based Transport Systems." In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA 2004).* New Orleans, LA, USA.

Wischnewski, R. 2007. *Transportsysteme mit Spurführung in der Virtuellen Produktion.* Diss., RWTH Aachen University, Germany. In German.

Wischnewski, R. and J. Rossmann. 2009. "3-D Modelling and Simulation of Tool Change Systems for the Virtual Production." In *Proc. of the 20th IASTED Int. Conf. Modelling and Simulation (MS 2009)*. Banff, Alberta, Canada.

AUTHOR BIOGRAPHY

JUERGEN ROSSMANN studied electrical engineering at the universities of Dortmund and Bochum in Germany. After this, he worked at the Institute of Robotics Research (IRF) of the University of Dortmund – first as a scientific assistant, then as a group leader and finally as head of department. He received his Ph.D. in 1993 from the University of Dortmund. After several stays in the USA as a visiting scientist, he was appointed visiting professor for robotics and computer graphics at the University of Southern California in 1998. After his postdoctoral lecture qualification he was appointed professor at the RWTH Aachen University where he now heads the Institute of Man-Machine-Interaction.

ROLAND WISCHNEWSKI studied applied computer science with focus on engineering technology at the University of Dortmund in Germany. From 1999 to 2005 he worked as a scientific assistant at the Institute of Robotics Research (IRF) of the University of Dortmund. Since April 2005, he works at the Department of Robot Technology at RIF e. V. in Dortmund as head of the team Industrial Simulation Systems. In 2007 he received his doctor's degree in electrical engineering from the RWTH Aachen University.

OLIVER STERN studied computer science with focus on robotics, engineering technology, and knowledge representation at the University of Karlsruhe in Germany and at the ENSAIS in Strasbourg, France. From 1996 to 2004 he worked as a scientific assistant at the Institute of Robotics Research (IRF) of the University of Dortmund. Since 2005 he works at RIF e. V. in Dortmund – first as project manager for supervisory control systems, and since April 2006 as head of the department Robot Technology.

SPECIFIED CAD (COMPUTER AIDED DESIGN) TOOL FOR RAPID MANUFACTURING TECHNOLOGY

Estitxu Martínez de Alegría Uriarte
Iñaki García
Aurelio Díaz de Arcaya
Gonzalo Lázaro
Luis Usatorre
Department of Engineering
EUVE (European Virtual Engineering technological centre)
01010, Vitoria, Álava
Spain
E-mail: emalegria@euve.org

KEYWORDS

CAD (Computer Aided Design), RM (Rapid Manufacturing), KBE (Knowledge Based Engineering), KBS (Knowledge Based System).

INTRODUCTION

The current work is based on solving design problems in Rapid Manufacturing field. The aim is not only to show results but the exchange of ideas with previous studies in order to share new challenges emerging in this area.

First, a short description of Rapid Manufacturing is required.
Rapid Manufacturing technologies, even emerging, are constantly evolving. Therefore, it is difficult to make a universal definition. One of the most useful definitions is: *"a group of techniques, technologies and methods that let a fast, flexible and competitive pieces manufacture. These pieces may be: prototypes, molds, matrix or final products. They always are made from digital information"*.

Rapid Manufacturing technology uses 3D CAD information in order to improve the graphics representation of the designed object. The model made by CAD is the first step for the application of KBE techniques which provide a support for the design decision processes.

OBJECTIVES

The main goal of this work is the development of new tools based on information and communication technologies (ICTs) in order to support the new manufacturing methods requiring Rapid Manufacturing technologies. These tools must develop new business models generated from Rapid Manufacturing according to the SME's requirements.

The objective is to create a base in order to develop new innovative models that use Rapid Manufacturing as an instrument for focusing on new functionalities. For that, it is important to research in the following topics:

1. To create design guidelines of unique and personified pieces which use Rapid Manufacturing technologies.
2. To increase the automation procedures in the design process for Rapid Manufacturing.
3. To introduce innovative features and new technologies in the design process for Rapid Manufacturing
4. To promote the use of Rapid Manufacturing technologies in the industrial design processes.
5. To create an infrastructure in order to help to the logistics in the Rapid Manufacturing field.
6. To study the variability of the Rapid Manufacturing organizations from the features ´models point of view.
7. To develop an inference and application KBE engine for Rapid Manufacturing independent from commercial CAD platforms.

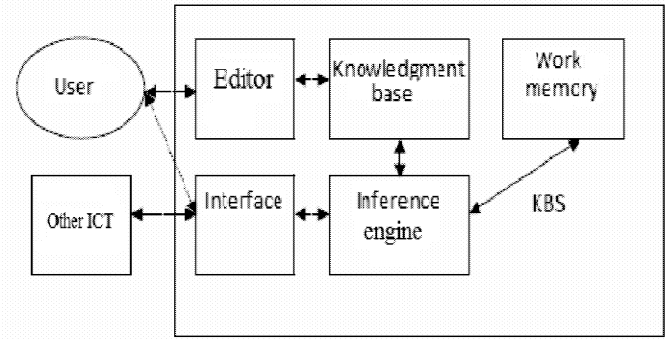
In brief, the work that is being described in this paper is focused on achieving the following goals:

- To develop CAD tools letting the inclusion of design paradigms and specified manufacturing parameters for Rapid Manufacturing technologies.
- A virtual marketplace that makes possible to create automatically agreements between different companies to create an automated supply chain.
- To make possible the optimization of the logistics systems of the organizations according to the Rapid Manufacturing domain.
- To develop the first prototype for Rapid Manufacturing over an inference engine associated to a commercial CAD tool. This will make easier the analysis phase and the next

development of the independent KBE application.

KBE is a discipline of the KBS [Picture 1]. They are Knowledgment Based Systems. They are used by the engineers in order to help them with their computer tools. The most usual application in the engineering is the development of CAD models during the product development.

Nowadays, KBE application is extending to other fields like CAM (computer aided manufacturing) during the manufacture and CAE (computer aided engineering) during the analysis phases. KBE tools are the needed complements for CAD-CAE-CAM tools for any engineering processes.



Picture1. KBS working way

KBE could be described as: A method in order to acquire Knowledgment and reuse it. KBE is a subgroup of KBS which is a part of artificial intelligence.

These systems include as innovations the adoption of new advances in the artificial intelligence in design.

Some advantages of these systems are:

- Reduction of time in the design phase
- Introduction of techniques like DFX (design for assembly and so on)
- The KBE development can be divided in three stages:
  - Knowledge management design process. It is required to analyse the current process to detect needs, more processes, data flows and so on in order to establish a new design process.
  - Knowledge management product. This includes aspects like creativity, intuition, aesthetics, functionality and so on.
  - Knowledge application. The knowledge is integrated in the application in order to control the good operation of the created product regarding to its restrictions and avoiding fail possibilities.

In short, the implantation of Rapid Manufacturing needs developments in ICTs in order to support the new models of the manufacturing chain of RM technologies.

As well, the business culture and the product design must change to be adapted to the demands and possibilities of the new Rapid Manufacturing technologies.

TECHNOLOGICAL INNOVATION

The RM discipline is not focused only on the additive manufacturing technologies.

This project works on the RM concept as a whole; this is, with the techniques, technologies and methodologies that allow a rapid and flexible manufacturing of pieces coming from digital information.

This digital information should let the integration of the customer's demands like time reduction, cost and so on.

A long time ago, the researches were based on technologies with a true prototype that was useful for a short period of time.

However, this prototype is used as an element in order to check the aesthetic functionality and dimensions, but it cannot be used as a final product.

Nowadays, the developed RM technologies are applied to obtain a prototype that can be used as a final product.

There are several researching studies which create relationships between design features of the manufacture regarding to the traditional processes.

However, when RM processes are being analysed, it is difficult to know the features between design and manufacture. The high variability of RM technologies and the number of used materials should be studied in order to be applied in the design processes.

To progress in this line, the project is focused on the following aspects of technological innovation:

1. To generate the most objective information that can help to the design users to use Rapid Manufacturing technologies based on final uses directly.

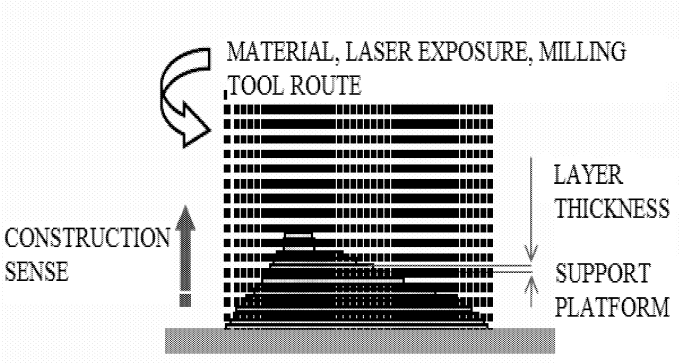
2. To propose a formal model of the design processes with its features to help to the users when the RM technologies are used. This is being linked to the manufacturing process, in relation to products and their functionalities and to the appropriate RM technology.

3. To adapt the technologies and materials to the product that is in the main interesting sectors.

The paper is studying the joint between parts and components (geometries), the union of the function between parts that involves the reduction of parts number to manufacture.

Furthermore, this paper is working mainly in the reengineering of the whole process of designing products in order to create an innovative tool. Closed structures and parts models are being studied for CAD systems.
Generally, the used elements are standard, that are not allowed to the customer's demands.

The result of the paper must be conceptualizingd as the addition of geometrical elements, because the RM technologies do not have the possibility to manufacture a personalized treatment for the pieces [Picture 2].
The target is to adapt the real design processes to that processes that make the RM technologies use.



Picture 2. Description Rapid Manufacturing working way

SPECIFIED APPLICATION FOR RAPID MANUFACTURING

There are two ways of implementing an application. On one hand, commercial applications such as CATIA, AUTOCAD, SolidWorks and so on. On the other hand, open source applications, whose functionality is shorter than the first ones.

This option is based on the development of a specified application to introduce specific parameters to RM technology.

Its main features are:

- It is used after the CAD tool: its entrance data is a CAD model.
- It lets to select the RM technology in its manufacture. This will condition the rest of the parameters to be introduced.
- It has a piece's 3D visualization, with the capacity in order to choose parts of the piece and assigning new properties in relation to the RM.
- It provides as an exit, the same model as the entrance, amplified with the new parameters or manufacturing strategies.
- The main advantage of this development in respect to the modification of the CAD is its universality:
- It is an application out of the CAD that can be adopted with any company with independence of the design tool that they are using.

But, also, it has some disadvantages that must be assessed:

- It needs a design phase after using CAD, and, therefore, it supposes an increase in the design time.
- To download models that come from different CAD software is a difficult process.
- Each tool has its own form file and it only let to export too few geometries interchange formats like IGEs, STL or STEP.
- In this exportation, some information about the composition of the model is lost.
- In any case, this format conversion was needed to take the model to RM applications. With this, application, this conversion should be made one stage before.
- The development of a whole application with tridimmensioanl visualization implies an extra cost in order to implement the specified functionalities for RM.

CURRENT WORKING WAY IN RAPID MANUFACTURING SOFTWARE

Nowadays, the current way to work with that kind of technology is as follows:

1. The piece is drawn to be built in CAD.
   - SolidWorks software is used to develop this task.
   - The customers send us STEP IGES or STL format files.
2. The file is saved in CAD (SolidWorks) and STL formats.
3. In RM, the common format to wok is STL. With the Magics software, errors are detected (overlaps, hollows and so on). These errors are corrected with Fix Wizards or, manually (which it requires too much time), if it cannot be corrected automatically.
4. The piece is located in the virtual work platform (simulated like the real machine) of Magics.
5. The supports are generated automatically (according to established parameters in the library. They can be modified). The exit file is *.f&s.

6. The *.f&s files are opened with Realizer software of the SLM machine (Selective Laser Melting machine).
7. Some parameters are assigned to a piece and the hatch strategy is defined.
8. If the piece needs supports, they are added doing a Z axis alienation. Then, everything is saved as a whole.
9. This is located on the work platform and the machine starts working. The laser scans the geometry of each layer until manufacturing the whole piece.

Due to the existing problems in Rapid Manufacturing, new developments in order to support new models in manufacturing chains are increasingly being demanded in communication and information technologies.

The existing problems in the software are the following:

- When customers send us STL files or our Magics files, there are problems to try to modify them.
- The default way to read files with Solidworks is as a graphic and nothing can be modified.
- Even being able to read as a solid, it lets make changes with a determinate file's size. From that size, all the imported files are read as graphics and it is not possible to modify them.
- There are problems even trying to reduce the file's size because the geometrical information is lost.
- Therefore, it would be good to be able to read STL files as solids with a very big size. Another improvement is to be able to modify these files without losing any information.
- Everything works with licenses for modules and it must be renewed every six months.
    - If the computer is reseted, the license is lost because it depends on the IP address computer. So, the users have to contact with the maker to get another password for the computer.
- When Fix Wizard (software that improves STL files) cannot repair the STL files is necessary to improve them manually. This cost too much time because it must be checked triangle by triangle. When a fail is located, it must be improved removing it and drawing it again.
- The CAD module of Magics is very simple. It has not the same philosophy as a common CAD. The icons are very basic.
- When a piece must be budgeted for a customer is necessary to know the machine time and the material+machine+postproccesed cost of the piece.
    - Magics software has an area to calculate the time according to the piece volume, area, height and so on. These parameters can be added or removed to achieve a real time according to the information of the SLM machine by means of "self learning".

Magics is able to generate the own platform and to modify all kind of parameters. The SLM machine data are not in the current library and they had to be copied to the machine to be able to work with the software.

OBTAINED ADVANTAGES WITH THIS TOOL

The free design and manufacture take importance with a bigger scope of materials.

The new paradigm let adapt the technologies to the required and specified problems and the customer's demands.

The product can be used as a final part removing intermediary phases like using tools and so on.

The restrictions that are found in the traditional manufacture are solved.

CONCLUSIONS

As a result of this work, it is important to emphasize the importance of making improvements to solve the current problems that exist nowadays. Next, some improvements are describing:

- In order to work with Rapid manufacturing philosophy, the CAD program should warn about minimum angles and thicknesses. Like this, the software user will be informed about the security limits of manufacturing. For example:
    - Minimum thickness to manufacture: 0.2mm
    - Maximum angle without supports with respect to the horizontal axis: 45º, and so on.
- To provide a material data base in which the approximated time will be warned for each material and material cost per kilogram.
- It is very important to be able to export STL files to common files like STEP, IGES, SLDRT and so on without problems.
- When a file is exported, it is important to pay attention to the piece resolution because in Magics (STL format) the surface is defined by the triangles that make to lose information in order to export the files.

REFERENCES

- Ghany KA, Moustafa SF. "Comparison between the products of four RPM systems for metals". Rapid Prototyping J 2006;12(2):86-94.
- Gibson I, Shi D. "Material properties and fabrication parameters in selective laser sintering process". Rapid Prototyping J 1997;3(4):129-136.

- Hague R, Mansour S, Saleh N, Harris R. "Materials analysis of stereolithography resins for use in Rapid Manufacturing". J Mater Sci 2004;39(7):2457-2464.
- Van Elsen M, Al-Bender F, Kruth J. "Application of dimensional analysis to selective laser melting". Rapid Prototyping J. 2008;14(1):15-22.
- Tolochko NK, Mozzharov SE, Yadroitsev IA, Laoui T, Froyen L, Titov VI, et al. "Balling processes during selective laser treatment of powders". Rapid Prototyping J2004;10(2):78-87.
- Proyecto Singular Estratégico IBE-RM. Ministerio de Ciencia e Innovación.

AUTHOR BIOGRAPHY

ESTITXU MARTÍNEZ DE ALEGRÍA URIARTE was born in Vitoria, Álava, Spain and went to the University of Deusto, in Bilbao, Spain where she studied computer engineering and industrial engineering and management and obtained her degrees in 2008 and 2009 respectively. Furthermore, she studied part of her first degree in the University of Saint Andrews (United Kingdom). She worked for a year in AERNNOVA as a logistics technician before moving in 2009 to EUVE (European Virtual Engineering technological centre) where she has been working ever since.

IÑAKI GARCÍA was born in Vitoria and went to the University of Basque Country, in Spain, where he studied telecommunications engineering. He started working in EUVE (European Virtual Engineering technological centre) in 2003. He has been working there ever since.

AURELIO DÍAZ DE ARCAYA was born in Vitoria and went to the University of Basque Country, in Spain, where he studied Quimical Engineering. He worked for a year in the meteorology department of EUVE and now, he is in the department of engineering of EUVE (European Virtual Engineering technological centre).

GONZALO LÁZARO was born in Vitoria, and went to the University of Basque Country, in Spain, where he studied computer engineering. He is working in the department of engineering of EUVE (European Virtual Engineering technological centre).

LUIS USATORRE was born in Vitoria and went to the University of Basque Country, in Spain, where he studied Industrial engineering. Has several years of experience in industrial companies and works in EUVE (European Virtual Engineering technological centre) as engineering manager since 2007.

MICRO PATTERN SHAPING SIMULATON

Su-Jin Kim, Jong-Min Lee and Duy Le
School of Mechanical and Aerospace Engineering, Gyeongsang National University
900 Gajwa-dong, Jinju, South Korea, 660-701
E-mail: sujinkim@gnu.ac.kr
Tae-Jin Je
Korea Institute of Machinery & Materials
104, Sinseongno, Yuseong-gu, Daejeon, South Korea, 305-343

KEYWORDS
Micro pattern, Micro machining, Micro cutting, Simulation

ABSTRACT

The micro pattern machining on the surface of a wide mold is not easily simulated using conventional methods. This paper represents the micro pattern cutting simulation software. The software simulates micro pattern shaping in 3D geometry and predicts the cutting force. The v-groove for prism and pyramid patterns and the rectangular groove for rectangular and pillar patterns are simulated. The code of this program is built using visual C++ and OpenGL.

NOMENCLATURE

F_c = cutting force
K = specific cutting energy
A = cutting area
C = experimental constant
n = experimental constant
t = chip thickness
d_i = cutting depth at i^{th} step
a = angle of diamond tool and pattern
p = pitch between each pattern
w = width of rectangular tool and pattern

INTRODUCTION

The light guide panel of a liquid crystal display or the deflection sheet of traffic signs is a wide sheet including a large number of micro patterns that are designed to refract or reflect light. The huge amount of micro patterns on the wide sheet is produced using a roll mold or plate mold. The molds are machined by micro shaping using a single crystal diamond tool [1].

In order to stay competitive in today's international markets, companies must deliver new, high-quality products in a short time with a variety of versions at minimum cost. Manufacturing simulation is rapidly rising in today's international markets as an interesting strategy for product development. Its primary aim was reducing the lead time to market and costs associated with new product development. Manufacturing simulation offers a test platform for the time-consuming and expensive physical experiments. Micro machining in particular has challenges and limitations, because simple scaling cannot be applied to modeling the phenomena of micro processing.

The machining simulation was studied for milling and the precise turning process [2-4]. The study continued to the experimental and analytical research of the micro cutting process [5-8]. Unlike conventional macro-machining processes, micro-machining displays different characteristics due to its significant size reduction. Most chip formation investigations are derived from macro-ultra-precision diamond and hardened steel cutting operations; numerous publications have focused on the effect of round edges and minimum chip thickness [5]. Often, the edge radius of the tools is relatively larger than the chip thickness. In contrast to the conventional sharp-edge cutting model, chip shear in micro-machining occurs along the rounded tool edge [8]. As a result, cutting has a large negative rake angle, which affects the magnitude of the plowing and shearing forces. Therefore, a relatively large volume of material has to become fully plastic for a relatively small amount of material to be removed, resulting in significant increases in specific energy levels [6-7]. Further, when the chip thickness is below a critical point, chips may not be generated during the cutting process; instead, the workpiece material elastically deforms. Critical chip thickness is proposed to be a quarter of the size of cutter edge radius [6].

The edge radius of the diamond tool used in this research is less than 0.1 μ m and the chip thickness in the roughing stage is much bigger than 0.025 μ m. Thus, the cutting force model does not consider plowing effects and elastic recovery but takes into account a significant increase in specific cutting energy that occurs when chip thickness is decreased.

The finite element method was also used to predict the cutting process [9, 10]. The nano-scale machining simulation is now possible using the molecular dynamics approach and finite element method [11-13]. The micro shaping force prediction model specialized for prism, rectangular, pyramid and pillar patterns needs to be developed.

In this paper, several key issues for developing a micro shaping optimization platform using virtual manufacturing technology are discussed, i.e., representation of micro cutting force modeling, introducing a method of predicting cutting force and simulating the shaping process. All issues are built upon one program. Its functions are predicting cutting force and simulating a shaping process in 3D space. The v-groove for prism and pyramid patterns and the rectangular groove for rectangle and pillar patterns are simulated.

CUTTING FORCE MODEL

The light guide panel of a liquid crystal display or the deflection sheet of traffic signs is a wide sheet including a large number of micro patterns that are designed to refract or reflect light. The huge amount of micro patterns on the wide sheet is produced using a roll mold or plate mold. The molds are machined by micro shaping using a single crystal diamond tool [1]. The principles of micro-machining are similar to conventional cutting operations. The surface of the workpiece is mechanically removed using micro-tools. The cutting force is directly related to chip formation. It is suitable to develop a micro cutting force model according to the specific cutting energy.

Considering the physical conditions that occur in the micro-domain, a cutting force analyzing technique will be applicable to micro cutting development. The developing program predicts the cutting force of micro shaping with the experimental constant, which is calculated from the experimental results. Due to specific cutting energy, the cutting force is calculated generally in the cutting cross section.

$$F_c = KA \text{ , where } K = Ct^{-n} \tag{1}$$

The cutting force model of prism patterns is shown in equation (2) when the cutting width is smaller than the pitch. This usually happens at the roughing step, as shown in Fig. 1 (a). The finishing force model is shown in equation (3) when the width is bigger than the pitch and the prism pattern is formed.

$$F_c = C\left((d_i - d_{i-1})\sin\left(\frac{a}{2}\right)\right)^{-n}\left(d_i^{\,2} - d_{i-1}^{\,2}\right)\tan\left(\frac{a}{2}\right) \tag{2}$$

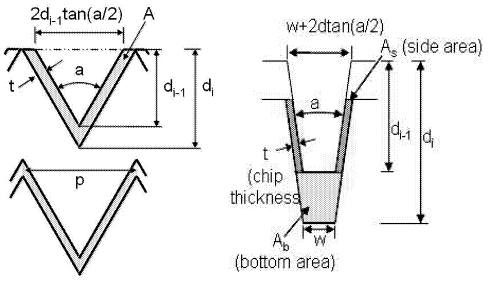
$$F_c = C\left((d_i - d_{i-1})\sin\left(\frac{a}{2}\right)\right)^{-n}(d_i - d_{i-1})p \tag{3}$$

For rectangular patterns, the cutting area is divided into the bottom and two side sections and the corner regions are considered bottom section as shown in Fig. 1 (b). The cutting forces of the bottom and two side sections are computed as shown in equation (4).

$$F_c = C\left((d_i - d_{i-1})\sin\left(\frac{a}{2}\right)\right)^{-n} 2d_{i-1}(d_i - d_{i-1})\tan\left(\frac{a}{2}\right) + C(d_i - d_{i-1})^{-n}\left(w + (d_i - d_{i-1})\tan\left(\frac{a}{2}\right)\right)(d_i - d_{i-1}) \tag{4}$$

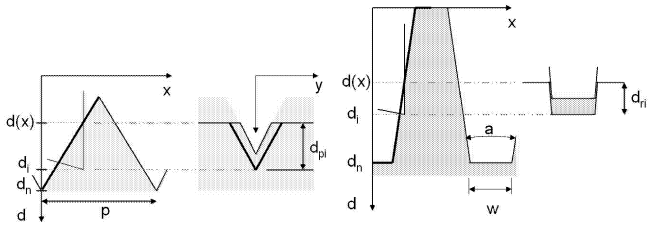
The experimental constant C and n are determined for the purpose of minimizing the error of the measurement cutting force Freal and the prediction cutting force Fmod following the least square method.

$$\min[\sum (F_{\text{mod}} - F_{real})^2] \to C, n \tag{5}$$



(a) (b)

Fig. 1 (a) (b) Cutting area of prism and rectangular patterns



(c) (d)

Fig. 1 (c) (d) Cutting depth of pyramid and pillar patterns

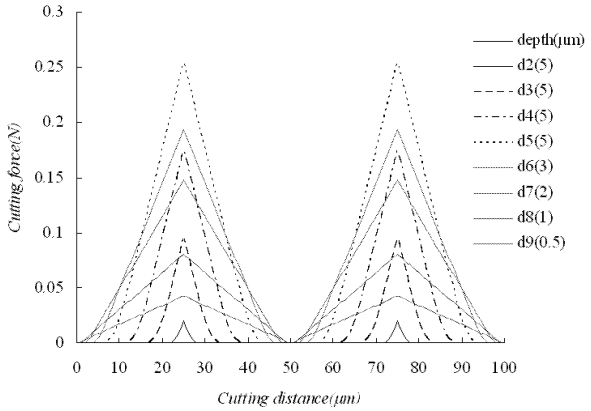


Fig. 1 (e) Cutting force of pyramid pattern

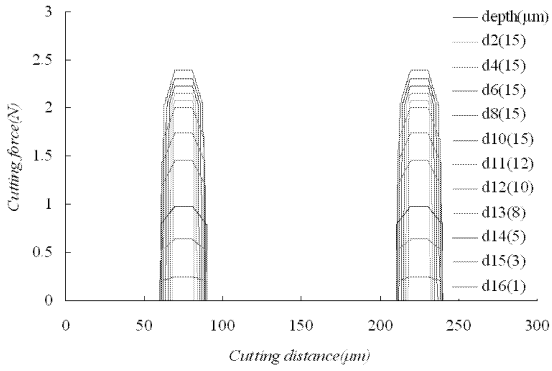
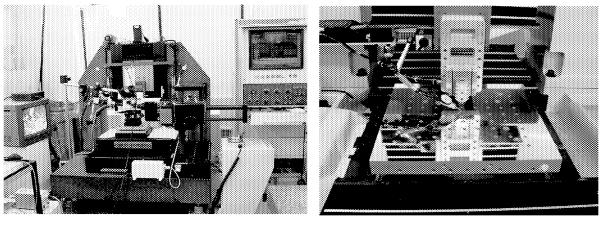


Fig. 1 (f) Cutting force of pillar pattern



Machine tools Tool and stock

Fig. 2 Experimental equipment

The pyramid pattern is fabricated by cutting a prism pattern along the perpendicular direction with the same tool and pitch. The first cutting direction is exactly the same as the prism pattern; the second cutting direction is different because the starting plate already includes a prism pattern perpendicular to the cutting direction, as shown in Fig. 1 (c). The depth wave function of the pyramid pattern is suggested in equation (6), and the cutting force is computed by inserting cutting depth dpi into the equation (2). Fig. 1 (e) shows the cutting force when the pyramid pattern is grooved on the prism pattern plate.

$$\begin{aligned} d_{pi}(x) &= 0 \qquad\qquad if \quad 0 \le x < (d_n - d_i)\tan(a/2) \\ d_{pi}(x) &= \frac{x}{\tan(a/2)} + d_i - d_n \quad if \quad (d_n - d_i)\tan\left(\frac{a}{2}\right) \le x < \frac{p}{2} \\ d_{pi}(-x+p) &= d_{pi}(x) \quad d_{pi}(x+pm) = d_{pi}(x) \quad (mirror\ \&\ array) \end{aligned} \tag{6}$$

The pillar pattern is fabricated by cutting a rectangular pattern along the perpendicular direction with the same tool and pitch. The second direction cutting plate includes a rectangular pattern perpendicular to the cutting direction, as shown in Fig. 1 (d). The depth wave function of the pillar pattern is suggested in equation (7), and the cutting force is computed by inserting cutting depth dri into the equation (4). Fig. 1 (f) shows the cutting force when the pillar pattern is grooved on the rectangular pattern plate. The pitch is 150 μm, the angle is 5.72° and the width is 120 μm.

$$\begin{aligned} d_{ri}(x) &= 0 \qquad\qquad if \quad 0 \le x < w/2 + (d_n - d_i)\tan(a/2) \\ d_{ri}(x) &= \frac{x - w/2}{\tan(a/2)} + d_i - d_n \quad if \quad \frac{w}{2} + (d_n - d_i)\tan\left(\frac{a}{2}\right) \le x < \frac{w}{2} + d_n \tan\left(\frac{a}{2}\right) \\ d_{ri}(x) &= d_i \qquad\qquad if \quad \frac{w}{2} + d_n tai\left(\frac{a}{2}\right) \le x < p/2 \\ d_{ri}(-x+p) &= d_{ri}(x) \quad d_{ri}(x+pm) = d_{ri}(x) \quad (mirror\ \&\ array) \end{aligned} \tag{7}$$

COMPARISON OF MODEL WITH EXPERIMENTAL DATA

Experiment equipments

The experiment for this study was conducted using the equipment at the Korea Institute of Machinery and Materials. The configuration of the precise machine tool, workpiece and single crystal diamond tool is shown in Fig. 2. Alongside the primary X, Y and Z axes, a turning axis was added. The X and Y axes were implemented with an air bearing structure with a maximum position error of 20 nm and average velocity error of 2%. The Z axis did not have any back lash, and a counter balance device was placed on the side of the lead screw to withstand high loads. The PC-based motion controller used the PMAC1 board from Deltatau, which is capable of controlling eight axes. A Kistler Co. 9256A2 3-axis dynamometer was attached to the bottom of the workpiece to measure the cutting forces.
The workpiece is 6:4 brass that was used for micro pattern plate mold in industries. The cutting speed was 1200 mm/min and cutting depth was different at each experiment.

Prism pattern

A v-shaped single crystal diamond tool with 90° angle was used to fabricate a prism pattern with 50 μm pitch. The SEM images of prism patterns at each cutting step are shown in Fig. 3 (a). The cutting forces were measured during the experiment to build and verify the cutting force model of this paper.
The cutting force model of a prism pattern was developed using the cutting mechanism in equation (1) and experimental data in Fig. 3 (a). The experimental constant C and n were computed using the least square method. The graph of the developed model and the measure data are plotted together. The cutting direction is shown on the y axis and the vertical direction is shown on the z axis in the graph.
The cutting force model, developed by the first experiment, was verified by the following experiment. The cutting force predicted by the developed model was compared to another experimental data set. The maximum error between the predicted and measured cutting forces is 0.04 N in the second experiment, as shown in Fig. 3 (b). There were errors among the three experiments. The cutting forces at the common cutting depths are plotted in Fig. 3 (c). The maximum error between the experimental data was 0.04 N, which shows that the error in the developed model is derived from the model itself and the noise of the measured data. It was proved that the micro pattern cutting force prediction model works well on prism patterns, because the errors between the cutting force predicted by the model developed in the first set of experimental data and the measured cutting force in the second and third experiments were smaller than 0.04 N.

Rectangular pattern

A rectangular-shaped diamond tool with 120 μm width and 5.72° taper angle was used to fabricate a rectangular pattern. The SEM images of rectangular patterns at each cutting step are shown in Fig. 4 (a). The cutting forces were measured during the experiment to build and verify the cutting force model of this paper.
The cutting force model of the rectangular pattern was developed using equation (4) and the experimental data in Fig. 4 (a). The model was made using both theory and data from the first experiment. The cutting forces of the model and the experimental data were plotted together.
The cutting force predicted by the developed model was compared to data from the second experiment. The maximum error between the predicted and measured cutting forces is 0.21 N in the second experiment, as shown in Fig. 4 (b). It was proved that the micro pattern cutting force prediction model works well for rectangular patterns, because the error between the cutting force predicted by the model and the measured cutting force was small.

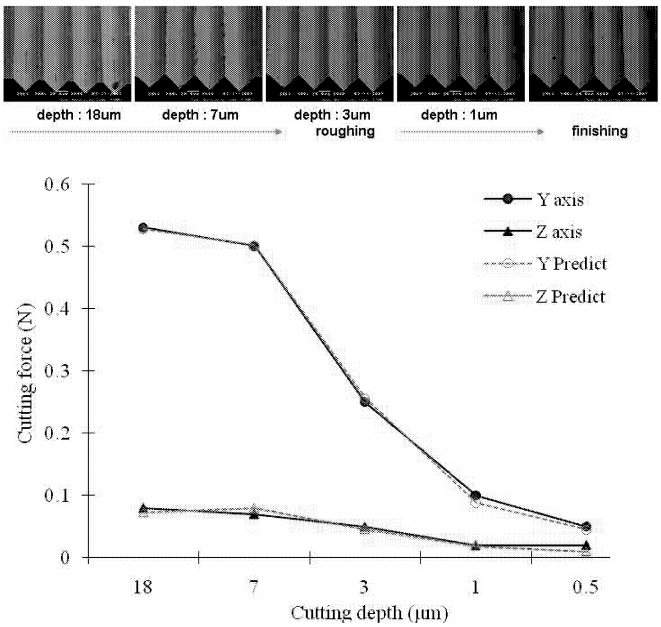


Fig. 3 (a) Force model developed from experimental data of the prism pattern: C_y=0.00174, n_y=0.026, C_z=0.00035, n_z=0.172

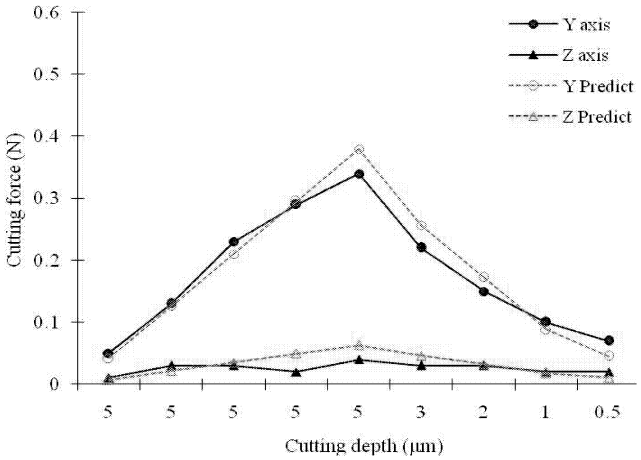


Fig. 3 (b) Force predicted by model (a) and measured force: Maximum error is 0.04 N

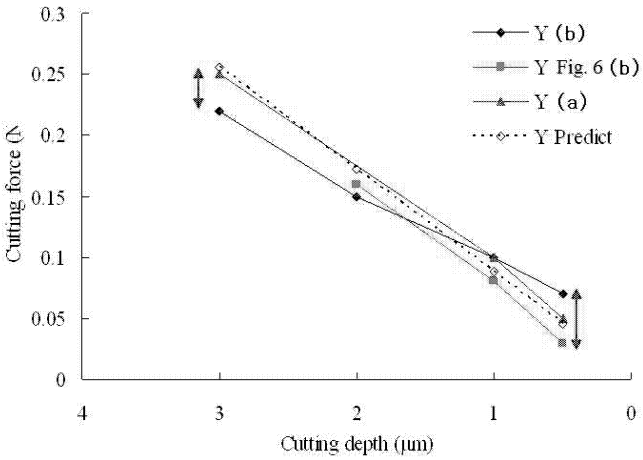


Fig. 3 (c) Measured error of each experiment: Maximum measured error is 0.04 N

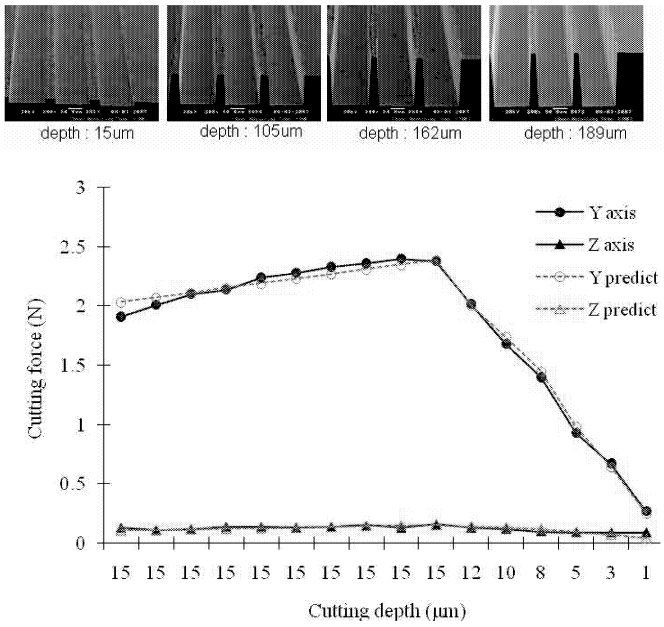


Fig. 4 (a) Force model developed from experimental data of the rectangular pattern: C_y=0.00168, n_y=0.149, C_z=0.00021, n_z=0.468

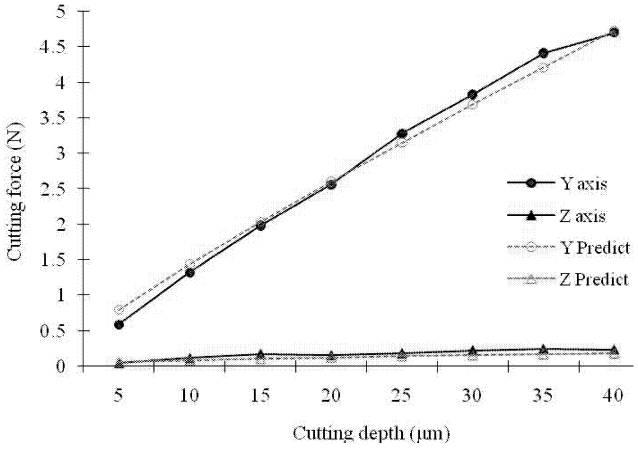


Fig. 4 (b) Force predicted by model (a) and measured force: Maximum error is 0.21 N

MICRO PATTERN SHAPING SIMULATION

Computation of the Cutting Section

The cutting section needs to be computed to predict the micro cutting force and suggest optimal cutting depths automatically. The 3D geometrical micro cutting simulation software is developed to compute the cutting section used to predict shaping mechanism automatically.
The flowchart of the developed micro pattern cutting simulation software is shown in Fig. 5 (a). The geometrical cutting simulation computes the remaining geometry and cutting section. The micro cutting force is computed automatically with the micro cutting force model, and the cutting section is computed using geometrical shaping simulation. Finally, the optimal cutting depths that make the cutting forces uniform are used as output for effective tool path generation. The developed program will be able to increase the productivity of the micro pattern shaping on a wide mold by applying the suggested conditions.

3D Visualization of the Micro Pattern

The 3D geometrical micro cutting simulation software is developed to compute the cutting section used to predict shaping mechanism automatically. It will take too long to visualize several million patterns on a single display. This research reduced the rendering time by arraying a basic pattern in the zoomed window of display. The pattern shapes are expressed by the boundary representation method, which is more precise than sampled representation.
The data structure of the software shown in Fig. 5 (b) is designed to be useable for various sections and multiple shaping directions. The software simulates prism, rectangular, pyramid, pillar, half circle prisms and various angle prisms, as shown in Fig. 5 (c). The shaping process visualization is useful because the real cutting process is difficult for the user to see due to critical temperature control of the environment and small pattern size.

CONCLUSIONS

In this research, the cutting force prediction model of micro pattern shaping was studied and the micro pattern cutting simulation software was developed.
The research proposed a cutting force prediction model based on the micro cutting mechanism and the experimental data. The maximum error between the predicted cutting force and experimental data was 0.04 N for the prism pattern and 0.12 N for the rectangular pattern, which means the model agrees with the real data.
The geometrical micro pattern cutting simulation software was also developed for the automatic computation of the cutting section before real machining, which is used to compute cutting force and suggest the optimum cutting condition. The simulator covers prism, rectangular, pyramid, pillar, half circle prisms, various angle prisms and random patterns.

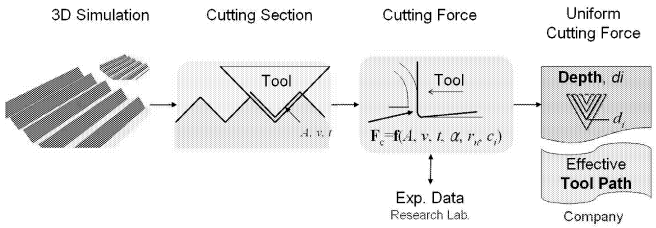


Fig. 5 (a) Flow chart of micro cutting simulation software

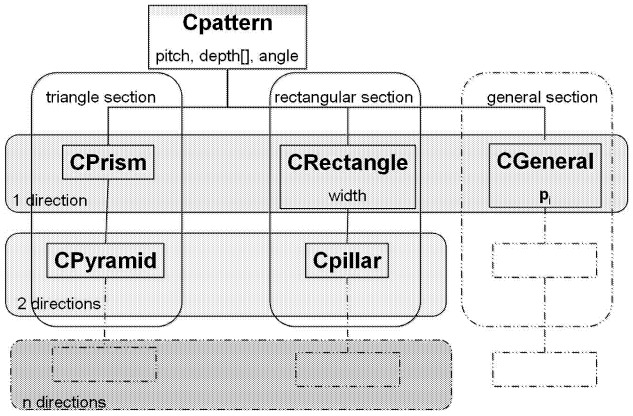


Fig. 5 (b) Data structure of the micro pattern geometry

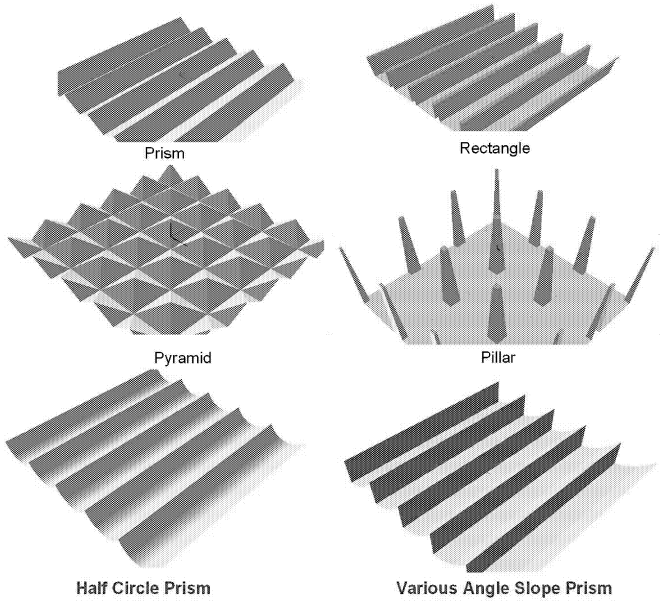


Fig. 5 (c) Visualization of micro cutting simulation results

FOOTNOTES

This research was supported by the Ministry of Knowledge Economy of South Korea. This research will be published in International Journal of Precision Engineering and Manufacturing.

REFERENCES

1. S.M. Hong, T.J. Je, D.J. Lee and J.C. Lee, “Micro machining characteristics of V-shaped single crystal diamond tool with ductile workpiece”, Journal of the Korean Society of Manufacturing Process Engineers, Vol. 4, No. 4, pp. 28~33, 2005.
2. C. Arcona, T.A. Dow, "An empirical tool force model for precision machining", Journal of Manufacturing Science and Engineering, Transaction of the ASME., Vol. 120, pp. 700-707, 1998.
3. P.J. Cheng, J.T. Tsay and S.C. Lin, "A Study on instantaneous cutting force coefficients in face milling", International Journal of Machine Tools & Manufacture, Vol. 37, No. 10, pp. 1393-1408, 1997.
4. Y.S. Yoon and S.J. Lee, “Cutting force prediction in single point diamond turning”, Transactions of the KSME A, Vol. 17, No. 6, pp. 1456-1464, 1992.
5. J. Chae, S.S Park and T. Freiheit, “Investigation of micro-cutting operation”, International Journal of Machine Tools & Manufacture, Vol. 46, pp. 313-332, 2006.
6. X. Lai, H. Li, C.F. Li, Z.Q. Lin and J. Ni, "Modeling and analysis of micro scale milling considering size effect, micro cutter edge radius and minimum chip thickness", International Journal of Machine Tools & Manufacture, Vol. 48, pp. 1-14, 2007.
7. C.K. Ng, S.N. Melkote, M. Rahman and A.S. Kumar, “Experimental study of micro- and nano-scale cutting of aluminum 7075-T6”, International Journal of Machine Tools & Manufacture, Vol. 46, pp. 929-936, 2006.
8. J.D. Kim and D.S. Kim, “Theoretical analysis of micro-cutting characteristics in ultra-precision machining”, Journal of material Processing Technology, Vol. 49, pp. 387-398, 1995.
9. H.Y. Wu, W.B. Lee, C.F. Cheung, S. To and Y.P. Chen, “Computer simulation of single-point diamond turning using finite element method”, Journal of Materials Processing Technology, Vol. 167, pp. 549-554, 2005.
10. J.S. Strenkowski, A.J. Shih and J.C. Lin, “An analytical finite element model for predicting three-dimensional tool forces and chip flow”, International Journal of Machine Tools & Manufacture, Vol. 42, pp. 723-731, 2002.
11. K. Maekwa and A. Itoh, “Friction ans tool wear in nano-scale machining-a molecular dynamics approach”, Wear, Vol. 188, pp. 115-122, 1995.
12. M.F. Aly, E. Ng, S.C. Veldhuis and M.A. Elbestawi, “Prediction of cutting forces in the micro-machining of silicon using a hybrid molecular dynamic-finite element analysis force model”, International Journal of Machine Tools & Manufacture, Vol. 46, pp. 1727-1739, 2006.
13. Z.C. Lin and J.C. Huang, “The influence of difference cutting speeds on the cutting force and strain-stress behaviors of single crystal copper during nano-scale orthogonal cutting”, Journal of Materials Processing Technology, Vol. 201, pp. 477-428, 2008.

COMPLEX ENGINEERING SYSTEMS SIMULATION

BUILDING OF A MOTION FEEDBACK ROLLER COASTER SIMULATOR. A CASE STUDY

Miguel A. Expósito-Sánchez
Ignacio García-Fernández
Miguel A. Gamón-Giménez
Marta Pla-Castells
Rafael J. Martínez-Durá

IRTIC – Universidad de Valencia
C/ Catedrático José Beltrán, 2.
46980, Paterna – Valencia (Spain)
e-mail: miexsan@alumni.uv.es,ignacio.garcia@uv.es

KEYWORDS
6-DOF Motion Platform, Vehicle Simulator, Roller-Coaster, Washout Filters

ABSTRACT

The use of motion systems in vehicle simulators makes it possible to generate acceleration cues and vibrations, allowing for more realistic simulations and therefore a much richer user experience. However, the use of a motion platform introduces an additional difficulty in the development of a simulator. In this paper we illustrate the process of building a motion feedback simulator of a roller coaster, paying special attention to the process of computing the data fed to the motion platform.

INTRODUCTION

Nowadays, virtual reality simulators are quite common in different applications that range from training, both in the civil and the military field, to leisure applications. Among the elements that compose a virtual reality system, an appropriate immersion system can be of crucial importance to get a good feeling of *presence* (Lombard and Ditton 1997).

The immersion system is basically composed by a projection system and other devices, such as haptic interfaces. In addition, when the application involves riding a vehicle, for an improved immersive effect, the screens and simulation controls can be mounted on top of a motion platform, thus adding the sensation of acceleration to the simulated virtual environment.

Motion systems are able to generate acceleration cues and vibrations that provide the user with inertial stimuli. These stimuli feed the equilibrium system of the user with a feeling of acceleration coherent with the movement that is visualized through the screens, allowing for more realistic simulations and therefore a much richer user experience.

Motion platforms can be classified attending to its mechanism (electric, hydraulic) and the number of degrees of freedom they offer, usually ranging from 2 to 6. Most motion platforms use motors or linear actuators that link to the floating frame. Therefore the simulator needs to periodically feed the platform with motion data which translates into an extension or a rotation of each one of the actuators.

This paper focuses on the process of building a simulator with motion feedback and immersive, multi-channel projection system. The system that has been used consists of a 4-channel projection system, driven by a cluster of four PC (one PC per channel). This projection system is mounted atop a 6-DOF Stewart motion platform (see Figure 1).

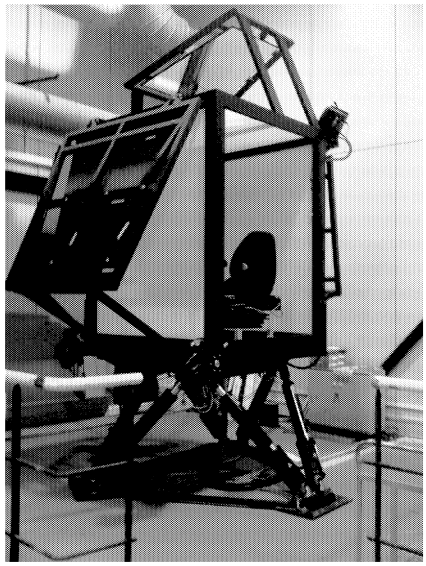


Figure 1: The simulation system that is used in this paper, consisting on a 4-channel projection system mounted atop a Stewart platform.

The application will simulate a roller coaster, as it involves a rather simple dynamic model that, in addition, requires no user actions during the simulation. This makes it easier to focus on the issues related to the use of the motion platform and to the multi-channel projection system.
In the existing bibliography, several authors have addressed the problem of controlling and parametrizing motion platforms in the context of simulation and virtual reality applications. However, these works focus on different techniques related to one of the components of the system. The main goal of this paper is to present a comprehensive description of the process that has been followed to develop the complete system, paying attention to how the different subsystems are related.
The paper is structured as follows. First a brief overview of roller coaster simulators is presented. Then, the system architecture is described, together with a review of its main components. Next, the techniques that have been used to feed the motion system are explained. Finally, some results are presented, together with a review of the ongoing and future work.

Roller coaster simulation

Typical roller coaster rides are essentially some kind of rail road system that rise in designed patterns on a support structure. Although not necessarily, the track is usually a closed circuit. Also, roller coaster simulators are rather common in the video-game field. Usually, they are 3D graphics applications which represent the roller coaster track by means of some kind of curve (most often, polynomials) within a virtual landscape. The car rides along the track using a dynamic model to determine its position during the simulation.
Rollercoaster simulators are becoming more and more popular games, the most widespread ones are NoLimits Rollercoaster (NoLimits 2010) Simulation and Rollercoaster Tycoon (Tycoon 2010). They both allow the user to build his own roller coaster ride and to test it. Whereas No Limits features realistic coaster physics with enhancements such as friction and audio with doppler effect as well as a nice editor, Roller Coaster Tycoon aims to simulate a whole amusement park with different kinds of rides.
The simulator that has been developed in this paper is based in the roller coaster simulator called *Rollercoaster 2000* (RC2K) (Plusplus 2010). The main reasons that led us to choose this application are the fact that it features a nice environment with a simple physics model, which can be easily removed or replaced, and its configurable track description format that makes it possible to modify the curves that define the circuit. In addition, RC2K source code is available under a GPL license (GNU 2010) which allows the modification of the code to derive new applications.
In our simulator, the track definition and the rendering engine of RC2K have been used with minor modifications. The dynamic model, however, has been removed, and replaced by a new one which runs separately to feed both the camera position in the 3D environment and the motion system.
As the track model used in RC2K is the basis of the system developed here, it is described next in some detail.

Track description in Rollercoaster 2000
RC2K models the roller coaster track with Bézier curves defined by means of control points stored in a plain text file. A Bézier curve is a polynomial that joins two points by means of a set of additional control points (Lengyel 2004). One of the properties of Bézier curves is that they allow to set the tangent directions of the curve at its ends. This makes it possible to use Bézier curves to interpolate a succession of points obtaining a continuous curve that also is continuously differentiable.
To define the track, it is first discretized into a succession of nodes together with the track tangent direction at every one of these points. These data are used to compute cubic Bézier curves that interpolate all the points by introducing two extra control points for every original track point. The circuit is automatically closed by linking the end of the last curve with the beginning of the first one. Banking is automatically computed attending to the curvature and torsion of the track.

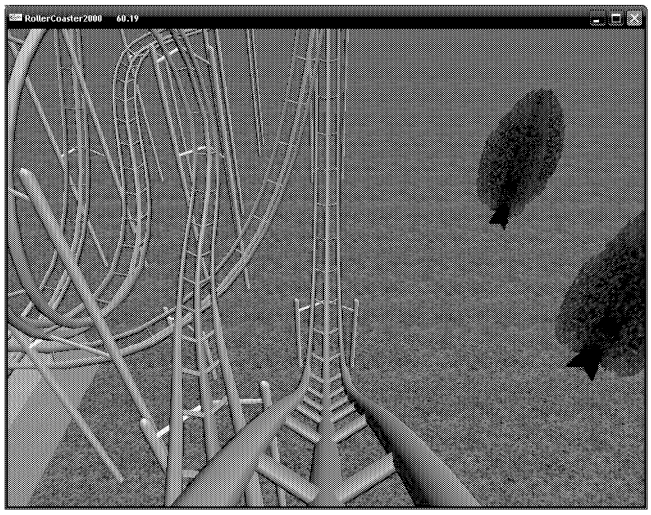


Figure 2: A screen capture of the roller coaster RC2K application

When the RC2K simulator starts, it first parses the track definition text file. Then, the extra control points are computed and the interpolating curve is generated. Using this curve, a new discretization of the track is performed. In this case a fixed segment length, which is usually much smaller than the one of the original discretization, is used. Finally, this new discretization is interpolated by means of straight cylinders that are used

for rendering. Figure 2 shows a screen capture of the RC2K application with the discretized track.
Once the track model has been created, the scene is completed with a skybox, a textured ground plane and some trees whose coordinates are defined also in the track definition file.
After describing the track definition that will be used, next the system proposed is described in detail, paying special attention to the main technical issues that come up during its design and implementation.

SYSTEM ARCHITECTURE

A virtual reality simulator is a rather complex system which involves several interconnected modules. Our roller coaster simulator architecture is composed of four modules:

- Geometric track description
- Dynamic model
- 3D graphics visualization system
- 6-DOF motion system

During the simulation, the geometric track description is used, on the one hand, for rendering the 3D view of the environment and, on the other hand, for computing the trajectory of the car using the dynamic model. The output of the dynamic model, in the form of a position and orientation of the car in space, is sent to the rendering engine to decide the camera position. In addition, a set of accelerations and angular velocities are sent to the motion platform to produce the inertial stimuli. Figure 3 shows a diagram of the different elements and their relationships. In what remains of this section a more detailed description of every module is presented.

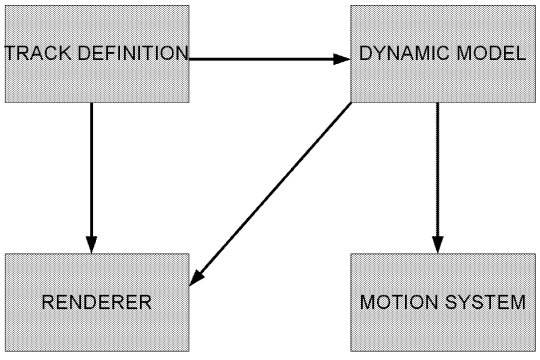


Figure 3: The four main modules that compose the system and the relationships between them.

Geometric track model

In our roller coaster simulator, we rely on Rollercoaster 2000 for the definition and visualization of the track. As it has been described previously, in RC2K the track of the roller coaster is defined by means of a succession of Bézier curves in space that interpolate a series of points. However, in the original track model, the final step is a linear interpolation of a fine-grained set of points along the curve. But this approach is not adequate for the dynamic model, as it causes discontinuities in the derivative of the car orientation which, in turn, would cause a rough movement of the Stewart platform. For this reason, in the geometric description that feeds the dynamic model the complete interpolation using Bézier curves is retained, avoiding the final discretization step. This modification leads to a much smoother behaviour of the car and of the motion system.

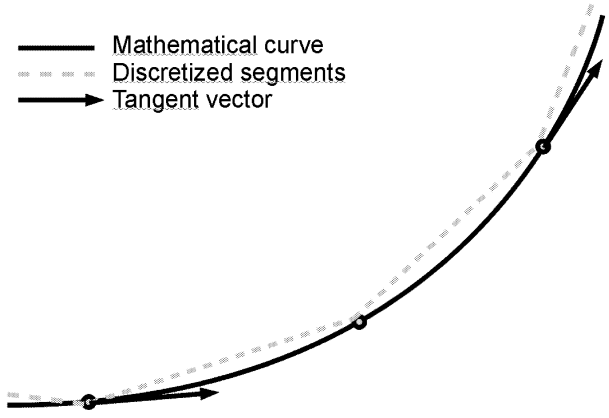


Figure 4: In order to increase smoothness of the accelerations, the Bézier curve is used for the dynamic model, as the use of the linear interpolation has *corners* which lead to discontinuities in the first derivative.

Dynamics of the vehicle

In order to obtain the dynamics of the vehicle along the track we need a model that provides us with its velocity at every instant of time. Then, given that we know this value of the velocity, the position of the vehicle can be integrated numerically.

Conservative system
If the system is conservative, then energy conservation principle can be used to obtain the vehicle velocity (Eberly 2004). For a body of mass m, its total energy can be written as

$$E = \frac{1}{2}mv^2 - mgh$$

which must be constant along time. Thus, computing the value for E at the beginning of the simulation

$$E_0 = \frac{1}{2}mv_0^2 - mgh_0,$$

with v_0 and h_0 the initial velocity and height of the body, the velocity at any height h can be obtained as

$$|v| = \sqrt{2mgh + E_0}. \quad (1)$$

Note that this equation has no solution for heights greater than the starting height $h > h_0$, which is a consequence of the energy conservation principle.

Non conservative system
By using expression (1) we are considering an ideal, conservative system. However, we are interested in modelling friction in the vehicle dynamics, to simulate air drag as well as wheel-track friction. On order to introduce this effect, we will use the difference in height between two consecutive simulation steps, instead of the absolute height of the vehicle.
Let's consider that the vehicle has a velocity v_1 and it is at height h_1. Its total energy in that configuration is $E_1 = \frac{1}{2}mv_1^2 - mgh_1$
After a small interval of time Δt, it has moved to a new height $h_2 = h1 + \Delta h$ and, due to friction or energy gain (we could consider a driven roller coaster) it has a new energy $E_2 = E_1 + \Delta E$. Thus, from the expression of the total energy,

$$\Delta E = E_2 - E_1 = \frac{1}{2}(v_2^2 - v_1^2) - mg\Delta h \quad (2)$$

If we consider that the variation of energy can be approximated as the average energy change rate, $\dot{E}$, multiplied by the time interval Δt, then we can approximate the new value of the velocity, v_2 as

$$v_2 = \sqrt{v_1 + 2\Delta t\dot{E} + 2mg\Delta h} \quad (3)$$

From this expression we can directly obtain the value of v_2 given the previous value of the velocity, the height increment of the body at a given instant of time, Δh, and the energy change rate, $\dot{E}$.
For the case of air drag, the energy change rate can be expressed as

$$\dot{E}_d = AC\frac{1}{3}v^3$$

where A is the area of the section of the body which is perpendicular to its movement direction and C is the air-object drag coefficient, which depends on the properties of the object (Wong 2001).
The model described by (3) is similar to the equation used in the original RC2K evolution model, although, in that case, it was a conservative model that did not consider any kind of friction.

Vehicle simulation
The movement of the vehicle along the track is driven by a state machine. At first, the train is stopped at the station and, then, it advances at a constant speed while going up trough the lift chain. The machine advances to the next state when the difference in height from the current segment to the next is less than zero, i.e when the wagon starts going downhill. This usually occurs at the top of the lift slope. In this state, the speed of the train at the next step is computed using (3), and the new position is obtained by numerical integration using an explicit Euler scheme.
The dynamic model is run in a separate application that uses the RC2K geometric track description with the aforementioned modifications to increase smoothness.
In addition, the use of a separate process to run the dynamic model makes it possible to uncouple the rendering and simulation steps. While in the original RC2K application the dynamic model evolved one time per frame, in our application it is executed at 60Hz, because it has to feed the motion platform at this frequency.

Graphics rendering

In order to obtain a visualization of the virtual environment throughout the projection system, a graphical view of the roller coaster and of the landscape needs to be rendered.
As well as the track geometric model, the core of the graphics rendering engine is inherited from the original RC2K system. However, in this case several modification needed to be done to adapt it to the distributed projection system that has been used. Next, the main properties of the modified rendering system are described.

Four-channel rendering
The available facility for rendering is a 4-channel CAVE system mounted on top of the motion platform. The system is run by means of four PC machines, interconnected by a Giga Ethernet network, which generate the view of one channel each. An additional machine acts as a master by coordinating and synchronizing the simulation.
For this reason, the main requirement for any application to run in this projection system is that a separate instance of the program need to be executed in every one of the four machines.
The cluster uses the OpenGL-VRJuggler suite for cluster-based real-time rendering (Just et al. 1998, Bierbaum et al. 2008). VRJuggler automates the process of the proper set-up of the projections that are rendered according to specific physical screen dimensions and positions. When an VRJuggler application starts, the view and projection matrices are already configured for the user who only has to send graphics primitives.
Also, it makes it possible to share simulation data across the cluster network to synchronize all the views of the

projection system so that they show the same state of the environment at every instant of time. These data are stored in a shared memory area in every node of the cluster, helping in the process of distributing the application to be run.
In order to integrate the graphics engine of RC2K in the distributed system, the dynamic engine has been removed from the application and substituted by a routine that reads the necessary data (position and orientation of the car in the environment) from the shared memory area.
A separate process is run at every node of the cluster, to render the scene of the corresponding channel using the incoming data. All the scene information, such as the track definition, textures, or any other geometric model included in the scene is replicated in every machine.

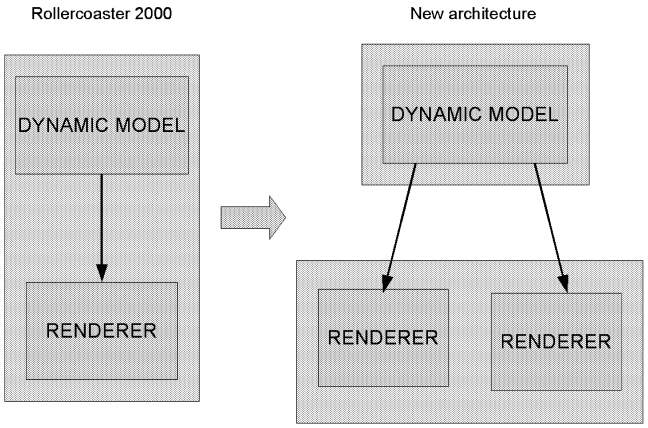


Figure 5: The architecture has migrated from a monolithic application to a client-server architecture in which the dynamic model serves simulation data to the rendering nodes of the cluster.

In the master node, the dynamic model is run at 60Hz using the high resolution version of the track and acting as a server for the four rendering nodes. Figure 5 shows the modification in the system architecture from the original RC2K application to the system proposed here. The data that are generated by the dynamic model are also sent to the motion system which is described next.

MOTION SYSTEM

Adequate motion perception is a crucial factor when enhancing the immersion of the user into a virtual environment. It provides the user with consistent stimuli through vision and through the vestibular system, responsible for equilibrium, thus giving a complete feeling of movement. Indeed, the absence of an adequate stimulation of the vestibular system not only reduces the feeling of realism but also can cause the user to feel dizzy.

In this manner, the motion platform needs to be commanded by a control algorithm, which transforms the trajectories generated by the dynamic model into actuator movements that are able to provide the rider with realistic motion.
However, as the motion of the platform along each axis is limited by mechanical constraints, low frequency accelerations would make the platform to reach its movement limits (Barbagli et al. 2002). Consequently, the use of adequate motion cueing algorithms that take into account such limitation is necessary (Gutridge 2004).
The motion platform used in this work is controlled by the so called Washout Filters, which consist of a series of filters that separate the input data into high and low frequency components. Among the available methods these are the most common algorithm used in simulation (Reid and Nahon 1985; 1986). After a brief overview of this method we explain how they are applied to our roller coaster simulation.

Filtering

Essentially, the Washout Filters act as follows: High frequency accelerations and angular velocities are passed to the platform, which reproduces them by means of the adequate vibrations. On the contrary, low frequency accelerations cannot be reproduced by means of platform displacements, as the it would quickly reach its limits. For this reason, linear acceleration is substituted by what is called *tilt-coordination*: A horizontal acceleration is got by tilting the cockpit of the simulator until the projection of the gravity force in the cockpit horizontal plane has the same magnitude. In absence of input accelerations it slowly brings the motion device to its zero reference position (Grant and Reid 1997). Figure 6 shows a diagram of the Classic Washout Filters.
If the user is immersed in a projection system in which there is no reference of its actual movement in the simulation room, the user would perceive all these vibrations and tilt angles as accelerations, and not as rotations or displacements.

Computing the filters inputs

In order to use the Washout Filters, the dynamic model has to be able to compute the linear accelerations of the vehicle, together with the angular velocity of the user's reference frame.
The linear accelerations can be obtained differentiating numerically the tangent vector to the curve, which is computed at every simulation step, in order to determine the orientation of the camera. The result of this differentiation is passed to the washout filter.
But to obtain the angular velocity of the curve reference frame we need to refer to differential geometry. For a given curve, $\mathbf{c}(s) \subset \mathbb{R}^3$, Frenet's trihedron defines the tangent vector $\mathbf{d}_1(s)$, the normal vector $\mathbf{d}_2(s)$, which

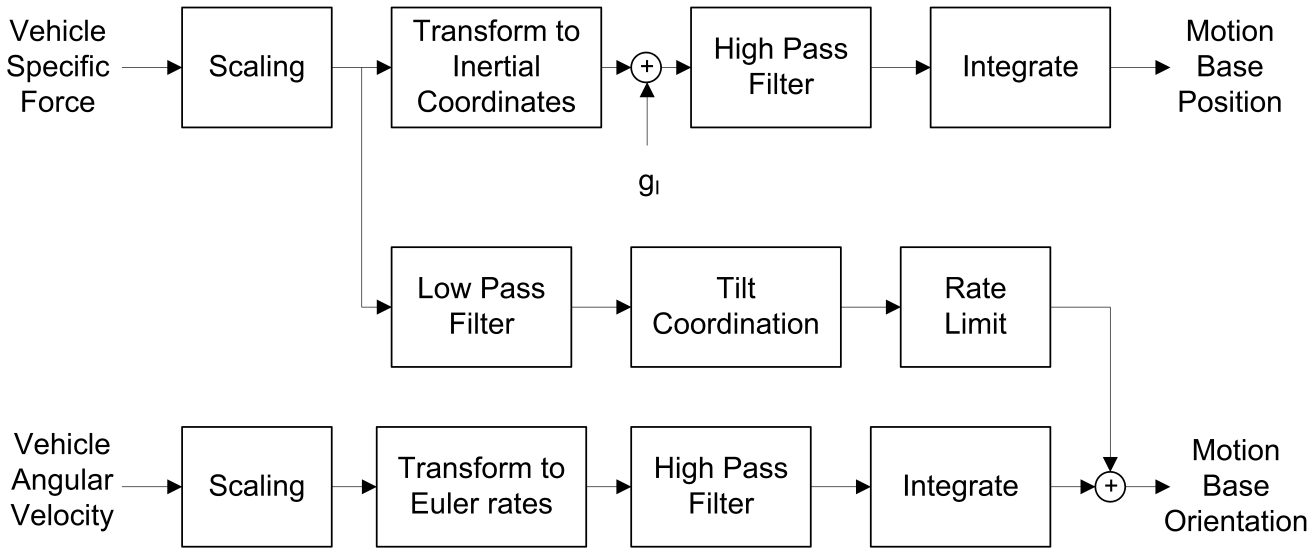


Figure 6: Diagram of the classical washout filter used to control a motion platform.

is perpendicular to the curve, and the binormal vector $\mathbf{d}_3(s)$, which completes an orthonormal, positively oriented, basis of the three-dimensional space. Frenet's trihedron is, indeed, what is being used to orient the camera in the roller coaster simulation.
Thus, given this basis of $\mathbb{R}^3$, the so called Darboux vector represents the angular velocity of a body whose reference frame is exactly our Frenet's trihedron, when parameter s is considered as time. This vector is defined as

$$\omega = \frac{1}{2}\sum_{k=1}^{3} \mathbf{d}_k(s) \times \mathbf{d}'_k(s) \tag{4}$$

where $\mathbf{d}'_k$ represents the derivative respect to s of vector $\mathbf{d}_k$ (Spillmann 2008).
The computation of Darboux vector is done as follows. First, Frenet's trihedron $< \mathbf{d}_1(s), \mathbf{d}_2(s), \mathbf{d}_3(s) >$ is obtained from the Bézier curve that defines the track. Then, differentiating with the previous value of the vectors $< \mathbf{d}_1(s-\delta s), \mathbf{d}_2(s-\delta s), \mathbf{d}_3(s-\delta s) >$, their derivatives are obtained and ω is computed by means of (4). Once it has been computed, ω is passed to the washout filters, expressed in the user's local reference frame, to compute the tilting-coordination.
Instead of using using numerical differentiation, the analytical expression of the Bézier curve's could have been used to obtain Frenet's trihedron. However, the track is a piece-wise defined curve. Thus, around every sub-interval change, numerical differentiation still would have been necessary to obtain the associated vectors. For this reason, we decided to use numerical integration all the track long.

RESULTS

The system described has been successfully built and tested. During the development of the simulator, one of the main issues that emerged was the use of linear interpolation in the final description of the track in the original RC2K application. In order to prevent discontinuities in the accelerations, which would lead to an inadequate movement of the platform, Bézier curves have been used to compute the dynamic model accelerations, improving the behaviour of the motion system.

Other aspect that needed to be improved was the rendering stage of the application. From a single screen system we migrated to a cluster of four computers running one visual simulation in every one of them. This has made it possible to reproduce four views of the scenery, with active stereographic projection, creating a highly immersive environment.

In order to prevent synchronization problems between the different visualization channels, a single dynamic model was used to serve the state information to the four cluster nodes.

Preliminary evaluation

Some informal tests have been performed with a set of users that ran the simulator. After two runs in the roller coaster, we had a brief interview with every user in order to know their impression.
The main conclusions after this test is that the overall impression is good, and all of them said that they did have the feeling of being in a roller coaster. The main limitations were found in the points where the track had a peak, where most users said that they missed a stronger vertical acceleration, and in very close curves, where a higher lateral g-force was expected.

CONCLUSIONS

In this paper we have addressed main technical issues that came up during the design and implementation of a roller coaster simulator with a motion feedback platform in a distributed architecture. Taking the existing software Rollercoaster 2000 as a starting point, several modifications have been introduced in order to run the simulator in a cluster of PC and to reach the requirement of the motion platform: smooth accelerations and at least a 60Hz update frequency.
After an informal evaluation by several users, it has been observed that the simulator runs in the available equipment with good results, providing good feeling of being in a real roller coaster. Some limitations were detected by the users, however. Additional tests have to be done in order to determine whether they are due to the range of accelerations that the motion platform is able to reproduce or, on the contrary, they are caused by the parametrization of acceleration filters.
The presented paper, however is still work in progress and several improvements and tasks are to be performed. A new graphics engine is being built from scratch, which benefits from the latest shading techniques available in modern graphics cards. Also, a rigid body model will be developed in order to consider the effect of inertia in the dynamics of the wagon. From the evaluation point of view, a set of tests is being prepared in order to evaluate formally the quality of the simulator and to identify its main weaknesses.

REFERENCES

Barbagli F.; Ferrazzin D.; Avizzano A.; and Bergamasco M., 2002. *Washout filter design for a motorcycle simulator.* In *Proceedings of the IEEE Virtual Reality Conference (VR 2001)*. 225–232.

Bierbaum A.; Just C.; Hartling P.; Meinert K.; Baker A.; and Cruz-Neira C., 2008. *VR juggler: a virtual platform for virtual reality application development.* In *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses*. 1–8.

Eberly D.H., 2004. *Game Physics.* Morgan Kaufmann.

GNU, 2010. *GNU Affero General Public License version 3.* URL `http://gplv3.fsf.org/`.

Grant P. and Reid L., 1997. *Motion Washout Filter Tuning: Rules and Requirements. Journal of Aircraft*, 34, no. 2, 145–151.

Gutridge C., 2004. *Three degree-of-freedom simulator motion cueing using classical washout filters and acceleration feedback.* Master's thesis, Virginia Polytechnic Institute & State University.

Just C.; Bierbaum A.; Baker A.; and Cruz-Neira C., 1998. *VR Juggler: A Framework for Virtual Reality Development.* In *Immersive Projection Technology Workshop.*

Lengyel E., 2004. *Mathematics for 3D Game Programming & Computer Graphics.* Charles River Media, 2 ed.

Lombard M. and Ditton T., 1997. *At the Heart of It All: The Concept of Presence. Journal of Computer-Mediated Communication*, 3, no. 2.

NoLimits, 2010. *No Limits Rollercoaster Simulator website.* URL `http://www.nolimitscoaster.com/`.

Plusplus, 2010. *Plusplus' rollercoaster 2000 website.* URL `http://plusplus.free.fr/rollercoaster/index.html`.

Reid L. and Nahon M., 1985. *Flight simulation motion-base drive algorithms: Part 1 - developing and testing the equations.* Tech. rep., University of Toronto Institute for Aerospace Studies (UTIAS).

Reid L. and Nahon M., 1986. *Flight simulation motion-base drive algorithms: Part 2 - selecting the system parameters.* Tech. rep., University of Toronto Institute for Aerospace Studies (UTIAS).

Spillmann J., 2008. CoRdE*: Cosserat Rod Elements for the Animation of Interacting Elastic Rods.* Ph.D. thesis, Albert-Ludwigs-Universitt Freiburg im Breisgau.

Tycoon, 2010. *Rollercoaster Tycoon website.* URL `http://www.rollercoastertycoon.com/`.

Wong J., 2001. *Theory of ground vehicles.* John Wiley & Sons, 3 ed.

BIOGRAPHY

MIGUEL A. EXPOSITO-SANCHEZ studied informatics at University of Valencia and now he is working with a grant at IRTIC, mainly in virtual reality applications for training.

IGNACIO GARCIA-FERNANDEZ studied mathematics at University of Valencia. He had his PhD in Computer Science in 2009 and holds a position of lecturer at Informatics Departament of University of Valencia. He is researcher at IRTIC and his main interest is physics based interactive simulation for training.

MIGUEL A. GAMON-GIMENEZ studied informatics at University of Valencia. Since 2001 he is researcher at IRTIC and has worked in the development of several washout filters for motion platforms.

MARTA PLA-CASTELLS studied mathematics at University of Valencia and had her PhD in Computer Science in 2009. She is lecturer at Computer Science department in Universty of Castellon and researcher at IRTIC. Her main research interests are Virtual Reality applications for training and interactive simulation.

RAFAEL J. MARTINEZ studied informatics and had his PhD in Computer Architecture in 1999. Since 2000 he is the head of simulation group LSYM at IRTIC, and has leaded several research projects within the field of simulation for training.

MODELLING AND SIMULATION OF SEVERAL INTERACTING CELLULAR AUTOMATA

Marta Pla-Castells
Ignacio García-Fernández
Rafael J. Martínez-Durá

IRTIC – Universidad de Valencia
C/ Catedrático José Beltrán, 2.
46980, Paterna – Valencia (Spain)
email: marta.pla@uv.es, ignacio.garcia@uv.es, rafael.martinez@uv.es

KEYWORDS
Cellular Automata, Neighbourhood, Modelling, Simulation, Interaction

ABSTRACT

Cellular Automata are used for modelling and simulation of many systems. In some applications, the system is formed by a set of subsystems that can be modelled separately, but, in such cases, the existence of interactions between these subsystems requires additional modelling and computer programming. In this paper we propose a modelling methodology for the simulation of a set of Cellular Automata models that interact with each other. The modelling methodology is described, together with an insight on implementation details. Also, it is applied to a particular Cellular Automata model, the Sanpile Model, to illustrate its use and to obtain some example simulations.

INTRODUCTION

Modelling and simulation of spatially distributed physical phenomena is one of the most common applications of Cellular Automata (Chopard and Droz 2005). In such cases, the evolution of the automata is usually related to the exchange of some kind of material, information or energy between adjacent cells.

The simulation of this kind of systems using a Cellular Automata requires the representation, by means of the Cellular Automata grid, of the domain in which the system is defined. This can be done by means of the definition of different types of cells in order to simulate boundary conditions. However, there are situations in which the use of a unique Cellular Automata to represent the whole domain is not straightforward.

Consider, as an example, the system of Figure 1: It is composed by two flat surfaces, one atop the other, and a fluid that is spread on the upper one. As soon as the fluid reaches the edge of the upper subdomain, it will start falling onto the lower one, thus exchanging matter from one subsystem to the other.

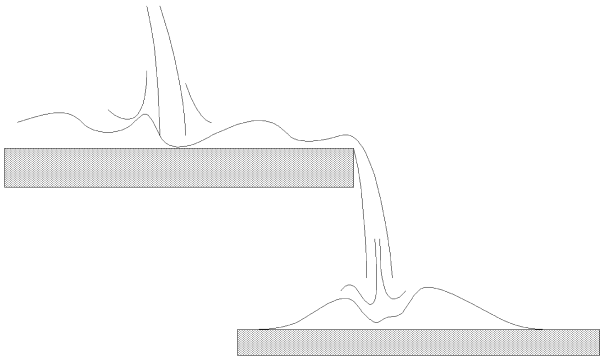


Figure 1: An example in which two subsystems exchange matter. This situation can be found in many applications of the Cellular Automata methodology.

Situations equivalent to this example can be found in different applications. Problems such as crowd dynamics in evacuation simulation (Zhou 2009, Pelechano and Malkawi 2008), traffic simulation (Aponte and Moreno 2006), lava and fire spread (Vicari et al. 2007, Alexandridis et al. 2008) or tumour propagation (Ribba et al. 2006) may involve the simulation of several interconnected levels and sub-regions. If the simulation domain can be easily decomposed in several subdomains, then a set of independent Cellular Automata models can be used. But, in this case, the interaction between the different subdomains has to be defined and simulated. Although some authors have used the simulation of more than one domain, the issue of connecting the subdomains is solved by modelling the interactions in a custom manner adequate to the particular problem Ribba et al. (2006), Pelechano and Malkawi (2008). However, a general methodology for this problem has not been addressed.

In this work a modelling methodology is proposed that

makes it possible to handle such situations in a very straightforward and general way. The paper is structured as follows. In what remains of this section, some definitions and notations that will be used later in this work are presented. Then, the modelling methodology for several interacting automata is presented. Next, several implementation details are given in the form of data structures and general procedures. Finally, the methodology proposed is applied to a particular example, the Sandpile Model, showing several particular applications.

Cellular Automata Formalization

In order to present the proposed scheme, first we state the basic notation that is used within this work. A classical two dimensional Cellular Automata based on a regular grid is considered. Every cell of the grid has a set of variables that describe the state of every cell during the simulation.
The evolution of every cell is governed by a local rule that, without loss of generality, will be considered the same for every cell. In order to determine the state change of a cell, the local rule takes the current state of the cell itself and of a group of cells of the grid, which are called its neighbourhood.
A Cellular Automata can be formally described as

$$CA = \langle \mathbf{M}, \mathbf{V}, \mathbf{S}, \boldsymbol{\varphi} \rangle \tag{1}$$

where

- $\mathbf{M} = \{(i,j) \in [1,N] \times [1,M] \subset \mathbb{N} \times \mathbb{N}\}$ is the set of cells that form the Cellular Automata.
- $\mathbf{V}$ is the neighbourhood of a cell.
- $\mathbf{S}$ is the set of possible states for a cell.
- $\boldsymbol{\varphi} : \mathbf{S}^{|\mathbf{V}|+1} \rightarrow \mathbf{S}$ is the local rule or the transition rule of the Automata.

The neighbourhood $\mathbf{V}$ of cell (i,j) will be considered as a subset of its adjacent cells

$$\mathbf{V} \subseteq \mathbf{Adj} := \{(i+k, j+l) : k,l = -1,0,1\} \setminus \{(i,j)\}.$$

This definition of neighbourhood includes the Moore neighbourhood, which corresponds to the complete set $\mathbf{V}_{moore} = \mathbf{Adj}$, and the Von Neumann neighbourhood, $\mathbf{V}_{vn} = \{(i-1,j),(i+1.j),(i,j-1),(i,j+1)\}$ (see Figure 2). Von Neumann and Moore neighbourhoods are the most common ones in the Cellular Automata literature (Chopard and Droz 2005).
The set of possible states of a cell, $\mathbf{S}$, is the Cartesian product of the sets that host the state variables of a cell. In the classical definition of Cellular Automata, such spaces are usually formed by a finite set of elements (Chopard and Droz 2005), although many applications consider more general sets, including real values.

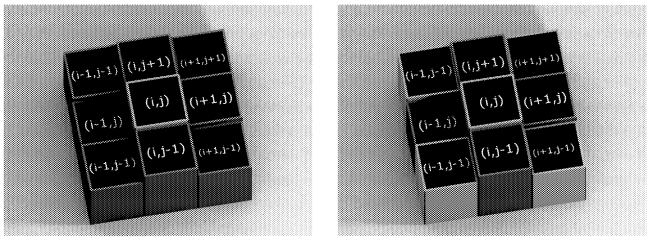


Figure 2: The figure shows the Moore neighbourhood (left) and the Von Neumann neighbourhood (right), which are used in many Cellular Automata models. The cells that belong to the neighbourhood are coloured in red.

The local rule $\boldsymbol{\varphi} : \mathbf{S}^{|\mathbf{V}|+1} \rightarrow \mathbf{S}$ is a function that obtains the evolution of cell (i,j) using its state and the state of the cells that belong to its neighbourhood. This function, together with the cell's state variables, is what actually defines the dynamic model of the system.

This notation will be used in the next sections to propose a modelling methodology for a system with several interacting subdomains. Each subdomain will be modelled with an independent Cellular Automata, and the different submodels will be interconnected.

CELLULAR AUTOMATA INTERACTION

Let's consider a Cellular Automata model to simulate the movement of a crowd of people in a building. The Cellular Automata grid represents a room and the state of each cell is related to its occupancy. A transition rule can be defined to simulate the behaviour of a crowd during an evacuation (Zhou 2009). In such a model, the cells that represent an staircase or a gate also *throw occupancy* out of the system.

Another example can be found in the Sandpile Models (Bak et al. 1987). In the Sandpile Model, cell state represents the amount of sand that is hosted in the cell, and transitions represent material flow in a granular system. A sandpile model is said to be *open* if sand exits the system when it reaches the grid frontier. In this case, the frontier cells have the particular property that can *send material* outside the system. If we consider a system that involves several sandpiles, then the outgoing flow of one sandpile can fall on another sandpile, becoming an input flow. This model will be discussed in more detail later on.

Indeed, we can find examples in industrial processes in which not only there exists a relationships between different subsystems but, in addition, these relationships vary with time. In the case of sandpiles this would happen if the relative position of the various subsystems vary with time; e.g a set of conveyor belts to transport bulk material, or an excavator loading sand into a truck.

Interaction modelling

The previous examples show that, if we have several subsystems modelled by means of Cellular Automata, we can be interested in simulating an exchange of matter, energy or other type of information between them. Moreover, the relationship between the different subdomains can vary along time during the simulation.
In order to model such situations we propose a methodology that is based in the following basic steps:

- Identification of the cells of the different automata that act as connection cells.
- Definition of models for the interactions that are to be simulated.
- Definition of a rule, or set of rules, that determine which cells need to be connected to simulate the interactions.
- Monitoring of the existing connections during the simulation, to modify or remove them.

Next, these four steps are explained in more detail.

Identification of emitting cells
In order to formalize the problem, and to allow a systematic definition of the procedure, first we are going to define an special type of cells, which will be called *emitting cells*.
An *emitting cell* will be any cell of a Cellular Automata that, due to the problem definition, is able to throw information out of the system and, potentially, send it to a cell of a different automata.
The first step of the methodology is the identification of such cells in every one of the involved Cellular Automata. This identification will depend on the properties of the system that is being simulated and on the kind of relationship that is to be modelled.

Modelling of the connections
The second step is to define a model for the information exchange that has motivated the definition of the *emitting cells*. The model will depend on the particular problem and on the features that we need to represent with our Cellular Automata model.
However, in many cases the information exchange will be the same that also takes place between adjacent cells during the update of the Cellular Automata. In this case, a general approach can be used that greatly simplifies the simulation phase.
When the interaction between an emitting cell and another Cellular Automata follows the same model as the one described by the local rule, the easiest way to set the relationship is the modification of the neighbourhood of the emitting cell. By doing this, the simulation of the interaction is featured when the local rule, φ, is applied to the emitting cell during the update of the Cellular Automata.

Configuration of the system of Cellular Automata
Once the *emitting cells* have been identified, and the relationships have been modelled, it is necessary to define which cells need to be connected to each other. This will be done by means of a set of rules and criteria that decide whether an emitting cell has to be connected to a certain cell of another Cellular Automata.
As it happens with the emitting cells, the identification of these criteria will depend on the properties of the system that is being modelled. However, they will be often be related to the position of the cells, due to the kind of systems that are usually modelled by means of cellular automata.

Update of the connections
It has been illustrated previously that the interaction between two or more Cellular Automata can change during the simulation as a result of, for example, the displacement of the different subsystems.
Thus, the connection rules will be used before the beginning of the simulation, to configure the simulation scenario, but it can be necessary to check them also after every simulation step to decide whether any relationship needs to be added, modified or removed.
After presenting the main steps of the proposed methodology, next we overview the way they can be applied to perform a simulation that involves several interacting Cellular Automata.

Interaction simulation

Cellular Automata are a computational approach to modelling systems and, often, they are used to simulate the modelled systems by means of a computer program. The previous modelling scheme can be applied very easily to an existing Cellular Automata computer model following the next steps:

1. Build a list of Cellular Automata that contains all the automata that are to be simulated.
2. In the data structure used to implement a Cellular Automata, build a list that holds all the *emitting cells* of the automata.
3. Either
   (a) implement the functions that decide if an emitting cell needs to be connected to any cell, or
   (b) set up manually the connections of emitting cells.
4. If the local rule of the automata is not enough to simulate the connections, then implement the necessary models.

5. Run the simulation.

   (a) Update every automata. Call the connection models, when necessary.

   (b) After every step run the procedures that update the connections.

Next section presents in more detail the data structures that allow the implementation of the cell connections and the simulation of the relationships.

CELL NEIGHBOURHOOD IMPLEMENTATION

In this section, a set of data types and procedures are presented. They are used to store the Cellular Automata information and to define and implement the neighbourhood of a cell in such a way that foreign cells (cells from another Cellular Automata) can be used during a simulation. For sake of simplicity they are presented for a one-dimensional Cellular Automata, in which cells are arranged in a linear array. Later in this section the generalization from this case to the two-dimensional case is discussed.
The data structures and the procedures that are presented next have been written using the C programming language conventions, especially in what regards pointers, which are expressed with asterisks after the type name, although they can, obviously, be translated into any programming language.

Overall Description of the Implementation

Usually, a Cellular Automata is implemented by means of a sequential data structure that stores the cells' data. When implementing the local rule, the neighbourhood is defined by means of explicit reference to the indexes of the elements of the array.
In opposition to this approach, and in order to allow the communication between different Cellular Automata, we propose the use of a data structure to represent cells that uses pointers to refer to the cells that belong to its neighbourhood. With this cell data type, the Cellular Automata implementation will become a multiply linked data structure, where the different links define the neighbourhood relationship.
Using this implementation, and once it has been decided that a cell must be connected to another one, the connection will be as simple as a pointer assignment.

Dynamic Neighbours in One Dimension

Let's consider a one-dimensional Cellular Automata with N cells. In this case, the grid is an array of cells $\mathbf{M} = \{(i);\ i = 1, \ldots, N\}$ and the neighbourhood of cell (i) is formed by its adjacent cells $\mathbf{V}(i) = \{(i-1), (i+1)\}$.

Cell Data Structure
In order to store the information of a cell, a basic data structure, that will be called `CellState` is needed. This structure will not be defined here, as it is composed by the parameters and state variables of a cell, and it depends on the particular application of the Cellular Automata. In order to store the complete grid of a Cellular Automata, an array of `CellState` structures is used. This array could be defined as

```
CellState CA_Grid0[N_CELLS];
```

for a Cellular Automata of size `N_CELLS`.
Computing the evolution of cell (i), which is stored in the data structure `CA_Grid0[i]`, involves the state of its neighbour cells. The simplest way of accessing such cells' information is by explicit indirection of the array elements `CA_Grid0[i-1]` and `CA_Grid0[i+1]`. But, this assumes that the neighbourhood relationship can only happen between adjacent cells of the same Cellular Automata, and it imposes a limitation in the situation described in the introductory section of this paper.
In order to overcome this limitation we propose to use pointers in order to refer to the neighbours of a cell. This new data structure will make it possible to modify very easily the neighbourhood of a cell to make any other cell to become its neighbour, no matter its position in the grid and, even, whether it belongs to the same Cellular Automata or not. Using this idea, the basic cell data structure is as follows:

```
struct Cell {
  CellState StateInfo;
  Cell * Prev;
  Cell * Next;
};
```

Next, each attribute of the `Cell` structure is described briefly:

`StateInfo` An structure of type `CellState`, that contains the basic information for a cell, including its state variables.

`Prev` A pointer to a `Cell` structure, that stores the data of the neighbour that is to the left of the cell, usually cell $(i-1)$.

`Next` A pointer to a `Cell` structure, that stores the data of the neighbour that is to the right of the cell, usually cell $(i+1)$.

Again, the Cellular Automata grid can be defined by means of an array, but now using the new data type, as

```
Cell CA_Grid[N_CELLS];
```

The `Cell` structure is general, in the sense that it allows the implementation of the Cellular Automata in the usual way, just by pointing `Prev` and `Next` to their default values, the structures of cells $i-1$ and $i+1$ respectively.

Basic Methods
Next, three basic methods are defined to show the use of the `Cell` data structure to define dynamic neighbours. First, procedure `Reset_Neighbours()` makes it possible to use the `Cell` structure in the traditional way, setting its neighbours as its adjacent cells.

```
Reset_Neighbours(int i)
{
  CA_Grid[i].Prev = &CA_Grid[i-1];
  CA_Grid[i].Next = &CA_Grid[i+1];
}
```

If, on the contrary, we are interested in linking cell (i) to another arbitrary cell, no matter whether it belongs to the same automata or not, the following procedures can be used.

```
Set_Neighbour_Prev(int i,Cell *_c)
{
  CA_Grid[i].Prev = _c;
}

Set_Neighbour_Next(int i,Cell *_c)
{
  CA_Grid[i].Next = _c;
}
```

Cell Evolution

At this point it is important to note that the computation of the update rule is not affected at all by this definition of the data structures. The only difference is the use of the value of the structures pointed by `Prev` and `Next` instead of using an explicit indirection of the array.
Moreover, this fact has an interesting collateral effect. Using this data structure and the related methods it is possible to make the neighbourhood relationship a non symmetric relationship; the reason is that cell a can take cell b as a neighbour while cell b keeps its original neighbourhood, thus allowing one-way information flow. As an example, consider the system described by Figure 1. In this case, a cell of the upper Cellular Automata will have as a neighbour one of the cells of the lower Cellular Automata. However, in the lower Cellular Automata we still want that cell to keep its original neighbours to obtain the evolution of the lower subsystem.

Dynamic Neighbours in Two Dimensions

The data structures presented for the one-dimensional Cellular Automata can be easily extended to be used in the two-dimensional case. Basically, the `Cell` structure needs to be extended so that it has references to the eight possible cells of its neighbourhood.

This can be done either by defining eight pointers using appropriate names, or by using an array of eight pointers as it is shown in the next piece of code

```
struct Cell2D {
  CellState StateInfo;
  Cell2D * Neighbours[8];
};
```

where `Cell2D * Neighbours[8]` represents an array of eight pointers to `Cell2D` structures. The `Reset_Neighbours()` method also needs to be extended accordingly.
The methods that were defined for neighbour assignment, `Set_Neighbour_Prev()` and `Set_Neighbour_Next()`, are now unified in one single function, which receives as a parameter the index of the neighbour to assign.

```
Set_Neighbour(int i,int n,Cell *_c)
{
  CA_Grid[i].Neighbours[n] = _c;
}
```

Thus, the step from one to two dimensions is rather straightforward and, as in the one-dimensional case, it does not affect the calculus of the evolution of the Cellular Automata. Based on this implementation of the Cellular Automata, arbitrary neighbourhood configurations can be defined in order to build a simulation scenario.

INTERACTION OF TWO SANDPILES

In order to illustrate the proposed methodology, the complete procedure is described for a particular Cellular Automata model; the Sandpile Model (Bak et al. 1987). Briefly, a sandpile model is a Cellular Automata on a regular square grid in which the value of the state variable represents the height of a sand pile over the grid.
The update rule of the sandpile model states that whenever the difference in height between two neighbour cells is higher than a threshold, an *avalanche* happens. This means that the upper cell reduces its height a fixed amount z, and the lower cell increases its height the same amount. Thus, in this case, the interaction between tow cells is set in terms of material exchange when an avalanche takes place.
The interaction between two sandpiles will be considered as a particular case of this situation; when a cell in the frontier of one sandpile receives material, then this material is considered to leave the system, and the cell does not increase its height. When this happens, if another sandpile is placed below the cell, then it receives the material that has been thrown by the upper sandpile.

Modelling of two interacting sandpiles

The proposed modelling methodology is applied to the case of two interacting sandpiles. Next, the different steps that have been defined are described when they are applied to this particular problem.

Emitting cells
The first step of the methodology is to identify the cells that will potentially interact with other automata. According to the description of the problem of two interacting sandpiles, the *emiting cells* in this case are the so-called *frontier cells* of the automata, which are placed at the edges of the grid (see Figure 3). In general, in a n-dimensional grid of sizes $N_1 \times N_2 \times \cdots \times N_n$, a cell is a frontier cell if and only if the $i-th$ coordinate is 1 or N_i.

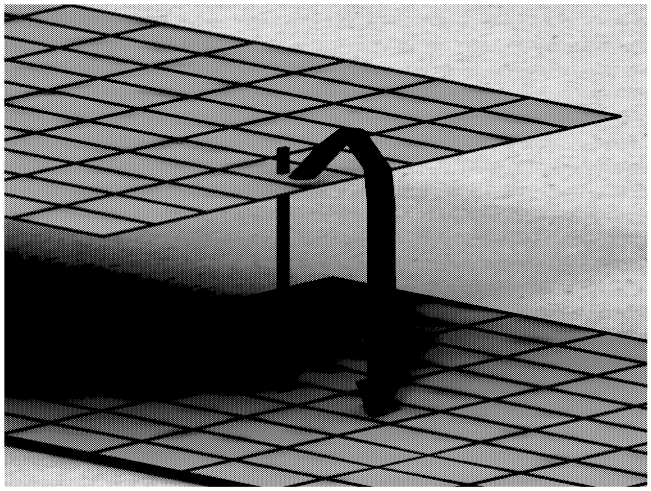


Figure 3: A frontier cell is an *emitting cell* in the problem of two interacting sandpiles.

Note that a frontier cell is an special cell in the sense that it will have several neighbours undefined, as it does not have adjacent cells in some directions. This can be implemented by initializing the corresponding pointers to `NULL` in the data structures defined previously in this paper.

Interaction modelling
The second step in the methodology is to define the necessary interaction models. However, in this case no particular interaction model is necessary, as the same kind of interaction is considered between cells and between the two automata. Thus, by modifying the neighbourhood of the frontier cells of the upper automata the interaction will be automatically simulated when applying the local rule to the modified cells.

Overlapping cells
The third step is the definition of a rule that decides when an *emitting cell* has to be connected to another cell during the simulation. In this case, the decision has to be made according to the relative position of the *emitting cell* in the upper sandpile respect to the lower sandpile.
In order to set the rule, let's first define this idea more precisely. Let us consider the upper sandpile S_u and the lower sandpile S_l. It is assumed that the vertical position of sandpile S_u is higher than the vertical position of sandpile S_l. We say that a cell $(c^u) \in S_u$ overlaps cell $(c^l) \in S_l$ if the centre of cell (c^u) is inside cell (c^l). Now we can define the rule.
Let's consider a cell c^u in the upper sandpile. If it overlaps a cell c^l of the lower sandpile, then c^u is connected to the lower sandpile. In order to do this, all the missing neighbours of c^u are made to point to the corresponding neighbours in c^l.

Neighbourhood update
In order to set up the neighbourhoods of the upper frontier cells, a procedure will be defined. This procedure can be called once, to configure the scenario at the beginning of the simulation, but also can be called after every simulation step, if the relative position of both sandpiles change along time.
Next, the procedure is described in pseudocode, using the data structures that were defined previously in this paper. The next additional notation will be used:

`Sandpile` A structure that contains the data of a complete sandpile, including its position in the simulation scenario.

`Cell * Is_Overlaping(Cell *c,Sandpile *s)` A procedure that checks if cell `c` overlaps any cell of the sandpile `s`. If that is the case, returns a pointer to the overlapped cell. It returns `NULL` otherwise.

`Fr_U[]` An array that contains a list of pointers or references to the frontier of the upper cellular automata.

`Sl` An structure of type `Sandpile` with the information regarding the lower sandpile.

Using the previous data structures, the following procedure is applied to very *emitting cell* of the upper automata, prior to the beginning of the simulation and after every simulation step:

Algorithm 1
Input : `Sandpile Sl, Cell Fr_U[ ]`
Variables : `Cell cl`
for `c` ∈ `Fr_U[ ]`
  `cl` ← `Is_Overlapping(c, Sl)`
  if `cl` ≠ `NULL`
    Assign missing neighbours in cell `c` using the neighbours in cell `cl`.
  else
    Reset neighbours in cell `cl`.
  fi
end

This procedure assumes that the overlapped cell `cl` is not a frontier cell of the lower sandpile, `Sl`. The reason is that frontier cells throw away the material they receive and, thus, they are discarded.

RESULTS

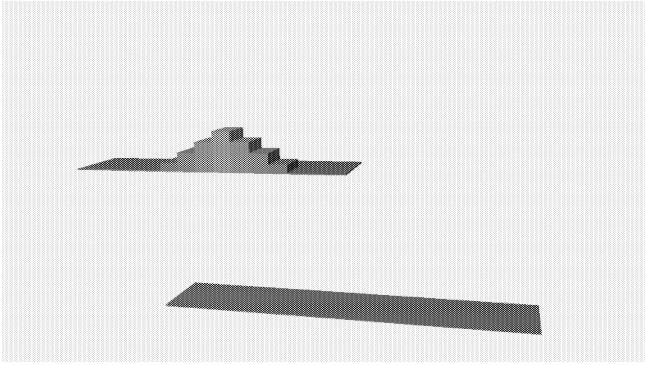
The example above has been used to simulate two different situations. A simple system composed by a pair of one-dimensional sandpiles, and a more general simulation scenario, which reproduces the load of granular material by a small loader machine.

A one-dimensional example

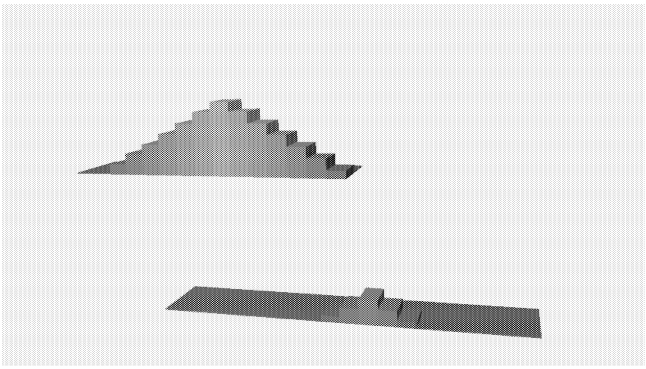
In order to illustrate the application of the methodology, first a simple one-dimensional example is presented. Using the methodology proposed, two sandpiles have been set up at different heights. The upper one has one of its edges over the centre of the lower one. During the simulation, a fixed amount of material has been added to the upper system every step. Figure shows a sequence of snapshots of this simulation.
In this case, the system was static; there was no movement of the sandpiles during the simulation. Thus, the Cellular Automata configuration was defined during the set up stage of the simulation and no update of the cells' neighbourhood was necessary after every simulation step.

Loader simulation

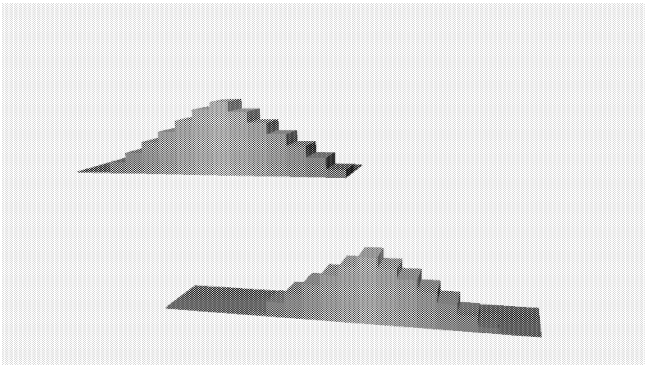
The modelling methodology presented in this paper has been applied to a more complex simulation application. It has been used for the dynamic model in a virtual reality application that simulates a small loader.
In this case, one sandpile is placed on the ground to simulate a heap of granular material and several additional sandpiles are placed over the loader bucket or any other object that needs to accumulate material (see Figure 5). The frontier cells of any moving sandpile are checked after every simulation step to dynamically reconfigure their neighbourhood.



(a)



(b)



(c)

Figure 4: A sequence of the simulation of two sandpiles that exchange material. From bottom to top: the inital state of the simulation (a), the moment when the material reaches the edge of the upper system (b) and the final state (c).

CONCLUSION

This paper has presented a modelling methodology for the simulation of several interacting Cellular Automata. By means of the reconfiguration of the cells' neighbourhood it is possible to define arbitrary connections between the different automata involved.
The methodology has been described formally, and implementation details have also been presented. In addition, it has been developed in more detail for the sandpile model, a particular type of Cellular Automata. Moreover, the applicability of the methodology has been illustrated by means of two applications, which show that it can be used even in complex simulation environments.

The procedures that have been described, based in the modification of cells' neighbourhood, provide with an automated method to intercommunicate two or more automata so that they can share information during a simulation.



Figure 5: A Cellular Automata fixed to an excavator bucket (upper figure). When the bucket is over another Cellular Automata it can load granular material into it.

REFERENCES

Alexandridis A.; Vakalis D.; Siettos C.; and Bafas G., 2008. *A cellular automata model for forest fire spread prediction: The case of the wildfire that swept through Spetses Island in 1990. Applied Mathematics and Computation*, 204, no. 1, 191–201.

Aponte A. and Moreno J., 2006. *Cellular Automata and Its Application to the Modeling of Vehicular Traffic in the City of Caracas.* In S.E. Yacoubi; B. Chopard; and S. Bandini (Eds.), *ACRI*. Springer, *Lecture Notes in Computer Science*, vol. 4173. ISBN 3-540-40929-7, 502–511.

Bak P.; Tang C.; and Wiesenfeld K., 1987. *Self-organized criticality: An explanation of the 1/f noise. Physical Review Letters*, 59, no. 4, 381–384.

Chopard B. and Droz M., 2005. *Cellular Automata Modeling of Physical Systems.* Cambridge University Press.

Pelechano N. and Malkawi A., 2008. *Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. Automation in Construction*, 17, no. 4, 377–385.

Ribba B.; Alarcón T.; Marron K.; Maini P.; and Agur Z., 2006. *The Use of Hybrid Cellular Automaton Models for Improving Cancer Therapy.* In *Proceedings of ACRi'06*. 444–453.

Vicari A.; Alexis H.; Negro C.D.; Coltelli M.; Marsella M.; and Proietti C., 2007. *Modeling of the 2001 lava flow at Etna volcano by a Cellular Automata approach. Environmental Modelling & Software*, 22, no. 10, 1465–1471.

Zhou S., 2009. *Study on the Evacuation Simulation Based on Cellular Automata.* In *ITCS '09: Proceedings of the 2009 International Conference on Information Technology and Computer Science*. IEEE Computer Society, Washington, DC, USA. ISBN 978-0-7695-3688-0-02, 481–484.

BIOGRAPHY

MARTA PLA-CASTELLS studied mathematics at University of Valencia and had her PhD in Computer Science in 2009. She is lecturer at Computer Science department in Universty of Castellon and researcher at IRTIC. Her main research interests are Virtual Reality applications for training and interactive simulation.
IGNACIO GARCIA-FERNANDEZ studied mathematics at University of Valencia. He had his PhD in Computer Science in 2009 and holds a position of lecturer at Informatics Departament of University of Valencia. He is researcher at IRTIC and his main interest is physics based interactive simulation for training.
RAFAEL J. MARTINEZ studied informatics and had his PhD in Computer Architecture in 1999. Since 2000 he is the head of simulation group LSYM at IRTIC, and has leaded several research projects within the field of simulation for training.

MODULAR TECHNOLOGY: A METHODOLOGY FOR THE CREATION OF VIRTUAL ENVIRONMENTS IN TERRAIN DRIVING SIMULATORS

Carlota Tovar
Ginés Jesús Jimena
Jose María Cabanellas
CITEF Research Centre on Railway Technologies
Universidad Politécnica de Madrid.
José Gutierrez Abascal 2. 28006 Madrid,
Spain
E-mail: ctovar@etsii.upm.es;citef-gjimena@etsii.upm.es ;jmcabanellas@etsii.upm.es

KEYWORDS
training driving simulators; geometric modelling; shaders; modularity; virtual environments.

ABSTRACT

This paper presents the latest research and developments in Modular Technology, a constructive methodology designed to create the environments of the virtual driving simulators of CITEF. Modular Technology discretizes the scene into a finite number of modules or portions of the environment, which after being instantiated and subjected to a series of geometric transformations using shaders, are meticulously assembled to reproduce the virtual scene. This tiling system has a set of particularities that make it different from the rest of tiling systems used up to now. These particularities are designed to get the maximum benefit of these type of virtual representations and can be summarized in four points: the shape of the module; the way in which the family is defined; the positioning system of these modules; and the deformation of these modules using shaders. In this way a substantial reduction in the number of geographical entities in the environment can be attained and an increase in the diversity and flexibility of the environment.
There are many advantages to be had from this form of generation: savings in scene size, loading times and resource requirements, greater flexibility and clarity of the scene graph and a more substantial upgrade capacity of the environment.

INTRODUCTION

The use of virtual driving simulators as a learning and training tool is a strongly established procedure that is becoming more and more widespread with a whole range of possibilities going from high and medium to low-cost. All this is a result of the rapid progress in hardware which has led to like increases in the demands for realism in virtual display scenes.
When creating virtual environments for training driving simulators, it is indispensable to take into account the final aim of these virtual representations: to train a driver. Sometimes drivers are not used to work with computers, so that, they need intuitive, handy and useful tools. The functionality of the environment must be their main aim and the aesthetic appearance has a secondary relevance. Trying to fulfill these requirements, the different entities that develop training driving simulators create their own applications.
CITEF (Research Centre on Railway Technologies) is one of these entities and it has developed Modular Technology, a constructive methodology designed to create the environments of terrain driving simulators. By constructive methodology of a virtual environment we mean a set of procedures that allow interpreting, processing, analysing, modelling, optimising and virtually displaying certain initial information.
In the development of its tools, Modular Technology has considered the change of priorities in scene optimization. The high processing capacity of present-day GPUs means that the CPU can be freed of tasks, which is vitally important in terrain driving simulation. The geometric load to be rendered can be increased making the CPU-GPU communication bus the new bottle-neck. At the same time, the programming possibilities of current GPUs (*shaders*) (Wolfgang,E 2004) offer great flexibility to build stunning environments.
In response to this change of priorities, a new generation of algorithms has begun to appear to adapt the level of detail (lods) whose work primitive has ceased to be the triangle (Gobbetti et al, 2006). Blocks of triangles (batches), are used instead. These are optimised in pre-load time to ensure optimal spatial organisation (stripping) thereby speeding up the CPU-GPU transfer.
Following this latest line of research, Modular Technology has converted the module into its base primitive and has created the environments using thanks to the use of instantiation and shaders.
Instantiation is an useful technique to save memory (Schultz, Schumann, 2000). For scenes with many similar objects it is desirable to save just the properties that are different between those objects and not to save the full geometric representation of each object. The *Master* is the object which holds full geometric information and the *Instance*, is a transformed variant of the *Master* object (Sutherland, 1963). In this way, as the *Instance* is a pointer to the *Master*, the geometry is saved just once and it can be reuse (instantiated) as many times as desired.
A tile-based (or instantiation-based) methodology, is characterised by its seeking the use of repetitive patterns as the basis of its designs to generate an environment by assembling and repeating a finite number of modules generated in pre-load time. This methodology usually goes

hand-in-hand with the use of a system of discrete levels of detail (Luebke 2003). The most usual geometric shapes for generating these tiles are the rectangle, the circle, the hexagon and the diamond shape. The purpose of the two latter is to provide a greater feeling of depth. These tiles are placed adjacently to one another as the following figures show, completely filling the 2D space.

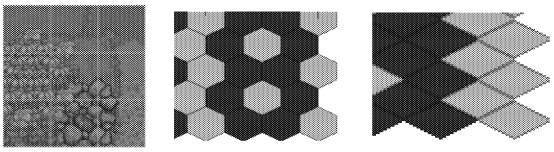


Figure 1. Examples of tiles

This methodology is used frequently in video games. Two of the best-known 3D games using this methodology are *Simcity*, and *Neverwinter Nights*. Their main feature is that they define the scene by using tiles that form part of what is known as a tileset. A tileset is nothing more than a set of tiles that follow a similar theme. For example, the City tileset can include tiles of pubs, inns…and other elements to be found in a city. The main aim of this methodology is to cut memory consumption and to reduce modelling tasks. The level of detail of these scenes, however, plays a secondary role in fulfilling the objectives of the game, which means that realism is not a decisive factor when generating the scenes.

Modular Technology is tile-based methodology that searches for repetitive patterns in the environment, geometric as well as functional. Using these patterns (*Masters*) discretizes the scene into a finite number of modules or portions of the environment, which after being instantiated and subjected to a series of geometric transformations using shaders, are meticulously assembled to reproduce the virtual scene.

MODELLING LARGE VIRTUAL ENVIRONMENTS IN TERRAIN DRIVING SIMULATORS

When it comes to choosing the most ideal methodology for generating large virtual environments in terrain driving simulators, it is imperative to choose the data structures that will respond to the constraints imposed by the environment to be displayed and minimise CPU consumption and maximise the resources offered by current GPUs.

In the case of overland driving simulations, the kind of paths to be driven over will be a decisive factor in the choice of the most ideal types of data structures. Depending on this, two environment types can be differentiated:

- Undetermined route environments.

This is the case with combat simulations in military training. Since the paths do not impose any geometric constraint, the data structures are more flexible (regular or irregular) and the variety of algorithms for creating multiresolution models is very wide. In this field, the latest trends, as stated in the introduction, point to lightening the CPU load and speeding up CPU-GPU transfer using the new work component: the batch (Livny et al, 2007; Pajarola and Gobbetti, 2007). The batches or tiles are constructed in the pre-load time optimising to the maximum the organisation of the triangles they contain (strips).Thanks to the batch the selection metrics are no longer evaluated at a vertex level, but are performed at a batch level, enormously reducing the load of the CPU. Although this means the number of polygons to be displayed is less optimised, the high processing capacity of current GPUs means this is not a problem.

- Pre-determined route environments.

This is the case with the railway and urban driving simulation dealt with in this paper. The enormous realism required for the path surroundings can only be achieved through the geometric insertion of the paths into the landscape. This has large repercussions on the choice of data structures to be used. Inserting this geometry involves a readjustment and a retriangulation of the terrain mesh in the geometry surroundings (Chew, 1987), so that it can be adapted to the highest path resolution. This supposes a notable increase in the number of calculations to be performed, but also an ensuing increase in the main drawback posed by the continuous levels of detail adaptation algorithms (continuous lods) and the heavy use of the CPU. Discrete levels of detail (Govil and Mourant, 2005; Papelis et al, 2003; Suresh and Mourant, 2005) therefore, become the preferred alternative, their main drawbacks being:

- The high memory consumption required to store the geometry associated with the different levels of detail during the pre-load time.
- Little constructive flexibility.
- Failure to make full use of the functionalities offered by present-day GPUs.

Modular Technology has arisen to meet the latest needs designing a new tile-based system, with some particularities that let us to solve all these drawbacks as it will be explained in next sections.

PRINCIPLES OF MODULAR TECHNOLOGY

Modular Technology uses route tracking as the starting point of its work from which it generates the surrounding environment using the assembly and repetition of a finite number of modules. It thus exploits one of the basic tools for resource optimisation: instantiation.

Calculating the number and optimum geometric design of the modules as well as the variety of their families will be decisive in obtaining the realism and refresh speed required in these simulations.

Knowing the driving paths allows setting the distance at which the different elements will be observed, thereby enabling the optimum parameters of the levels of detail to be defined in pre-load time: the cut-off distances and polygon load for each level of detail.

The outcome is a scene with all the realism required by driving simulations with the desired refresh speeds, with a comfortable communication interface with the simulation module, graphic engine and graphic designers (Figure 2).

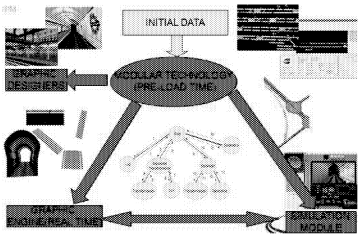


Figure 2. Areas involved in the creation of a virtual environment in CITEF

Modular Technology analyses the initial data, solving incoherences and mistakes. Once this information is ready it distributes it, sending:

- to the simulation module all data that the vehicle needs from the environment to function properly. Data related to the geometry of the paths, sections, signals, magnets, etc.
- to the graphic engine the geometry that it has generated automatically and a text file in which this geometry is positioned and structured in a scene graph. This scene graph contains information about the levels of details, switches, shaders..
- to the graphic designers the information of the geometry that can not be generated automatically because of the high level of detail required.

All this is done automatically and with the possibility of easily inserting any environmental restructuring, something that is essential in this kind of virtual display. Moreover, the main problems associated up to now with the discrete level of detail adaptation algorithms (discrete lods) are eliminated:

- Thanks to instantiation and the shaders a straight module of each family need only be stored during the pre-load time, which means that memory ceases to be a problem and load times are minimised. This last point is very important for the user to be able to launch different scenes easily and efficiently in a training session.
- The scene is synthesised from a set of repetitive parameters, which means modelling is simplified.
- The modules are subjected to geometric transformations using shaders during execution time which lets the realism of these scenes be enhanced. These transformations will be defined during the pre-load time by means of a set of parameters integrated next to the scene graph.
- The modularity of the environment endows it with high constructive flexibility with a high range of virtual scenes being able to be generated with a minimum of constructive parameters.

MODULAR TECHNOLOGY PARTICULARITIES

The particularities of this tiling system will be explained throughout this section. These particularities, which make Modular technology different from all other tiling systems utilised up to now, are related to:

1. The shape of the module.
2. The way in which the family is defined.
3. The positioning system of these modules.
4. The deformation of these modules in execution time using shaders.

Particularity 1: the shape of the module

The shape of the module usually used by the different tiling systems does not take into account the shape of the driving paths. Squares, circles and rhombus are the shapes more frequently used by these tiling systems.

In Modular Technology, the module originates from a basic design made up of a straight longitudinal portion of the environment. This basic design is curved into a set of degrees to represent the shape of the driving paths.

This group of modules of different curvature forms what is called a basic set of modular components or modules (Figure 3.). In this way the basic set is characterised by the existence of a finite number of modules of fixed curvature and discrete curvature values. If the straight module is curved in execution time using shaders, this basic set of modules will be compound by infinite elements.

The module comprises three basic components: a longitudinal path and a set of transversal profiles with their corresponding textures (Figure 3.).

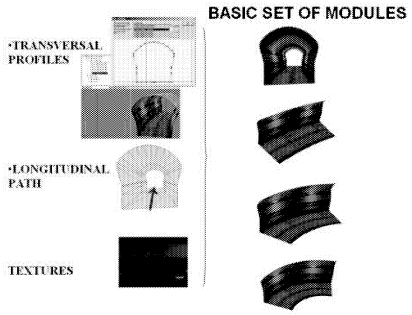


Figure 3. Components of a module and basic set of modules

Particularity 2: The definition of the family

Frequently, the design of these families is very rigid, so that, the environments created with the repetition of these families tend to be very unrealistic. These families include too much detail and the user can perceived the repetition. Next figure shows an example of one of these families.

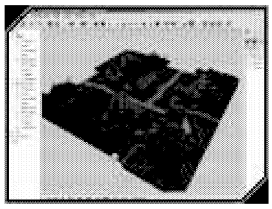


Figure 4. Example of family used by a tiling system

Modular Technology takes into account the driving paths to define these families. So, the concept of a family of modules now appears as the base set of modules that shares the same definition of transversal profiles and textures (Figure 5.). These families will verify relationships of compatibility, which will be what ensures there are no incoherencies or discontinuities after the modules have been correctly positioned.

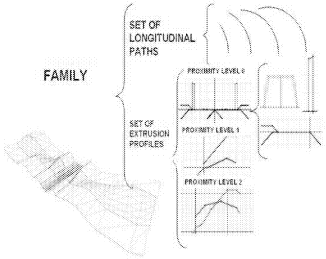


Figure 5. Components of a family

Modular Technology takes advantage of the priori knowledge of the paths to generate discrete pseudo-variant levels of detail with the view point. This is achieved by discretizing the environment transversally to the path in different levels of proximity (Figure 6).

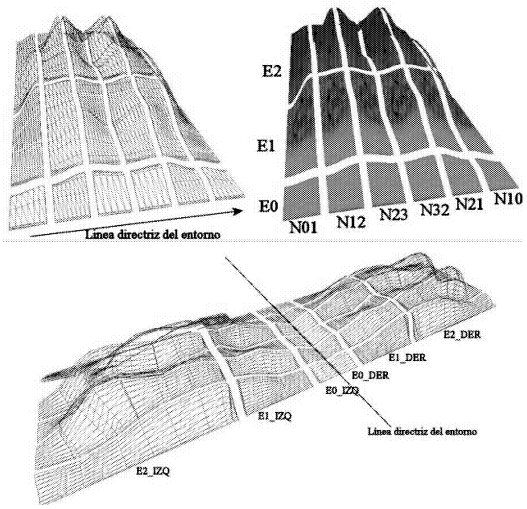


Figure 6. Example of a terrain family

In order to ensure these families are correctly positioned, the guidelines are divided into intervals. Each interval is associated with a different family type.

Particularity 3: The positioning system

Up to now, the different tiling systems positioned the tiles forming a mesh. The distance to the driving paths was not taken into account, so that, the level of detail could not be calculated with precision. Modular Tecnology solves this problem positioning the tiles along the tracks, so that, the level of detail of each module can be optimized evaluating this distance.
The modular positioning algorithm has to solve the following issues (Tovar, 2003):

- The optimum number of modules, responsible for a higher or lower graphic load and a better or worse scene graph path optimisation. In order to decide the optimal number of modules to make up the environment, it is firstly necessary to define the families comprising it and secondly to define the basic length, LB, for each of these families. Although it is obvious that instantiating involves shorter scene loading times, it is advisable to proceed with caution when deciding on the number of elements and how to define the elements that are instantiated, since an excessive number lead to an opposite effect:
    - a large number of changes in state results in a GPU overload and a drop in frame rate.
    - a large number of elements involves more calculations in the scene graph in order to determine how many elements need to be displayed.

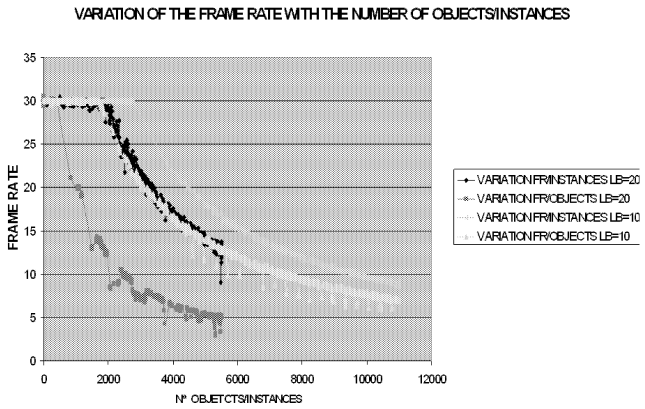


Figure 7. Evaluation of the the number of elements on the frame rate

In the above example, the frame rate has been limited to 30. As can be seen, there is a threshold in the number of instances to be used and if this is not exceeded the performance of the graphic engine is stable and supplies a maximum frame rate regardless of the basic length used. When creating each scene, this threshold must be calculated, and consequently, the basic length.
This threshold varies according to the number of changes of state linked to each module, which, in turn, depends on the number of drawables and textures defining the threshold (Figure 8).

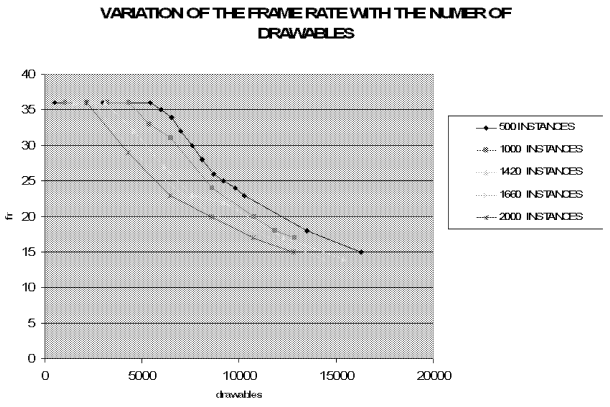


Figure 8. Evaluation of the number of drawables on the frame rate

Therefore, the procedure to be followed to calculate the basic length, LB, will consist in firstly determining the families that will set the environment and the number of drawables that is to comprise the module. Taking this information and using the graph depicted in Figure 7, after an iterative process, the appropriate number of instances and therefore the basic length is determined.

- The type of module to be placed at each point of the scene: if the module curvature is wrong, spatial and tangential discontinuities are produced.
- Scaling each module. The modules can be subjected to an X, Y and/or Z scale. The maximum scale admitted will vary depending on the length of the module (basic length, LB). Exceeding this scale leads to an excessive deformation of the overlapping and texture. On the other

hand, excessive scaling leads to a noticeable variation in module curvature that results in defective module coupling. These scaling tolerances can only be overcome by shaders to correct these distortions. In these circumstances Modular Technology will calculate these deformation parameters through the graphic engine, which will be responsible for managing the execution time.

- The positioning coordinates of each module. The geometric positioning site for modules is called a polygon site and consists of a set of points obtained by interpolation on the said biarc. The modules are positioned at the mid-point of their left end and orient their chord in line with the polygon segment on which they are positioned, taking in the angle of arc included between both ends of its chord. Thus the modules circumscribe the guideline as can be seen in the following Figure 9. Figure 10. shows how a set of railway platform modules is put in place in accordance with the positioning algorithm. Each module appears on the left in a different colour so that the correct coupling between them can be more clearly seen since in the right-hand image the perfection of the coupling makes it impossible to discern this joint.

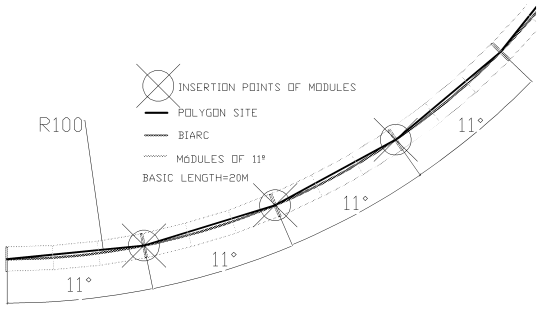


Figure 9. Positioning algorithm



Figure 10. Railway platform and rail modules positioned according to the positioning algorithm

Determining the module to be inserted at each point of the polygon from those that the geometric discretization has deemed to be a basic set of modules, will be determined by the biarc angle of embrace between the two points of the polygon:

$$\alpha = \frac{L_{arc}}{R}$$

where L_{arc} is the length of the mid-line of the module and R the radius of the biarc.

The longitudinal scaling to which the module will be subjected is:

$$e = \frac{L_{chord}}{LB}$$

where LB is the basic length taken for constructing the module family and L_{chord} the polygon segment length or module chord length.

Particularity 4 : The use of shaders

Up to now, the different tiling systems did not deform the geometry of the tile in execution time to create more realistic environments. The use of shaders lets each modular family be formed during the pre-load time by a single straight module. In execution time the shader will take charge of deforming this module until it attains the desired geometry. Thus, modelling effort is reduced as well as initial load times and the modular positioning algorithm is made more versatile since the shader is capable of deforming the start and end sections of each module so that they fit perfectly without any spatial or tangential discontinuities. As a result, each family goes on to be formed in execution time by infinite modular components.

Modular Technology starts out from a straight module of basic length LB. This length is measured in the direction of the axis i_1, as Figure 11. shows.

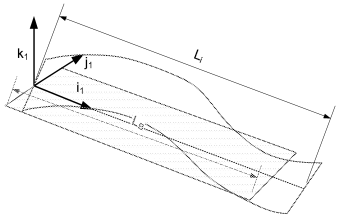


Figure 11. Straight module deformed by the shader

The axis i_1 in turn, coincides with the direction marked by the guideline that positions it, while its direction is marked by the increasing pks.

Using the composite transformation shader, Modular Technology seeks to subject the module to a deformation that will guarantee perfect coupling with the adjacent modules after being positioned by the modular positioning algorithms.

To do this, firstly a dimensionless parameter s must be defined as the relationship between the x_0 coordinate and the basic length LB. The parameter s will vary between 0 and 1.

$$s = \frac{x_0}{LB}$$
$$0 \le s \le 1$$

What is sought is to define a function that will ensure zero displacement at the ends of the module and whose tangent angles coincide with those required (those defined by modular positioning algorithm).

In this way, the function calculating the displacement at y to which the module points have to be subjected, should verify:

$$f_{\Delta y}(s) \equiv \begin{cases} f_{\Delta y}(0) = 0 \\ f_{\Delta y}(1) = 0 \\ \dfrac{df_{\Delta y}}{dx_R}(s=0) = \tan\gamma_{ini} \\ \dfrac{df_{\Delta y}}{dx_R}(s=1) = \tan\gamma_{fin} \end{cases} \quad (1)$$

In the above equation x_R represents the actual module coordinate after scaling. The relation with x_0, the module path coordinate before deformation and with s is:

$$\begin{aligned} x_R &= x_0 \cdot E_x \\ \frac{dx_R}{dx_0} &= E_x \\ \frac{dx_0}{ds} &= L_B \\ \frac{dx_R}{ds} &= L_B \cdot E_x = L_i \end{aligned} \tag{2}$$

The group of equations (1) bearing in mind (2) may be re-written as:

$$f_{\Delta y}(s) \equiv \begin{cases} f_{\Delta y}(0) = 0 \\ f_{\Delta y}(1) = 0 \\ \frac{df_{\Delta y}}{dx_R}(s=0) = \frac{df_{\Delta y}}{dx_R}\frac{ds}{ds}(s=0) = \frac{df_{\Delta y}}{ds}\frac{ds}{dx_R}(s=0) = \tan\gamma_{ini} \\ \frac{df_{\Delta y}}{ds}(s=0) = L_i \cdot \tan\gamma_{ini} \\ \frac{df_{\Delta y}}{ds}(s=1) = L_i \cdot \tan\gamma_{fin} \end{cases} \tag{3}$$

The functions f_1 and f_2 have the following properties:

$$\begin{aligned} f_1(s) = s\cdot(1-s)^2 \quad & f_2(s) = s^2\cdot(1-s) \\ f_1(0) = 0 \quad & f_2(0) = 0 \\ f_1(1) = 0 \quad & f_2(1) = 0 \\ \frac{df_1}{ds}(s) = (1-s)(1-3s) \quad & \frac{df_2}{ds}(s) = s\cdot(2-3s) \\ \frac{df_1}{ds}(0) = 1 \quad & \frac{df_2}{ds}(0) = 0 \\ \frac{df_1}{ds}(1) = 0 \quad & \frac{df_2}{ds}(0) = -1 \end{aligned} \tag{4}$$

By correctly putting together the requirements of (3) and the results of (4) a suitable function for $\mathbf{f_{\Delta y}}$ is obtained:

$$\begin{aligned} f_{\Delta y}(s) &= f_1(s)\cdot L_i \cdot \tan\gamma_{ini} - f_2(s)\cdot L_i \cdot \tan\gamma_{fin} \\ f_{\Delta y}(0) &= 0 \\ f_{\Delta y}(1) &= 0 \\ \frac{df_{\Delta y}}{ds}(0) &= L_i \cdot \tan\gamma_{ini} \\ \frac{df_{\Delta y}}{ds}(1) &= L_i \cdot \tan\gamma_{fin} \end{aligned}$$

For the increases in z, $\mathbf{f_{\Delta z}}$ the conditions for (3) will need to be appropriately adapted:

$$f_{\Delta z}(s) \equiv \begin{cases} f_{\Delta z}(0) = 0 \\ f_{\Delta z}(1) = 0 \\ \frac{df_{\Delta z}}{ds}(s=0) = -\frac{L_i \tan\beta_{ini}}{\cos\gamma_{ini}} \\ \frac{df_{\Delta z}}{ds}(s=1) = -\frac{L_i \tan\beta_{fin}}{\cos\gamma_{fin}} \end{cases}$$

Where the sign is negative it indicates that rotation β is in the opposite direction to the derivative. The displacement function is finally given as:

$$f_{\Delta z}(s) = \frac{-f_1(s)\cdot L_i \tan\beta_{ini} + f_2(s)\cdot L_i \tan\beta_{fin}}{\cos(\gamma(s))}$$

There are no deformations through deformation in the direction of x, except the scaling.
The rotations α(s), β(s) and γ(s) that are coherent with the displacement functions described are obtained by deriving. A simple reflection shows us that the derivative in respect of x_R of the displacement function at z corrected by the cosγ also gives the angle β(s). The slope angle is a linear extrapolation of the initial and final slopes.

$$f_{\Delta y}(s) = f_1(s)\cdot L_i \tan\gamma_{ini} - f_2(s)\cdot L_i \tan\gamma_{fin}$$

$$\frac{df_{\Delta y}}{ds}(s) = (1-s)(1-3s)L_i \tan\gamma_{ini} - s(2-3s)L_i \tan\gamma_{fin}$$

$$\frac{df_{\Delta y}}{dx_R} = \tan(\gamma(s)) = \frac{df_{\Delta y}}{ds}\frac{ds}{dx_R} = (1-s)(1-3s)\tan\gamma_{ini} - s(2-3s)\tan\gamma_{fin} \Rightarrow$$

$$\gamma(s) = \arctan((1-s)(1-3s)\tan\gamma_{ini} - s(2-3s)\tan\gamma_{fin})$$

$$\gamma(0) = \gamma_{ini} \quad \gamma(1) = \gamma_{fin}$$

For the angle β(s):

$$\frac{df_{\Delta z}}{dx_R} = -\frac{\tan(\beta(s))}{\cos(\gamma(s))} = -\frac{df_{\Delta z}}{ds}\frac{ds}{dx_R} = \frac{(1-s)(1-3s)\tan\beta_{ini} - s(2-3s)\tan\beta_{fin}}{\cos(\gamma(s))} \Rightarrow$$

$$\beta(s) = \arctan((1-s)(1-3s)\tan\beta_{ini} - s(2-3s)\tan\beta_{fin})$$

$$\beta(0) = \beta_{ini} \quad \beta(1) = \beta_{fin}$$

For the angle α(s) a linear interpolation is used:

$$\alpha(s) = (1-s)\cdot\alpha_{ini} + s\cdot\alpha_{fin}$$

The results of this deformation with shaders are shown in next Figure.

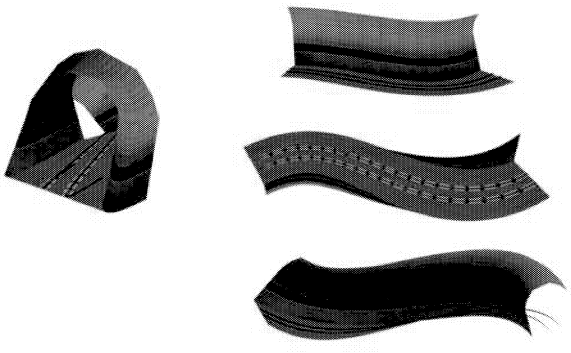


Figure 12. Straight module deformed by the shader

APPLICATIONS

All virtual environment projects developed in CITEF have a modular component to a greater or lesser degree, which helps develop automation.
The development of the simulator for Metro of Madrid is an example of the advantages of this tiling system. Per example, the 80% of Line 7 has been created modularly. Stations and deposits are the only elements created ad hoc because of their high number of specifics details. This implies a great reduction in modelling tasks: the 23 km of tunnel has been reduced to the creation of 5 families of tunnel, i.e, 5 modules of 20 meters. So that, the resources envolved in the creation of this geometry are drastically reduced.
Next table shows the savings in memory and loading times thanks to the use of Modular Technology in the representation of the tunnels of this line.

Table 1. Loading times and memory used by tunnels in Line 7 of Metro of Madrid

USE OF MODULAR TECHONOLOGY	LOAD TIME (seconds)	MEMORY
YES	0.059	173KB
NO	31	133MB

These loading times and memory savings are crucial during a training session. Drivers need to launch several exercises during these sessions, so they can not waste time loading them. On the other hand, these trainings imply to visualize huge environments, per example, the Madrid Metropolitan Network, so the memory used has to be minimize. Moreover, drivers need to modify and create new environments during their training sessions. Thanks to the use of Modular Technology this tunnel families can be reused without an extra modelling effort.
Figure 13 shows different scenes that have been modularly generated. These scenes are taken from the virtual environments of the driving simulators designed by CITEF for companies such as Metro de Santiago de Chile, Metro de Madrid, CAF, Alstom and Invensys.

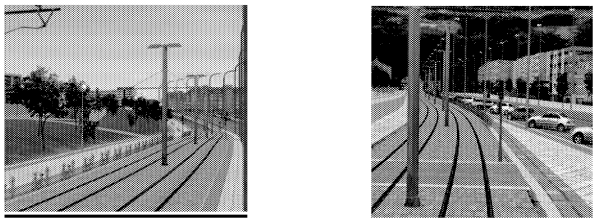


Figure 13. Examples of virtual environments created with the Modular Technology

CONCLUSIONS

From what we have seen from the applications used, we are convinced that modular technology has advantages when it comes to generating environments. Ten years of successful results in the generation of virtual environments for renowned driving simulators have proved it. Applying shaders increases the possibilities of modularity and with ever more powerful hardware the quality of virtual environments will increase spectacularly in the near future.

Future developments will be to apply geometric shaders that create the full modular geometry in execution time so that the scene is inserted parametrically in line with the profiles and paths with only the most singular and least modular geometry being loaded. Real-time generation would therefore be adaptive. The greater the capacity for geometric calculation, the greater the complexity and quality of the environment.

REFERENCES

Chew,L.P.1987."Constrained Delaunay triangulations". In Proceedings of the Third Annual Symposium on Computational Geometry (Waterloo, Ontario, Canada, June 08 - 10, 1987). D. Soule, Ed. SCG '87. ACM Press, New York, NY, 215-222.

Gobbetti, E.; F.Marton ;P. Cignoni. 2006. "C-BDAM - Compressed Batched Dynamic Adaptive Meshes for Terrain Rendering". Computer Graphics Forum, 25(3), (Sept). Proc. Eurographics 2006.

Govil, V. and Ronald R. Mourant. 2005. "A Tile/Scenario Algorithm for Real-Time 3D Environments". Proceedings the 32nd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2005), Los Angeles, California. (August).

Livny,Y.; Z. Kogan; J. El-Sana. 2007. "Seamless Patches for GPU-Based Terrain Rendering". WSCG 2007.

Luebke, D., M. Reddy, J. Cohen, A. Varshney, B. Watson and R. Huebner. "Level of Detail for 3D Graphics". Morgan Kauffman.2003.

Pajarola,R. and E. Gobbetti. 2007. "Survey on Semi-Regular Multiresolution Models for Interactive Terrain Rendering". The Visual Computer, 23(8). 583-605.

Papelis, I.; O. Ahmad; G. Watson. 2003. "Scenario Definition and Control for the National Advanced Driving Simulator". In Proceedings of the Driving Simulation Conference North America, Dearborn, Michigan 2003.

Schultz,R., Schumann,H. 2000. "Efficient Scene Descriptions Using Advance Modelling Techniques in the RenderMan Context". Proceedings of Spring Conference on Computer Graphics 2000.

Suresh, P. and Ronald. R. Mourant. 2005. " A Tile Manager for Deploying Scenarios in Virtual Driving Environments". Proceedings of the Driving Simulation Conference North America, December, 2005.

Sutherland, I.E. 1963."Sketchpad: A man-machine graphical communication system". Proceedings of the Spring Joint Computer Conference 1963.

Tovar, C.; J.M. Cabanellas; J. Félez. 2003. "Procedimientos geométricos para el ajuste de trayectorias ferroviarias en simuladores 3d de diseño modular" Proceedings of ADM-INGEGRAF'2003 (June) Naples, Italy.

Wollfgang, E. 2004 "Programming vertex & pixel shaders". Charles River Media Graphics.

CARLOTA TOVAR received her Mechanical Engineering Master degree from the Madrid Polithecnic University and her Master degree on Geographic Information Systems from the University Pontificia of Salamanca. She has been working for ten years in CITEF, Research Centre on Railway Technologies. Her main activities and research interests are mainly focused on the field of simulation, computer graphics, virtual reality and geographic information systems. Her main contribution is in the field of automatic generation of virtual environments. She has just finished her doctoral thesis on this subject. She has been actively involved in over 20 research and development projects.

SIMULATION OF AN HEV EQUIPPED WITH A VARIABLE DISPLACEMENT ICE

Valerian CROITORESCU*
Andrei MACIAC
Marius BATAUS
Mircea OPREAN
Cristian ANDREESCU

University POLITEHNICA of Bucharest
Automotive Engineering Department
313 Splaiul Independentei st.
6th Sector, PC 060042,
Bucharest, ROMANIA
E-mail: valerian.croitorescu@gmail.com

KEYWORDS
Mechanical Engineering, transportation, dynamic modeling, continuous simulation, system dynamics

ABSTRACT

The automotive manufacturers intend to improve average fuel economy and low harmful emissions from cars. As the automotive industry is progressing faster than ever it needs systems development which can ensure the new regulations demand to reduce climate change by improving fuel economy. Pollution control devices used on cars cannot reduce the car`s CO_2 emissions down to zero, they only can reduce them partially. Less CO_2 emissions and better fuel consumption will be obtained by combining more control devices. Hybrid-electric vehicles (HEVs) combine the benefits of gasoline engines and electric motors and can be configured to obtain different objectives, such as improved fuel economy, increased power, or additional auxiliary power for electronic devices and power tools. The electric motor provides additional power to assist the engine in accelerating, passing, or hill climbing. This allows for a smaller, more efficient engine to be used. In some vehicles, the motor alone provides power for low-speed driving conditions where internal combustion engines are least efficient. The variable displacement of the internal combustion engines by cylinder deactivation combined with an electric motor can build a hi tech HEV that better complies with the demands.
This paper aims to create models and simulate the operation of a parallel hybrid propulsion system equipped with an internal combustion engine, with 6 cylinders when three of the engine cylinders are deactivated trying to study the behavior of the propulsion system and the engine.

INTRODUCTION

New European standards will require auto manufacturers to reduce CO_2 tailpipe emissions down to 130 g/km by 2015. Hybrid vehicles are not the only answer to reduce harmful emissions and fuel consumption. The automotive manufacturers are developing an array of other promising technologies. To meet the requirement of decreasing emissions and fuel consumption they will have to enhance current internal combustion engine (ICE) technologies and develop some new ones. The variable displacement of the internal combustion engines by cylinder deactivation is one of them. Cylinder deactivation is part of a tool kit that the auto industry is using to improve fuel economy and reduce to CO_2 emissions. This technology helps the engine to seamlessly operate on a reduced number of cylinders under light conditions. Using cylinder deactivation is a way to reduce the number of active cylinders under light load conditions, or when the electric motor ensures most of the work.
Cylinder deactivation saves fuel by deactivating cylinders when they are not needed.
Disconnecting cylinders, in other words deactivating some of them at partial load, leads to savings of fuel consumption by 10 to 15%. This technology was introduced a few years ago in large engines (V12 and V8) and has been applied more recently in a V6 engine having a displacement of 3.5 liters. In the future, its application could be extended to 4-cylinder in-line engines, representative of the great majority of European applications, provided that a way can be found to relieve the problems of vibration and roughness resulting from the deactivation.

MODEL AND SIMULATION PARAMETERS

The HEV equipped with a variable displacement ICE will be simulated using the AMESim software tool. Developing the model for simulation consists in linking submodels from various software`s libraries. The submodels are shown as icons. After linking the icons (which represent each component of the vehicle) in sketch mode and choosing the proper submodel in submodel mode, the parameters will be set. Each icon covers a fragment of C code, written using the specific equations for the system.
The submodel for the driver used in this model controls/calculates braking, acceleration and gear shifting. When the vehicle velocity is below a certain threshold, the neutral is engaged to avoid the engine stalling. Acceleration and brake control take into account the anticipation of the speed control.
The submodel for the ICE, which is a V6 engine, computes the torque, the emissions, the fuel consumption, the exhaust gas flow rate and the consumption thermal losses. The ICE is controlled by a special ECU which manages the different regulation modes and controls the torque request and computes the four following variables: load requested by the driver, combustion mode, thermal coefficient combustion release, controlled idle speed. The ECU performs regulations for idle speed and maximum speed. Also the ECU controls the electric motor (the rotary velocity) and the battery (the state of charge), analyzing them in order to minimize the consumption of the battery. The electric motor and the engine are connected to the

manual gearbox; the ECU manages the power requested to the engine and the electric motor; if the battery needs to be regenerated, the engine is used to drive the vehicle, and if the power requested is lower than its optimum power (according to its rotary velocity), the difference is sent to the electric motor to charge the battery. This ECU can be used only with a parallel hybrid vehicle.
The model for the electric motor can be used as an electric motor and a generator, being independent from the technology of the engine.
The battery is an internal resistance model, which characterizes the battery with a voltage source and an internal resistance. The battery output voltage is calculated as follows V=V0-I*R, where V [V] is the output voltage, V0 [V] is the open circuit voltage, R [Ohm] is the equivalent internal resistance, I [A] is the input current. The battery consists of banks in serial and parallel arrangements; each battery bank consists of cells.
The submodel for the 6 ratio gearbox includes the clutch and is controlled by the driver. The gearbox ratio is read in a data table. The primary shaft inertia and the powered axle are taken into account.
The submodel for the vehicle models a simple vehicle load when considering no longitudinal slip between the tire and the ground.

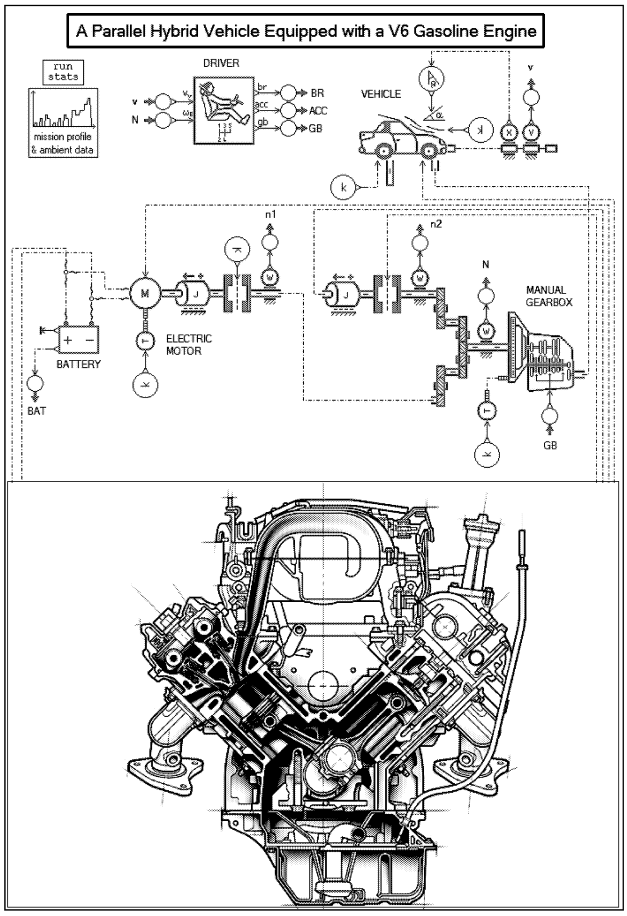


Figure 1: Model of the HEV equipped with a variable displacement ICE

The parameters used in the simulation are borrowed from an existing vehicle and are shown in Table 1.

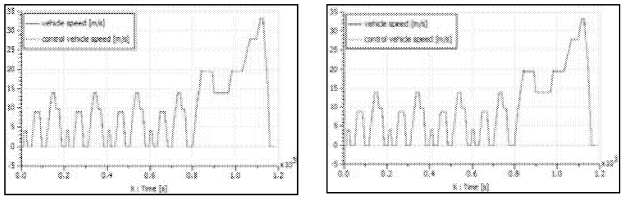
Table 1: The most important parameters used in simulation

Submodel	Parameter	Value / Unit
Engine	Type	ICE-Spark Ignition
	Displacement	2997 cm^3, 499,5cm^3/cylinder
	Maximum power	180 kW / 5500 rev/min
	Maximum torque	298 Nm / 3750 rev/min
	Low threshold for engine temperature	20 degC
	High threshold for engine temperature	80 degC
	Hot engine idle speed	700 rev/min
	Cold engine idle speed	1100 rev/min
	Hot idle consumption	500 g/h
Electric motor	Type	M&G
	Maximum power	18,5 kW
	Maximum torque	208 Nm
	Efficiency	0.85
	Maximum rotary velocity	9000 rev/min
Battery	Voltage	288 V
	SOC	0.9
	Cells in series per battery bank	8
	Battery banks in series	30
Gearbox	Final drive gear ratio	4,25
	Final drive efficiency	0,97
	1^{st} / 2^{nd}/ 3^{rd}/ 4^{th}/ 5^{th}/ 6^{th} gear ratio	3,5 / 2/ 1,4 / 1 / 0,85 / 0,7
	Gearbox efficiency	0,9
Vehicle	Kerb weight	1310 kg
	Wheel Inertia (4 wheels)	2,5 kgm^2
	Wheels	235/45R18 99V
	Maximum brake torque	1000 Nm
	Drag coefficient	0,3
	Frontal area	2,14 m^2

SIMULATION AND RESULTS

The simulation mode must follow the mission profile defined using tables and it will use the New European Driving Cycle (NEDC). The NEDC consists in the fact that the engine starts at 0 (zero) seconds and the emission sampling begins at the same time. The duration of the cycle is 1180 seconds.
The ECU analyses the driver command (acceleration and braking) in order to minimize the battery consumption and also the fuel consumption and CO_2 emissions. During the braking the electric motor behaves as a generator to charge the battery. Depending on vehicle speed, the electric motor or/and the ICE

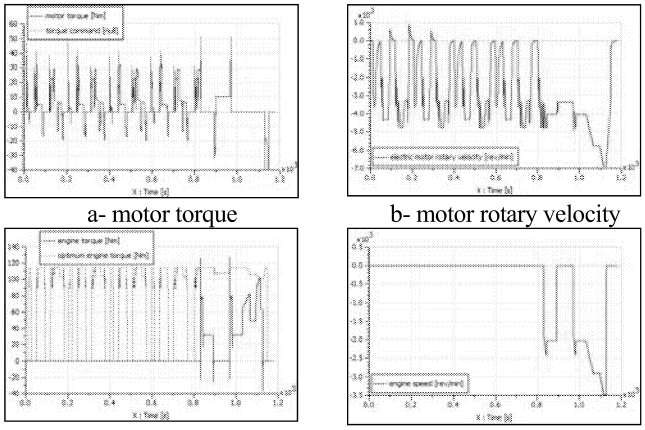
is used to realize the displacement of the vehicle. Depending on the load given by the driver and the battery state of charge the ICE can sustain the electric motor with partial or full torque and power with or without cylinder deactivation. If the optimum power of the ICE is higher than the requested one, when it is working, the difference could be used to charge the battery. Three of the cylinders were deactivated when the load value was 30% of the maximum engine load.
Comparing the curve of vehicle speed with the control speed in both cases (with and without cylinders deactivated) it can be observed that the vehicle speed follows the control speed, showing that the model is functioning correctly, with cylinders deactivated (figure 2). The settings of the driver allow a good agreement between the curves.



a-NCD b-CD

Figure 2: The evolution of the vehicle speed and the control speed during NEDC (1180s): a-NCD (without cylinders deactivated),
b-CD (with cylinders deactivated)

Figure 3 describes the behavior both of the electric motor and of the engine; it can be seen that the engine works at the final of the cycle for the last acceleration, when the battery is being charged. The behavior of the electric motor and of the engine is quite similar to the case of NCD, as shown in figure 4. Deactivating three of the cylinders won`t harm the behavior of the HEV. The electric motor will provide more torque when three of the cylinders are being deactivated, at the final of the cycle, when the ICE starts.

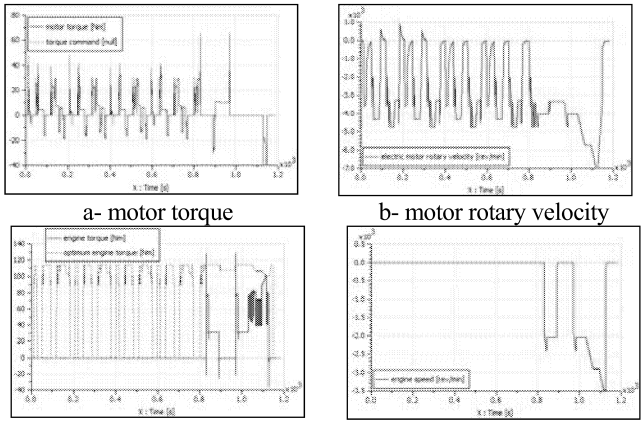


a- motor torque b- motor rotary velocity

c- engine torque d- engine speed

Figure 3: Behavior of the electric motor and ICE (NCD)

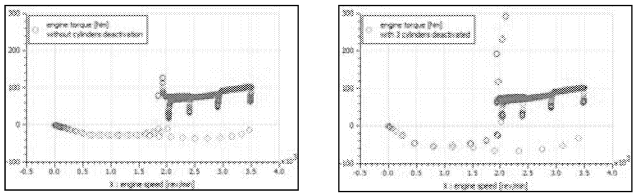
Evaluating the engine torque in both cases (figure 5), with three cylinders deactivated and, respectively, without deactivated cylinders, shows that the mission of the electric motor to achieve the propulsion meets the requirements of fuel economy and less harmful emissions. The electric motor ensures the displacement of the vehicle during all the NEDC assisted only by 3 from 6 of the ICE`s cylinders. ICE operates between 2000-3500 rev/min developing the demanded torque. The engine is generally used for a load higher than 50% to improve its efficiency. In this case, the electric generator limits the optimization of the system because the optimum engine torque cannot be reach.



a- motor torque b- motor rotary velocity

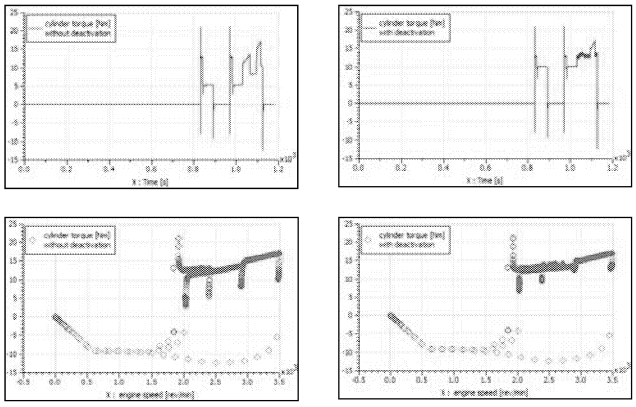
c- engine torque d- engine speed

Figure 4: Behavior of the electric motor and ICE (CD)



a-NCD b-CD

Figure 5: Evaluating the engine torque: a- without cylinders deactivation; b-with 3 cylinders deactivated



a-NCD b-CD

Figure 6: Evaluating torque from one cylinder
a- without deactivation; b-with deactivation

Deactivating some of the cylinders tries to realize fuel economy. The hybrid system does not suffer from these changes, keeping the required properties. Evaluating ECU behavior and considering the regenerating function of the batteries no changes occurred, as shown in figure 7.
During the NEDC the model without cylinders deactivation realized a fuel consumption of 349,07 g, while the battery state of charge reached 88,16 %, as shown in figure 8 and table 2 (NCD).

The model with three cylinders deactivated realized a fuel consumption of 305,55 g, while the battery state of charge reached 88,16 %, as shown in figure 9 and table 2 (CD).
In both cases, the ICE started at a vehicle speed of 15 m/s, at which the hybrid mode starts, after running 830 seconds of the cycle. While deactivating three of the cylinders the economy obtained reaches 12,47 %.

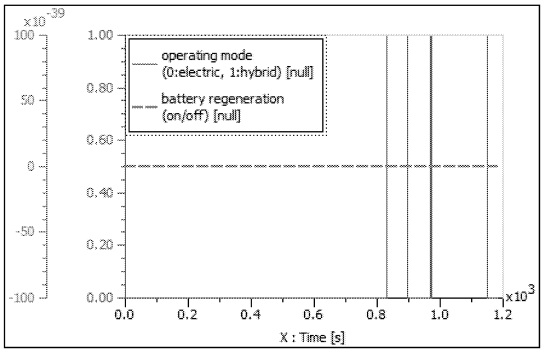
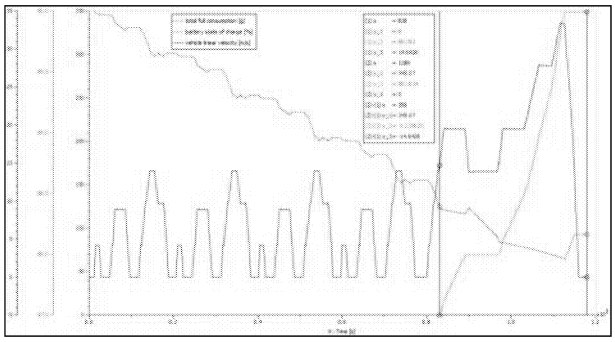


Figure 7: The vehicle mode (electric or hybrid mode) compared with the necessity to charge the battery


Figure 8: Total fuel consumption [g] / Battery state of charge [%] / Vehicle linear velocity [m/s] during the New European Driving Cycle (NEDC) without cylinders deactivated

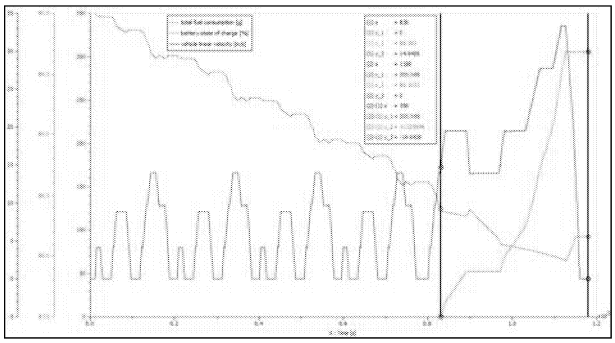

Figure 9: Total fuel consumption [g] / Battery state of charge [%] / Vehicle linear velocity [m/s] during the New European Driving Cycle (NEDC) with cylinders deactivated

Table 2: Results for fuel consumption, battery state of charge and vehicle linear velocity during NEDC

NCD		CD	
(1) x	= 830	(1) x	= 830
(1) y_1	= 0	(1) y_1	= 0
(1) y_2	= 88.393	(1) y_2	= 88.393
(1) y_3	= 14.6426	(1) y_3	= 14.6426
(2) x	= 1180	(2) x	= 1180
(2) y_1	= 349.07	(2) y_1	= 305.549
(2) y_2	= 88.1634	(2) y_2	= 88.1632
(2) y_3	= 0	(2) y_3	= 0
(2)-(1) x	= 350	(2)-(1) x	= 350
(2)-(1) y_1	= 349.07	(2)-(1) y_1	= 305.549
(2)-(1) y_2	= -0.229625	(2)-(1) y_2	= -0.229846
(2)-(1) y_3	= -14.6426	(2)-(1) y_3	= -14.6426

CONCLUSIONS

It is essential to know the evolution of the fuel consumption depending on the engine load when the cylinders are deactivated. Taking values from 10% to 90% of the maximum engine load (EL in figure 10) for the moment when the cylinders will be deactivated will result the dependence of the fuel consumption on this parameter. Following the example above there were studied the next cases: hybrid combined with cylinder deactivation (HEVCD), hybrid combined with usual ICE, without deactivation (HEVNCD), ICE with cylinder deactivation (NORCD), ICE without cylinder deactivation (NORNCD).

Table 3: Comparing studied cases

	HEV CD		HEV NCD		NOR CD	NOR NCD
EL	TFC [g]	SOC [%]	TFC [g]	SOC [%]	TFC [g]	TFC [g]
0,1	348,82	88,163	349,07	88,163	688,20	691,62
0,2	331,53	88,163			651,41	
0,3	305,54	88,163			615,76	
0,4	291,32	88,159			576,36	
0,5	280,4	88,159			554,88	
0,6	272,12	88,160			541,15	
0,7	267,28	88,161			535,87	
0,8	263,87	88,162			532,47	
0,9	263,87	88,162			532,61	

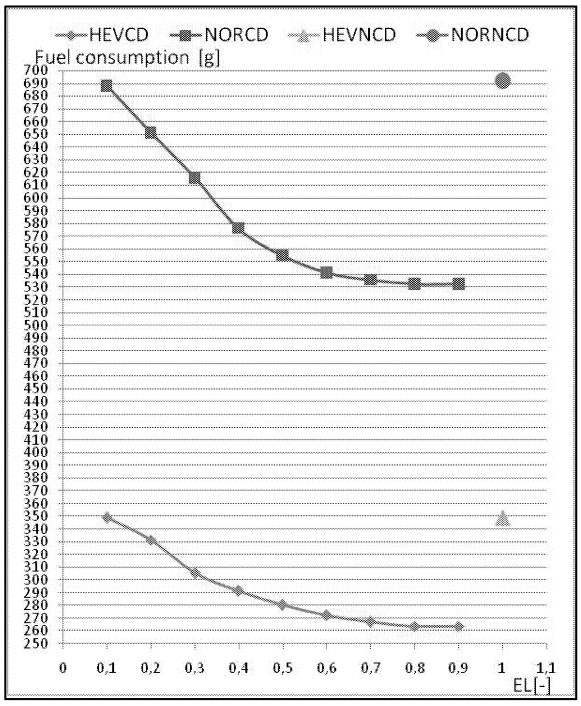


Figure 10: The fuel consumption during NEDC

Using both hybrid technology and cylinder deactivation, the fuel economy reaches high values. Comparing the HEVCD model with NORCD model the fuel economy using the hybrid technology was improved. The HEVNCD model fuel consumption is higher than the highest HEVCD fuel consumption. The HEV is able to operate approximately two times more efficiently than conventional vehicles.

The model presented in this paper provides an efficient tool for the simulation of a parallel hybrid vehicle associated with cylinder deactivation. It can be used for issues as the development, validation and optimization of the energy strategy, taking into account fuel consumption and pollutant emissions. This model can also be used as an initial application in order to build a more advanced model and to take into account more issues.

For a hybrid electric vehicle the electric motor and the battery are the core of economically functioning. Associating the cylinders deactivation less fuel consumption and CO_2 emissions will be obtained.

AMESim Software simulates different hybrid powertrain solutions and cylinders deactivation method in order to obtain data about fuel consumption and other parameters, and compares these solutions even if they were not yet physically achieved.

More efficient vehicles can bring important advantages to society in terms of environmental benefits. Hybrids will never be true zero-emission vehicles, however, because of their internal combustion engine. But improving cylinder deactivation method combined with hybrid technology the emissions and fuel consumption will go down.

AKNOWLEGEMENT

The present paper has been conducted within the framework of the `IDEI` contract, ID_1091/2007 and `POSDRU/6/1.5/s/19` contract, ID_7713/2007.

REFERENCES

Croitorescu, V., Maciac, A., Oprean, M., Andreescu, C, "Simulation Of A Spark Ignition Engine With Cylinders Deactivation", FISITA 2010

Oprean, I.M., "Automobilul modern (The modern automobile)", Editura Academiei Române, Bucharest, 2003

Negurescu, N., Pană, C., Popa, G.M., Racoviță, A., "Variable Valve-Control Systems for Spark Ignition Engine", SAE World Congress, 2001-01-0671, Detroit, 2001

Pană, C., Popa, M.G., Negurescu, N., Silvestru, C., "Dinamica motoarelor cu ardere internă", Editura Matrix Rom, 2005

Grunwald, B., "Teoria, calculul și construcția motoarelor pentru autovehicule rutiere", Editura Didactică și Pedagogică, 1980

Croitorescu, V., Maciac, A., Oprean, M., Marin, A., Andreescu, C., "Hybrid Powertrain Simulation", Proceedings ISC2008

Croitorescu, V., Maciac, A., Negruș, E., Oprean, M., Andreescu, C., "Simulation of an all wheel drive hybrid electric vehicle", Proceedings ESFA 2009

LMS Imagine Suite – Tutorials

http://www.bankrate.com/finance/auto/hybrids-not-the-only-answer-1.aspx

http://findarticles.com/p/articles/mi_m3012/is_3_179/ai_54216811/

http://www.star-motoring.com/news/story.asp?file=/2007/10/20/motornews/20071020234410&sec=motornews

http://www.freepatentsonline.com/6732506.html

http://www.freepatentsonline.com/6752121.html

http://motortrend.com/70/64622/the-general-forum/hybrid-or-cylinder-deactivation/index.html

http://green.autoblog.com/2006/10/08/a-tale-of-deactivating-cylinders-after-25-years-gm-finally-succ/

http://green.autoblog.com/2007/11/16/its-friday-videos-of-variable-transmission-cylinder-deactivat/

http://www.weather.com/activities/driving/greenvehicle/hevwhy.html

http://www.streetdirectory.com/travel_guide/49158/cars/why_hybrid_cars_are_better.html

http://www.fueleconomy.gov/feg/hybridtech.shtml

http://arsanavs.co.cc/reduce-co2-emission

http://www.faqs.org/patents/app/20100050993

BIOGRAPHY

VALERIAN CROITORESCU was born in Ramnicu Valcea, Romania, and attended the courses of University `POLITEHNICA` of Bucharest, in 2002, where he studied Automotive Engineering and obtained his Automotive Engineer Degree in 2007, being also the valedictorian. He prepared his final degree dissertation paper (diploma project) in France, at `Ecole Nationale D`Ingenieurs De Tarbes`, and also presented it in Romania, at University `POLITEHNICA` of Bucharest. He also obtained two Master of Science Degrees, after attended two Master programs, namely `Economicity And Security In Automotive Engineering` and `Environmental Management`, at University `POLITEHNICA` of Bucharest. His academic records include numerous awards and certifications. In September 2007 Valerian joined the academic stuff of Automotive Engineering Department. In August 2008, he founded AUTOJOBS.RO, a specialized web portal in automotive industry having the goal to help and provide job seekers and employers the opportunity to be connected. He is currently attending a PhD Program, having the main interest Hybrid Electric Vehicles Development, at University `POLITEHNICA` of Bucharest.

TRAFFIC SIMULATION

HIERARCHICAL ARCHITECTURE FOR URBAN TRAFFIC MANAGEMENT – SIMULATION

Monica Voinescu, Andreea Udrea, Simona Caramihai, Calin Munteanu
Department of Automatic Control
University "Politehnica" of Bucharest
Spl. Independentei, 313, Bucharest
Romania
E-mail: mvoinescu@ics.pub.ro

KEYWORDS
urban traffic management architecture, conservative networks controllers, genetic algorithms

ABSTRACT

This paper presents a study on urban networks and proposes a three layers traffic management architecture and a solution for congestion prevention in a traffic sub region by finding the optimal green-red light ratios. For congestion prevention, a macroscopic model was created by making an analogy between the traffic network and compartmental networks. Blockage is avoided by the use of a GAS nonlinear controller particularly designed for this type of network.

INTRODUCTION

As an essential part of modern society, traffic continually draws the attention of the research community in terms of modeling, simulation, traffic management, optimization and control, as well as traffic forecasting.
From a systemic perspective, traffic can be seen as both a complex system – large scale, dynamic, nonlinear - and as an open system.
Given the complexity and the particularities of urban traffic control and optimization problems, research in the area has turned towards intelligent techniques approaches, such as rule-based systems for predefined plan selection, fuzzy rules control systems, neural networks, or neuro-fuzzy systems, cellular automata modeling and evolutionary algorithms optimization (Sanchez, 2008).

TRAFFIC MANAGEMENT ARCHITECTURE

Due to the complexity of the system we decided on a three levels management architecture as follows:
At the bottom level, junction controllers are managed by software agents that implement specific autonomous behaviors to accomplish junction monitoring and control tasks. Each bottom level entity – henceforth called a junction agent – manages a single crossroad.
Junction agents on the bottom level are interfaced to the medium and top level of the architecture. The optimal crossroad cycle duration, the order of the phases and the green – red ratios are received from the second and third levels.
The communication between layers is bidirectional – the data acquired from traffic on the lower layer is sent to be processed and used by the upper levels, while processing results are dispatched as new control parameters from top to bottom.
In its turn, the lower level provides the upper levels with information such as: the car densities on each input and output road segments, the percentage of cars that leave each entrance towards each allowed exit direction, the maximal allowed green light per phase in order to prevent blocking the junction. Such information is measured or calculated online and stored within an internal world model of the junction agent. This model also holds the last received cycle duration, the order of the phases, green-red ratios, minimal and maximal green values durations and a mechanism for local congestion prevention that works in parallel with the upper levels and also as standalone module in case of faulty communication with the supervisors.
The second level addresses sub regions containing groups of crossroads and road segments and performs optimization tasks.
A group of junction agents are monitored, synchronized and coordinated by a regional controller/agent which deals with traffic optimization for that specific sub region. Its function is to increase the amount of cars that pass through that region by using an algorithm that calculates the optimal green-red ratios for each phase of the junctions.
The third level is the global supervisor that implements a series of control strategies as a function of the traffic dynamics. Because regional optimization does not necessarily yield a global optimum solution for the entire traffic system, sometimes leading to a deterioration of global performances, the implementation of a top level is needed. The function of this top level entity is to provide the durations of light cycles and the order of phases for the network, based on various traffic scenarios. This level also handles complex problems like prioritizing special vehicles (public transport and intervention vehicles), estimating the effect on the overall traffic dynamics of such priority schemes and of special situations such as accidents.
The top level supervisor selects the most suitable scenario from a range of predefined scenarios, function of each sub region dynamics and status. The scenarios stored by the top level supervisor are generally conceived offline, using the information sent over time by the bottom and medium levels. A coarse classification of traffic scenarios can be made based on the degree of junctions' load:

- extremely rarefied traffic – generally night and suburbs traffic: the traffic light cycle can be overridden and green can be accorded for any direction where a vehicle is approaching. This is a first arrived – first served strategy.
- medium density traffic – generally not leading to congestion – weekend/official holydays traffic.
- dense traffic – possibly leading to congestions.
- saturated traffic – congestion beginning to affect the neighboring junctions.

The strategies for the three latter scenarios can be: green wave or GA strategy on the second level, decentralized control using conservative network controllers.

THE CONSERVATIVE NETWORKS APPROACH

The bottom layer agents work in the timeframe of the real traffic environment, therefore their blockage prevention mechanisms must be based on fast response, direct methods derived from current measurements only.

In general, the green/red ratio for each phase doesn't change according to the requirements of traffic and is a factor that causes the appearance of blockages (along with the fact that the cars density is larger than the density for which that part of the network was designed).

In (Udrea, 2010) are detailed the considerations related to the modeling of traffic networks as conservative networks.

A conservative system (Bastin, 2006; Imsland, 2004) is a compartmental system with a network-like structure presenting nodes (compartments, buffers) that can contain a certain amount of matter (mass, elements, energy); $x(t)=[x_1(t)\ x_2(t)\ \dots\ x_n(t)]^T$ is the vector of the states of the system. Using the notations $f_{ij}(x(t))$ is the transfer rate (debit; flow), from compartment i to compartment j and $b_i(t)$ and $e_i(t)$ are the input and output flows in/out the system the balance is given by:

$$\dot{x}_i = \sum_{j\neq i} f_{ji}(x) - \sum_{k\neq i} f_{ki}(x) - e_i(x) + b_i, i = \overline{1,n} \quad (1)$$

where: $x_i(t) \in R_+, t > 0$; $f_{ij}, e_i : R_+ \to R_+$; $b_i(t) \in R_+$; and, if $x_i(t) = 0$ than $f_{ij}(t) = 0$ and $e_i(t) = 0$.

If this condition is satisfied and if $f_{ij}(x)$ and $e_i(x)$ differentiable, then: $f_{ij}(x) = r_{ij}(x)x_i$ $e_i(x)=q_i(x)x_i$, where the specific flows $r_{ij}(x)$ and $q_i(x)$ are continuous, differentiable and strictly positive, with positive function argument. So (1) becomes:

$$\dot{x}_i = \sum_{j\neq i} r_{ji}(x)x_i - \sum_{k\neq i} r_{ki}(x)x_i - q_i(x)x_i + b_i, i = \overline{1,n} \quad (2)$$

In order to model junctions, it was considered that the road segments represent network nodes - buffers that accumulate cars – and the directions of movement in a junction are the arcs that modeled the transfer of cars in the network.

If the storage capacity of a destination node is exceeded, the extra cars are going to accumulate in the junction causing blockage. To prevent this, when the capacity limit of a destination node is overcome, the green traffic light must change into red. The algorithm for the blockage prevention will be presented next.

The cars flows (number of cars per unit of time) are similar with transfer ratio (flow) between the nodes of a compartmental network. The categories of flows are d_i - inflow to the source node, v_j outflow from the destination nodes and f_{ij} outflows of the source nodes (transfer flows between source and destination nodes) – see figure 3.

The inflow to the source node is a time variable rate of cars that depends on the features of the section. The outflows depend on the state of node - the occupancy level of the node - and the state of the section which is connected with the node. These flows are defined as follows: $v_i(x_i) = \mu_i \frac{x_i}{x_i + 1}$ where μ_i is the flow of cars on the section i and the function $g(x_i) = \frac{x_i}{x_i+1}$ models the behavior of cars leaving the node. The outflow will increase monotonously to the value dictated by the section.

The cars flow can be decomposed into three components like the flow of the compartmental network: $f_{ij} = \beta_{ij} v_i(x_i)$ where $\sum \beta_{ij} = 1$ are the percentages of the total flow leaving the source node i which are distributed to the destination nodes j.

The compartmental systems have a series of important mathematical properties: are positive systems (if $r_{ij}(x)$ and $q_i(x)$ are differentiable the system is time invariant); they assure mass conservation $M(x) = \sum_{i=1}^{n} x_i$ and if (2) is written as:

$$\dot{x}_i = A(x)x + b \quad (3)$$

- $A(x)$ – the compartmental matrix- is Metzler (all elements except the ones on the principal diagonal are nonnegative ($a_{ij}(x) = r_{ji}(x) \geq 0$); the principal diagonal contains only non positive elements ($a_{ii}(x) = -q_i(x) - \sum_{j\neq i} r_{ij}(x) \leq 0$); and is a diagonal dominant matrix ($|a_{ii}(x)| \geq \sum_{j\neq i} a_{ij}(x)$).

$A(x)$ is nonsingular and stable, $\forall x \in \mathbb{R}_+^n$ if and only if the compartmental network is fully connected to the output flow (Bastin, 2006). The system's (3) Jacobian is:

$$J(x) = \frac{\partial [A(x)x]}{\partial x}$$

When $J(x)$ has a compartmental structure and is invertible, the system has a unique equilibrium point which is global asymptotical stable (GAS) for a system with input and output flows. $J(x)$ is singular; but the uniqueness of the equilibrium GAS point is preserved for al closed systems that are strongly connected.

The congestion that can appear for input flow demand larger than the network capacity can be avoided by use of a nonlinear controller with a compartmental system like structure.

The control problem states that, given a totally input and output connected compartmental network with limited flows and buffers capacities and with a certain demand for the input flows - d_i; the congestion prevention can be done by controlling the input flows $b_i(t)$ - injected in the system – by making them less than the demand $d_i(t)$:

$$b_i(t) = u_i(t)d_i(t), 0 \leq u_i(t) \leq 1,$$

where: $u_i(t)$ is a fraction of the $d_i(t)$. It is consider that the output flows are the measurable outputs of the system.

The control problem solution is a dynamic nonlinear controller:

$$\dot{z}_i = y_i - \Phi(z_i)\sum_{k\in Q_i}\alpha_{ki}d_k \,(i\in I_e)$$

$$u_j = \sum_{k\in P_j}\alpha_{jk}\Phi(z_k)\,(j\in I_i)$$

with: - $\alpha_{jk};(j,k)\in R$ - the characteristic parameters of the network; they satisfy the following property:

$$0\le\alpha_{jk}\le 1 \text{ and } \sum_{k\in P_j}\alpha_{jk}=1;$$

- $\Phi: R\to R_+$ is a monotonically increasing, continuous and derivable function, such that $\Phi(0)=0$ and $\Phi(\infty)=1$.

The closed loop equations lead to the expression:

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}=\begin{pmatrix}A(x) & B(d)K(z)\\ C(x) & G(d)F(z)\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix}\overset{def}{=}L(x,y)\begin{pmatrix}x\\ y\end{pmatrix}. \qquad (4)$$

L(x,z) is a compartmental matrix so the system is positive; is a closed system and the state variables are limited:

$$0\le x_i(t)\le\sigma; i=\overline{1,n}$$

$$z_j(t)\le\sigma; j=\overline{1,p}$$

The first objective of blockage control is obtained by using the controller in the proposed form: σ is less than the maximal capacity of the compartments x_i^{max}, such that the saturation avoidance is assured. Moreover, it can be observed that σ depends on the initial conditions *(x(0),z(0))*. Many applications impose that the real system begins from null initial conditions *(x(0)=0)*, σ can be chosen exclusively by using the initial conditions of the controller $z_j(0)$ and equals: $\sigma=\sum_{j=1}^{p}z_j(0)$.

The command measures $u_j(z)$ take values between 0 and 1, so, with the restriction $0\le\Phi(z_i)\le 1$, it results: $0\le uj(z)=\sum_{k\in Pj}\alpha_{jk}\Phi(z_j)\le\sum_{k\in Pj}\alpha_{jk}=1$.

Since the controlled network is input and output fully connected and due to the controller's structure, (4) is a strongly connected compartmental closed system. If the controlled system's Jacobian is a compartmental matrix, then the Jacobian of the closed loop system is a compartmental matrix and for a certain demand flow, there is only one equilibrium point and it is GAS ($\mathbb{R}_+^n$)

A good choice of Φ in order to satisfy the conditions is $\Phi(z_i)=\frac{z_j}{z_j+\varepsilon}$, where - ε is a small positive constant.

For the modeling and blockage prevention example an X-crossroad with two double phases per traffic lights cycle (phase generalization) is considered. During each double phase, the cars can enter the crossroad from opposite direction and can go towards any of the three possible exits – Figure 3. a. A control algorithm must prevent blockage in the following circumstances: large inflow, small outflows, and destination buffers almost saturated.

For this study the simplified model for this type of phase was used: the effect of the intermediary buffers is similar to a saturation on the input flow allowed to go right – equal to the crossroad capacity for left turnings.

The model for this type of phase is:

$$\begin{cases}\dot{x}_1=d_1u_1-v_1(x_1); & \dot{x}_2=\beta_{12}v_1(x_1)+\beta_{52}v_5(x_5)-v_2(x_2)\\ \dot{x}_3=\beta_{13}v_1(x_1)-v_3(x_3); & \dot{x}_4=\beta_{14}v_1(x_1)+\beta_{54}v_5(x_5)-v_4(x_4)\\ \dot{x}_5=d_2u_2-v_5(x_5); & \dot{x}_6=\beta_{56}v_5(x_5)-v_6(x_6)\end{cases}$$

and the controller is:

$$\begin{cases}\dot{z}_2=v_2(x_2)-\Phi_2(z_2)(\alpha_{12}d_1+\alpha_{52}d_2) & \dot{z}_3=v_3(x_3)-\Phi_3(z_3)\alpha_{13}d_1\\ \dot{z}_4=v_4(x_4)-\Phi_4(z_4)(\alpha_{14}d_1+\alpha_{54}d_2) & \dot{z}_5=v_6(x_6)-\Phi_6(z_6)\alpha_{56}d_2\end{cases}$$

Two commands u_1 and u_2 are:

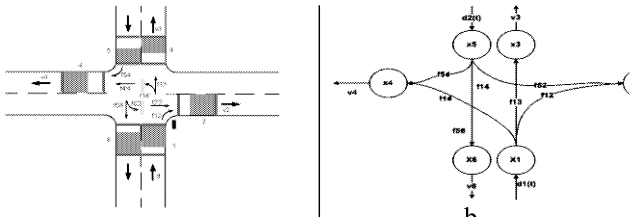
$$\begin{cases}u_1=\alpha_{12}\Phi_2(z_2)+\alpha_{13}\Phi_3(z_3)+\alpha_{14}\Phi_4(z_4)\\ u_2=\alpha_{52}\Phi_2(z_2)+\alpha_{56}\Phi_6(z_6)+\alpha_{54}\Phi_4(z_4)\end{cases}$$

where: $v_i(x_i)=\mu_i\frac{x_i}{x_i+1}, i=\overline{1,4}$ are the total evacuation flows; $\Phi_i(z_i)=\frac{z_i}{z_i+1}, i=\overline{1,4}, \varepsilon=0.1$ are the virtual evacuation flows from the nonlinear controllers nodes; μ_i is the maximal allowed flow that can exit from compartment *i*; α_{ij},β_{ij} are stand for the proportion of the flow that exits from *i* node.

The simulations were done in Matlab- Simulink. The used values were determined from real data measured on a typical congested crossroad; the maximal evacuation flows - μ_i – were also measured such they reflect the real traffic situation; they can be adapted in real-time from measured data:

$$\mu_1=300/60,\mu_2=10/60,\mu_3=25/60,\mu_4=5/60;$$

$$\alpha_{12}=\beta_{12}=0.2,\alpha_{13}=\beta_{13}=0.7,\alpha_{14}=\beta_{14}=0.1$$



a. b.

Figure 3: a. X-crossroad ; b.- model

The maximal evacuation flows - μ_i – are chosen such they reflect the real traffic situation and are measured in cars/sec. The open loop and closed loop results are showed in the fig. 7. In open loop mode, the congestion appears because the cars accumulate in the crossroad because the following buffers' capacity is exceeded.

The congestion is avoided – the cars do not accumulate; the inflow is limited to a value that the crossroad can process.

It can be observed that the nonlinear controller can avoid the congestion. It generates a command that limits the large inflow – letting just a portion of the flow to pass in order not to block the crossroad. The command values are always between 0 and 1. When obtaining an extremely small inflow – it is preferred not to activate that phase, because the green time will be extremely small and the cars passing through the crossroad would probably block it.

The command transfer (command values in green light duration) to the traffic light controller, which is the actuator of the crossroad system, can be done using this formula:

$$dt_v = d_c t_{vmax}$$

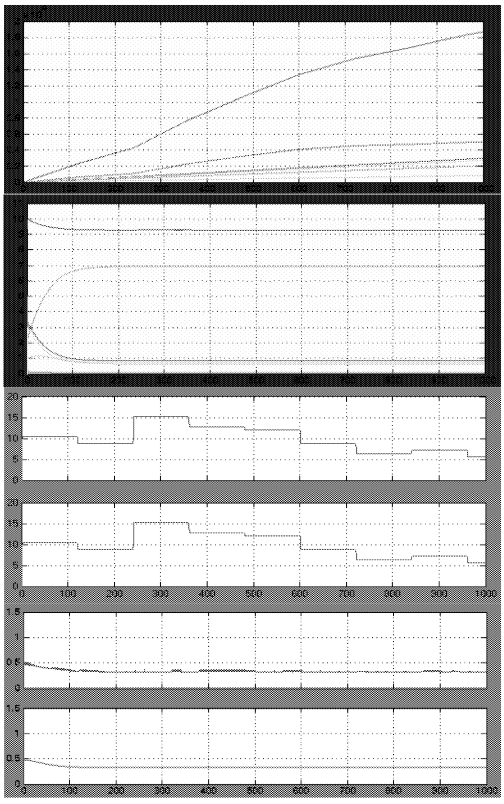
where: d – is the determined mean flow between the input node and the destination node; d_c – inflow calculated by the controller:

$$d_c = du_m$$

- u_m – mean command value
- t_{vmax} –the maximal green time allowed for a phase;
- Δt_v – the green light duration – calculated by the controller:

$$t_v = \frac{d_c t_{vmax}}{d} = \frac{d u_m t_{vmax}}{d} = u_m t_{vmax}$$

This duration is a percent - u_m - of the maximal green light. If u_m is too small, that phase will not begin. This mechanism can be used along with the optimal values transmitted from the second level in order to prevent any eventual blockage situation and, also, alone, in case of connection failure.



a.

b.

c.

d.

Figure 4 : a. open loop evolution; b. closed loop evolution of the 4 outflows; c. inflows d_1 and d_2 (flow - cars/sec); d. inflows d_1 are d_2 limited by the commands u_1 and u_2 (flow-cars/sec)

THE GENETIC ALGORITHMS ZONAL SUPERVISOR

The second layer of the proposed architecture consists of zonal supervisor agents whose task is to monitor, coordinate and improve the operation of several junction agents within a delimited area.

The zonal supervisor works on a larger timescale, implementing tasks like statistic estimation and prediction planning for zonal optimization.

This section studies the effect of genetic optimization of green light timing on the traffic loads of junction inputs in a given area. Two cases have been considered: first, the optimization of a single junction in the traffic network of the area considered (Figs. 6 and 7), and second, the optimization of multiple junctions in the traffic network area (Fig.8). Simulations have been run with (i) the total cycle time constant (Fig. 6) and (ii) the total cycle time also an optimization variable (Figs. 7 and 8. a,b,c).

The optimality of a solution is evaluated in terms of minimizing the weighted average of the waiting queue lengths on all road segments entering the junction/junctions. The computational and temporal requirements to reaching a solution are also taken into consideration to verify the feasibility of the implementation in the real traffic environment.

For example, we have chosen an urban area of three interconnected junctions - two cross junctions (CJ1 and CJ2) and one T-junction, all built on a model described in (Voinescu, 2009)..

We have simulated this traffic network in Matlab 2010a with the original timing settings over a time span of an hour. Initial simulation results show imminent congestion (Fig. 5).

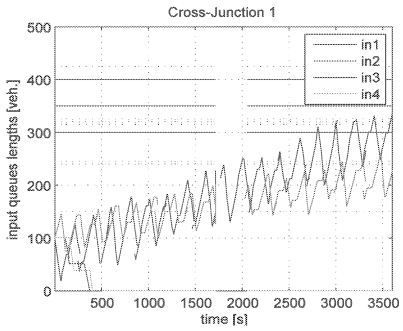


Figure 5: Simulation of CJ 1 with the original timing settings

A zonal supervisor was added, running a genetic algorithm for optimizing green light phase timings, implemented in Matlab by the use of the Genetic Algorithms and Direct Search Toolbox. Also, assuming a multi-core cluster for the implementation of the supervisory level algorithms we have used the Parallel Processing Toolbox, running each junction model on a separate Matlab worker.

The genetic algorithm was run with the following options:

- the default type *double* for chromosomes, within a certain lower bound and upper bound [lb, ub] derived from the time limits imposed on each phase;
- population size of 15 chromosomes;
- a maximum number of 15 generations as a stopping criterion or a maximum of 7 stall generations;
- stochastic uniform selection function;
- adaptive feasible mutation, randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation

Simulating the junctions in the area considered shows imminent congestion.
Fig. 6 presents the waiting queue lengths for the same cross junction after performing genetic algorithm optimization of green light phases on a time horizon of 1200 seconds, imposing a constant total cycle (CTC) time. The traffic network was then simulated over the time span of an hour, showing that the congestive state on input 2 was permanently removed, while the accumulation on input 1 was reduced.
Figure 7 shows results of GA optimization when removing the constraint of constant duration for the total light cycle. It can be seen that traffic load on all connected inputs is preserved within acceptable values for the entire time of the simulation (1 hour).
However, the improvements visible in Figures 6 and 7 for Cross-Junction1 (CJ1) are obtained in disregard of the other junctions in the network, which, in some green phase timing scenarios of CJ1, may be driven towards congestion.

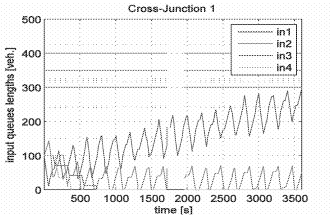


Figure 6: Simulation of CTC-optimized CJ 1 input queues

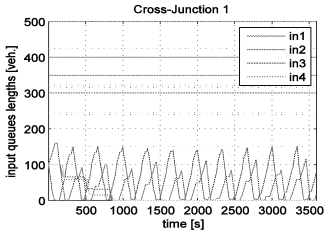


Figure 7: Simulation result for optimized CJ1

To overcome this problem, we have considered designing a multi-objective genetic algorithm, taking into account the most important or all junctions in a sub region and optimizing by all queue lengths.
Simulation of all three junctions in the area considered over the time span of an hour are presented in Figures 8 a, b, and c. As can be seen, all input queues remain consistently below a certain threshold considered as a fractional limit of the total road segment capacity (red horizontal line in all Figures).

CONCLUSIONS AND FURTHER RESEARCH

Time is a critical issue for the problem of congestion prevention, therefore we shall proceed in studying methods for reducing the runtime of the genetic algorithm in finding a solution. One mean towards achieving this objective is adopting a hybrid solution in conjunction with artificial neural networks (ANNs) (Issa, 2009). Since the most time consuming element in running a genetic algorithm optimization is evaluating the fitness function at each step, we propose the offline training of a neural network model for the urban area under consideration which will then be used in evaluating the chromosomes produced by the GA. To ensure more accuracy for the ANN model of a traffic area a library of such models may be built to take into consideration the steep traffic load variations such as daytime vs. nighttime traffic, work shifts: morning time vs. evening, working days vs. weekends and holidays.

We also intend to use in the future the NCIT-Cluster, a high performance computing cluster infrastructure available at the Politehnica University of Bucharest to test the distributed implementation presented above.

ACKNOWLEDGEMENT

The paper presents partial results of a CNMP budgeted project: “Intelligent Techniques for the Modeling, Analysis, and Optimization of Urban Traffic”, contract no. 71108.

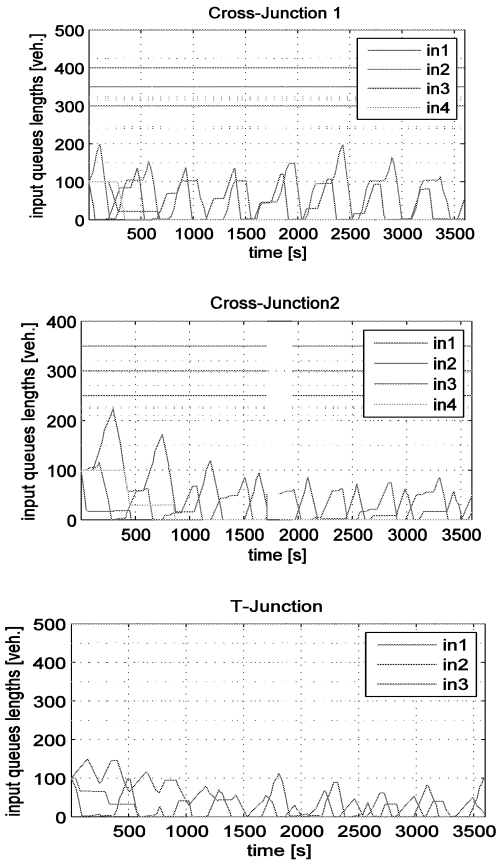


Figure 8: a., b., c. Simulation results after optimisation

REFERENCES

Bastin G., Guffens V., 2006. “Congestion control in compartmental network systems “, *Systems and Control Letters*, Vol. 55(8), pages 689 - 696

Caramihai S., Voinescu M., Munteanu C., Dumitrache I., 2009. “An Agent-Based Approach for Urban Traffic Simulation and Control“, *Proceedings of the 6-th IFAC International Workshop on Knowledge and Technology Transfer in/to Developing countries: Automation and Infrastructure*, Ohrid, Macedonia

Imsland L., Foss.A., 2004. Positive Systems,Ed. Springer Berlin

Issa, F., 2009, “Contributions to Optimizing Control Strategy Assignment in Collaborative Mobile Robots Applications”, *PhD. Thesis*

Sanchez Medina, J. J., Galan Moreno, M. J., Rubio Royo, E., 2008, “Evolutionary Computation Applied to Urban Traffic Optimization”, *Advances in Evolutionary Algorithms*

Voinescu, M., Udrea, A., Caramihai, S., 2009, "On Urban Traffic Modelling and Control", *Journal of Control Engineering and Applied Informatics*, Vol 11, No 1

MONICA VOINESCU is a BSc (2006), MSc (2008) and Ph.D. student of Automatic Control and Computers Faculty, University Politehnica of Bucharest. Her research interests are intelligent systems and multi-agent systems.

ANDREEA UDREA received the BSc and MSc degrees at Automatic Control and Computers Faculty, University Politehnica of Bucharest in 2006, respectively 2008. Since then she is a teaching assistant and PHD student and her research interests include numeric control and chaos teory.

SIMONA CARAMIHAI received her Ph.D. in 1997 and is a full time Professor with the Faculty of Automatic Control and Computers, University Politehnica of Bucharest. Her research interests are discrete-event systems, modeling, hybrid systems, interoperability and knowledge management.

Simulation of Traffic Operations for Sea-crossing Bridges

Guiyun Liu[1]
Pengjun Zheng[1,2]
[1]Maritime Faculty, Ningbo University
818 Fenghua Road, Ningbo, China, 315211
E-mail: liuguiyun@nbu.edu.cn
[2]Transportation Research Group, University of Southampton
University Road, Southampton, SO17 1BJ UK
E-mail: pjzheng@gmail.com

KEYWORDS
Sea-crossing bridge; traffic operation; microscopic simulation

ABSTRACT

As a part of the research project "Traffic Operations for Sea-crossing bridges", a study was carried out with the aim to investigate the effects of marine environment on traffic operations on bridges. A microscopic simulation model was developed based on fuzzy logic which was then calibrated to observed driving behaviour. Initial simulation results suggested that the simulation can reproduce traffic operation on bridges realistically, and the microscopic simulation could be a very useful tool in developing traffic control and management strategies for bridge traffic operations.

INTRODUCTION

With the rapid economic development in China, sea-crossing bridges are increasingly becoming an important part of arterial road networks in coastal area. Just in Hangzhou Bay region (part of Yangtze River Delta), there have been three important sea-crossing bridges opened to traffic in recent years: the Hangzhou Bay Bridge, the East China Sea Bridge and the Zhoushan Bridge. Among these, Hangzhou Bay Bridge is the longest sea-crossing bridge in the world (36 kilometres or 22 miles long) with a construction cost of 11.8 billion CNY (US$1.70 billion). As illustrated in Figure 1, the Bridge is an S-shaped stayed-cable bridge with three lanes in each directions, spanning across the Hangzhou Bay on the East China Sea and is linking Ningbo's Cixi county in the South to Jiaxing in the North.

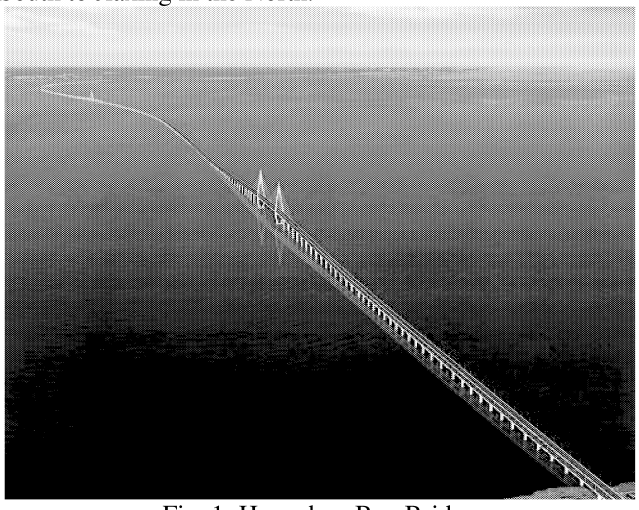


Fig. 1: Hangzhou Bay Bridge

The Hangzhou Bay Bridge consists of two navigable bridges, in-water sections such as bridge approaches, and bank lead roads. The South navigable bridge is a cable-stayed bridge with an A-shaped single tower, double-cable and steel box-girders (main span 318m). The North navigable bridge is a cable-stayed bridge with twin diamond-shaped towers, double cable and steel box-girders (main span 448m). The total length of sections on the sea is 32km.

Sea-crossing bridges are usually constructed to provide a shortcut link on existing road networks or a new and only connection to remote sites on the sea. For instance, the Hangzhou Bay Bridge shortens land transportation distance from Ningbo to Shanghai by 120km and travel time from four hours to two hours by providing a shortcut. On the other hand, East China Sea Bridge provides the only road connection to a newly developed seaport remotely located offshore, as shown in the Figure 2.

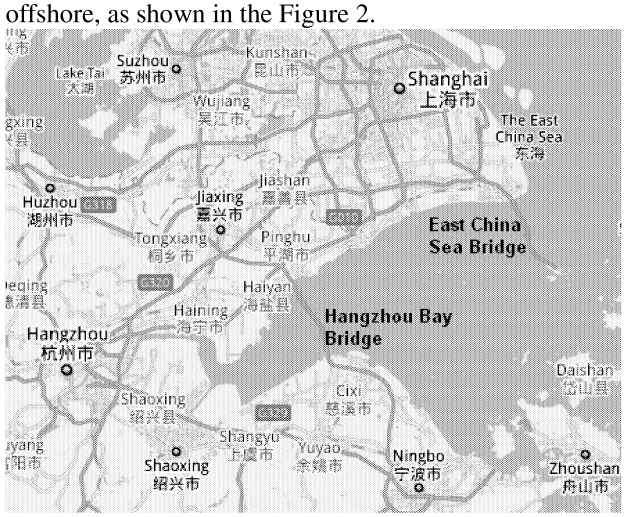


Fig. 2: Bridges in Hangzhou Bay

Sea-crossing bridges are high value assets and are of pivotal importance to the traffic operation of strategic road networks. Optimal traffic operations of bridges can not only affect throughputs of bridges themselves but also the overall efficiency of wider network. As sea-crossing bridges are located in the marine environment, they are vulnerable to heavy weather conditions not typical on highways, and drivers may exhibit different driving characteristics in response to changes in such aspects as landscape, visibility, high wind and other weather conditions. These may affect traffic operations on sea-crossing bridges. Understandings of traffic operation on sea-crossing bridges will be essential for effective management of such high value infrastructures. Despite extensive studies on traffic operations on highways,

however little has been reported on the traffic operations of sea-crossing bridges. The research reported in this paper is part of an empirical and simulation study of traffic operation of sea-crossing bridges. Traffic operations will be investigated using microscopic simulation based on detailed observation of driving behaviour on a sea-crossing bridge, the Hangzhou Bay Bridge. It is believed that understandings obtained from this research is applicable to many other large scale sea-crossing bridges, and will be very useful for the development of be-spoken traffic control and management strategies of bridges.

METHODOLOGIES

Traffic operations can be examined microscopically and macroscopically. Traditional traffic flow theories describes macroscopic traffic phenomena in terms of fundamental diagram which shows relationships between traffic flow rate, traffic speed and traffic density (Greenshields, et al., 1935; May, et al., 1968). The macroscopic theory is based, however, on fluid analogue of traffic flow with the assumption that individual vehicles behave exactly the same; there is a gap between the actual situation and assumption.

On the other hand, traffic operation can also be examined from microscopic interactions between vehicles. Car following theory describes dynamic interactions between leading and following vehicles on the same lane using mathematical models, focusing on the changes of distance headway between vehicles in different traffic speed. As drivers can only have direct control on the speed of vehicle (through the accelerator and brake), most of the car following theories took the form of a function using speed of a following vehicle as output and relative speed and distance headway to a leading vehicle as input. By considering all interactions between vehicles, it is possible to understand and describe the traffic operation on the lane, so that linkages between micro behaviour and macro characteristics of traffic flow can be established. For example, Gazis and Hermen (1961) derived a traffic flow-traffic speed curve from simple linear "stimulus - response" car following model.

The aim of this research is to understand traffic operations on sea-crossing bridges and to identify possible traffic control and management measures to improve bridge capacity and service level, especially in adverse weather conditions. This would need to examine traffic characteristics as a result of behavioural adoption when driving on bridges. Microscopic traffic flow models are established based on microscopic driving behaviour and have the ability to simulate complex traffic operations under different driving environment and control conditions. The approach will be especially useful to explore possible scenarios which can not be observed directly. For example, currently, traffic level on bridges is relatively low, so that traffic operations at high demand and under the influence of marine environment cannot be examined from observation.

It is therefore appropriate to investigate traffic operations on sea-crossing bridges using microscopic simulation. By changing model parameters of the vehicle-driver entity in the micro simulation, which can be observed from driving behaviour in real traffic, different characteristics of traffic operation can be explored and investigated (Zheng et al., 2005). The key will be driving behaviour observations to calibrate behavioural model parameters and microscopic traffic flow simulation to study the traffic operation and traffic flow characteristics under different scenarios of interest.

REVIEW OF CAR FOLLOWING MODELS

Transportation systems are a collection of driver-vehicle elements moving on physical networks, each element interacts with other elements and the environment. Behavioural models govern the movement of the driver-vehicle element. Accurate traffic operation is produced by programming objects to interact in a natural way. Of the several behavioural models defined in the simulation, the most important is the car following model which governs longitudinal separation between vehicles at different speed, which determine the stability and capacity of traffic operation. This is especially important for bridge traffic operation as other factors such as lane changing and merging (no on ramps for the length of the bridge) are relatively insignificant.

Car following behaviour is one of the most important factors closely related to traffic operation. Previous studies have established a direct link between microscopic car following behaviour and macroscopic traffic. Many car following models have been developed in the last half-century (Brackstone and McDonald, et al., 2000). The modelling approaches range from traditional manual control theory to fuzzy logic methods. Well-known models included GM model, cross-over model, stopping distance model, action point model and fuzzy logic model etc.

GM models were developed in the late 1950's by a General Motors Research Group. The fundamental concept behind these models was stimulus-response theory. A driver was treated as an information processor, and it was assumed that there existed a stimulus-response relationship that could be expressed as a form of stimulus-response equation:

$$\text{Re}sponse = Sensitivity * Stimulus$$

The response was generally the acceleration (deceleration) rate of the following vehicle. Chandler et al. (1958) obtained a linear stimulus-response equation using relative speed as the argument of the stimulus function. Gazis et al. (1959) proposed that the sensitivity was inversely proportional to the space headway. A more general form of the sensitivity term was proposed in 1961, resulting in a generalised representation of the General Motors (GM) car following model:

$$x''_f(t+\tau) = \frac{\alpha [x'_f(t+\tau)]^m}{[x_l(t) - x_f(t)]^n}[x'_l(t) - x'_f(t)]$$

where α, m, n are parameters to be determined (Gazis, et al., 1961).

GM models are not full-functioning microscopic simulation models because only speed control behaviour is considered. Another important aspect in car following, i.e., the headway control behaviour, was not modelled. The reason is that human bandwidth (or approximately, gain, λ) in headway

control is much lower than that of speed control, so that the headway control term is ignored. This has been experimentally verified and is not a problem when modelling the dynamics of car following behaviour because speed control is dominant in a car following process. The only steady state in the model is, however, found at zero relative speed irrespective of the following distance. Headway adjustment at a relative speed of zero will be impossible according to the model, which, of course would result in unrealistic simulation behaviour.

For generic close-loop tracking performance, McRuer et al. (1959) developed a crossover model by directly looking for the relationships between perceived error (*e(t)*) and system output (*O(t)*). The underlying assumption in a crossover model is that the human responds in such a way as to make the total open-loop transfer function behave as a first-order system with gain and effective reaction time. That is:

$$O(t) = k\int e(t-\tau)dt \quad \text{or} \quad O'(t) = k \cdot e(t-\tau)$$

This implies that a human operator is able to adjust his or her performance according to the dynamics of the system to be controlled to achieve a good control over the entire human-machine system.

It can be noted that in the mathematical form, the crossover model is identical with the linear form of GM model (*m=n=0*). However, it is based on more generic manual control theory and discussed under frequency domain.

Allen et al. formulated a simple crossover car following model characterised by crossover frequency (gain, K) and the reaction time (τ) (Allen, et al., 1997). The open-loop input-output relationship of the driver-vehicle system is modelled as a linear first-order differential equation. The output of the model is acceleration rate and input to model is the perceived relative speed delayed by a time τ. Allen et al. also devised an extended crossover model by taking headway error into account.

Stopping-Distance Model was derived based on the assumption that a follower always maintains a safe following distance so that he/she could bring his/her vehicle to a safe stop if the leader came to a sudden stop. At any time *t*, this required a following distance of:

$$s \geq \frac{v_f(t+\tau)^2}{2A_f} + \frac{v_f(t+\tau)+v_f(t)}{2}\cdot\tau - \frac{v_l(t)^2}{2A_l}$$

where, v is speed, τ is the reaction time of follower, A_f is the maximum acceleration the follower wishes to undertake, A_l is the maximum acceleration of leader estimated by follower. Subscription f represents for following vehicle and l for leading vehicle (Gipps, 1981). The model has seen widespread use in microscopic simulation because only little calibration is needed, i.e., the expected maximum acceleration the leading vehicle and the maximum acceleration the following vehicle wish to apply. However, the assumption that a driver always maintains a safe headway is not consistent with empirical observations.

Action point model was established on human motion perception theory. Michaels (1965) reporting on human perception of motion, stated that the dominant perceptual factor, or the stimulus, in a car-following situation, was the changing rate of visual angle (angular velocity). He suggested that a driver would take proper acceleration-deceleration action if the angular velocity were above a threshold. If the driver remained under this threshold, his/her action would be on whether he/she could perceive the change of space between them, i.e. at the 'Just Noticeable Distance'. That is, the driver is only able to modulate the acceleration above that threshold, and will stay with the last acceleration rate until another threshold is broken.

The Mission model was the first microscopic car following model to incorporate perception thresholds (Leutzbach, et al., 1986). More thresholds were defined such as relative speed threshold for the perception of opening (OPDV) and closing (CLDV) etc. Car following situations were further divided into sub-groups, e.g. uninfluenced driving, closing processes, following processes and emergency situations. A driver was supposed to adopt different acceleration behaviour in different situations if and only if the perceived physical (perception) stimulus exceeded certain minimum values, i.e., the thresholds. The model took into account the human threshold of perception, and thus established a more realistic rationale. However, most calibration efforts were focused on identifying threshold values, while other important aspects, the modulation of acceleration above a threshold had seen less detailed considerations, normally assumed to be a constant or generated from a distribution. In addition, further analysis of model dynamics was virtually impossible, because the model was established on several different behaviours and switching between them is random according to the distribution of the threshold.

Fuzzy logic car following models describes a car following law using linguistic terms and associated rules, instead of deterministic mathematical functions. This is an important advantage since the system can be described in a natural manner that reflects the imprecise and incomplete sensory data provided by human sensory modalities.

McDonald et al. (1997) developed a fuzzy logic car-following model based on an implicit speed and headway control assumption, which used relative speed and distance divergence as the principal inputs and acceleration-deceleration rate as output. A rule base in natural language was established to map the input and output space. The model was validated with time-series data collected by an instrumented vehicle and more consistent results had been achieved (Wu, et al., 2000, Zheng, et al., 2003). Kikuchi et al. (1992, 1996), on the other hand, assumed the inputs of model to be distance between leading vehicle and following vehicle (DS), relative speed (DV) and acceleration/deceleration rate of leading vehicle (ALV). Two models differed in the form of output membership function and their deffuzzification method. The former model used five triangle membership functions while the later used several linear ones.

In this research, fuzzy logic car following models were considered to be most suitable because a generic fuzzy logic model that is able to approximate any existing models as well as describe new formulations (Buckley, 1993 based

on observation of driving behaviour.

SIMULATION STUDIES

Traffic operations on sea-crossing bridges were investigated using microscopic simulation to explore the effects of different environmental factors. The interactions between vehicles in follow-the-leader situation were investigated for bridge driving. In order to represent longitudinal interactions accurately, a specific fuzzy logic car-following model was developed, with environmental influences on driving behaviour accounted in the rule base, as well as in membership functions. For example, the membership for the fuzzy variable 'distance headway' was adjusted towards more conservative value when driving was affected by visibility or high wind (so that a 30-meter headway with a maximum memberships of 'ok' became 'too close'). Lane-changing processes were considered to be the same as in ordinary highway driving so that a fuzzy logic lane changing model was adopted directly (Zheng, 2003). However, the desire for lane changing was adjusted according to observation.

A research-oriented microscopic simulation model, FLOWSIM (Fuzzy Logic based roadWay traffic SIMulation) was used in the investigation. The FLOWSIM model is a windows-based object-oriented microscopic simulation model that has been developed by the Transportation Research Group of the University of Southampton through EPSRC funding since 1995 (Wu and McDonald, 2000). The model was established using fuzzy sets to describe driving behaviour in a natural way. Fuzzy logic car following, lane changing and merging models were used to model behaviour of driver-vehicle units. Emphasis was placed on research undertaken to develop fuzzy logic driving behaviour models. Model validation results had shown that the fuzzy model was able to closely replicate real systems (Wu, et al., 2003).

FLOWSIM is able to implement user-defined longitudinal control logic to represent specific driving behaviour precisely, and some source programs were modified to provide special simulation abilities for this investigation. The core simulation model used was developed and validated through a number of research projects based on traffic observations on a UK motorway (Zheng, 2003). Modifications to the model for this investigation included:

- A new fuzzy logic car following model for longitudinal control behaviour was implemented, which can account for behavioural changes due to changes of driving environment
- The lane change model was modified by reducing desire to change lane as there is no on and off ramps in the whole stretch of the bridge, so there will be no lane-changing desire to either avoiding merging traffic or getting into lane for exit.
- Distribution of desired speed was re-calibrated based on speed limit imposed (100 km/h, varied based on type of vehicles), and better discipline of bridge traffic achieved by comprehensive monitoring system

Through these modifications, a bespoke microscopic simulation environment was created which enabled traffic operations on sea-crossing bridges to be explicitly modelled.

The traffic operation was investigated based on actual layout of the 3-lane sea-crossing bridges. Layouts of carriageways, hard shoulder and collision rails are illustrated in Figure 3. By varying traffic demands and environmental factors such as visibility, wind etc, key features of traffic operation were readily addressed using this 32 km corridor.



Fig. 3: Three-lane Carriageways on Hangzhou Bay Bridge

PRELIMANARY RESULTS AND CONCLUSIONS

To verify the effectiveness of the proposed microscopic simulation model, traffic operations on the sea-crossing bridge was simulated based on four traffic demand levels (1600 to 2200 vph at 200 vph increment per lane on average) and two weather conditions (Normal and Reduced Visibility). Each scenario was simulated for a two hour period. Relevant traffic flow and journey time data were collected using emulated loop detectors and ANPR cameras. Locations of those emulated loop detectors are in the 1/3, ½ and 2/3 length of the bridge. A screen shot of the simulation is shown in Figure 4.

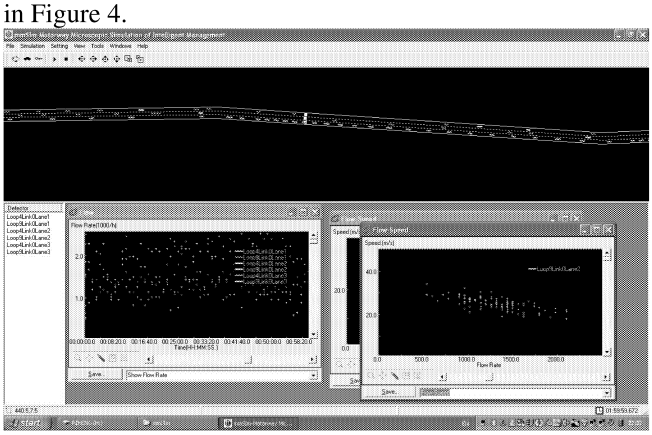


Fig. 4: Screen shot of Microscopic Simulation

To help understand traffic operation in the various situations, speed-flow relationships were examined. The relationships were derived from the loop detector data at three locations, based on average of three lanes. The fundamental speed-flow diagrams for normal traffic operation and traffic operations at reduced visibility are shown in Figure 5 and 6 respectively.

It can be seen that traffic operation on the bridge demonstrated similar phases such as free flow, near capacity and congestion as are typically found in traditional traffic operations on highways. From Figure 5, it can be seen that

traffic speed can sustain when traffic flow reaches as high as 2300 vph. This may be considered to be slightly higher than that can be achieved on highways. This may be because a slightly lower speed limit was adopted, and also, because traffic disturbances from on and off ramps did not exist in the simulated bridge traffic operation. This implies that a slight higher capacity may be achieved long bridge compared to normal highway.

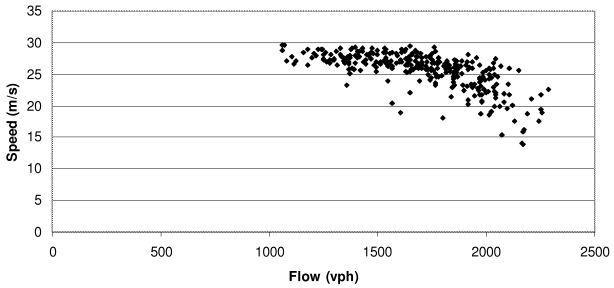


Fig 5: Normal Traffic Operation

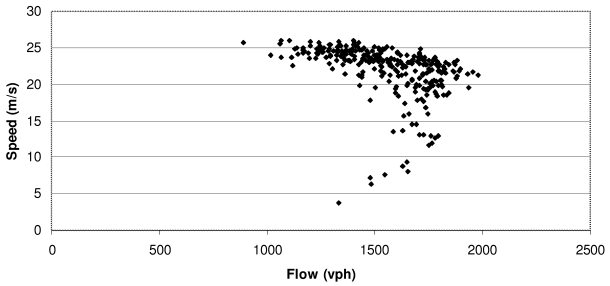


Fig 6: Traffic Operation at Reduced Visibility

At reduced visibility, traffic started to break down at a flow rate of about 2000 vph. This is intuitively reasonable because drivers tend to keep a longer headway (and lower speed) in reduced visibility, which resulted in capacity reduction. When the demand is very high, the speed-flow diagram shows a typical congested traffic pattern (the lower part in Figure 6). This implies that congestion on bridge would happen when demand exceeded the reduced capacity in heavy weather, which implies that traffic control such as speed and demand restrictions may be necessary in certain circumstances in order to maintain efficient operation.

The initial results suggested that microscopic simulation developed for investigation of sea-crossing bridge operation was able to reproduce scenarios realistically, and can be used to explore different potential scenarios of bridge traffic operations. In the next stage, the model will be calibrated to traffic behaviour under controlled bridge and be used to identify effective traffic control and management measures for bridge traffic operation.

ACKNOWLEDGEMENT

The research was supported in part by Natural Science Foundation of Ningbo under grant 2009A610118.

REFERENCES

1. Allen, R. W. and R.E Magdaleno. C. Serafin. S. Eckert. T. Sieja. 1997. 'Driver Car Following Behaviour Under Test Track and Open Road Driving Condition.' SAE Paper 970170, SAE.
2. Brackstone, M. and M. McDonald. 2000. "Car Following: A Historical Review." *Transportation Research (F)*, Vol. 2(4), 181-196.
3. Buckley, J.J. 1993. 'Sugeno Type Controllers are Universal Controllers.' *Fuzzy Sets and Systems*, No. 53, 299-304.
4. Chandler, F.E. and R. Herman. E.W. Montroll. 1958. 'Traffic Dynamics: Studies in Car Following.' *Operations Research*, Vol. 6, 165-184.
5. Gazis, D.C. and R. Herman. R.B. Potts. 1959. 'Car-following Theory of Steady-State Traffic Flow.' *Operations Research*, Vol. 7, 499-505.
6. Gazis, D.C. and R. Herman. R.W. Rothery. 1961. 'Non-linear Follow the Leading Models of Traffic Flow.' *Operations Research*, Vol. 9(4), 545-67.
7. Gipps, P. G., 1981. 'A Behavioural Car-following Model for Computer Simulation.' *Transportation Research (B)*, Vol. 15, 105-111.
8. Greenshields, B.D., A study of highway capacity. *Proceedings Highway Research Record*, Washington, Vol. 14, pp. 448-477, 1935.
9. Kikuchi, S. and P. Chakroborty. 1992. "Car-following Model Based on Fuzzy Inference System." *Transportation Research Record*, No.1365, 82-91.
10. Kikuchi, S. 1996. 'Fuzzy Inference-based Driver Decision Processes and Traffic Flow Simulation.' ITS-IDEA Program Project Final Report, Transportation Research Board, USA, pp 27.
11. Leutzbach, W. and R. Wiedemann. 1986. 'Development and Applications of Traffic Simulation Models at the Karlsruhe Institute fur Verkehrwesen.' *Traffic Engineering and Control*, Vol. 27(5), 270-278.
12. May, A.D.; Keller, H.M.: Evaluation of single and two regime traffic flow models. *Proceedings of the 3rd International Symposium on Transportation and Traffic Theory*, Karlsruhe, 1968
13. McRuer, D. T. and E.S. Krendel. 1959. "The Human Operator as a Servo System Element." *Journal of the Franklin Institute*, Vol. 267, 381-403.
14. McDonald, M. and J. Wu. M. Brackstone. 1997. "Development of a Fuzzy Logic Based Microscopic Motorway Simulation Model." *Proceedings of the IEEE Conference on Intelligent Transport Systems.* Boston, USA.
15. Michaels, R. M. 1965. 'Perceptual Factors in Car-following.' *Proceedings of the Second International Symposium on the Theory of Road Traffic Flow*, 44-59.
16. Wu, J. and M. Brackstone. M. McDonald. 2000. 'Fuzzy Sets and Systems for a Motorway Microscopic Simulation Model.' *Fuzzy Sets and Systems*, Vol. 116, 65-76.
17. Wu, J. and M. Brackstone. M. McDonald. 2003. 'The Validation of a Microscopic Simulation Model: a Methodological Case Study.' *Transportation Research Part C*, Vol. 11(6), 463-479.
18. Zheng, P. A Microscopic Simulation Model of Merging Operation at Motorway On-ramps. PhD Thesis, University of Southampton. 2003.
19. Zheng P., McDonald M., 2005. "Application of Fuzzy Systems in the Car-Following Behaviour Analysis." *Lecture Notes in Computer Science*, 3613, 782-791

A DETERMINISTIC ANNEALING ALGORITHM FOR SIMULTANEOUS ROUTING OF LOADED AND EMPTY CONTAINERS

Kris Braekers
Gerrit K. Janssens
An Caris
Transportation Research Institute (IMOB) – Hasselt University
Universiteit Hasselt – campus Diepenbeek
Wetenschapspark gebouw 5, 3590 Diepenbeek, Belgium
E-mail: {kris.braekers,gerrit.janssens,an.caris}@uhasselt.be

ABSTRACT

Empty container management is an important issue in the hinterland of a port. An empty container allocation model is used to determine the optimal repositioning of empty containers based on demand and supply. Unfortunately, such an allocation model does not take into account vehicle routing decisions. Vehicle routes are then created by a routing model. Loaded container requests are often routed separately, ignoring cost saving opportunities.
In this paper, the simultaneous routing of loaded and empty containers is studied. First, the empty container allocation problem is modeled as a transportation problem to define empty container transport requests. Next, loaded and empty container transportation requests are routed simultaneously. The problem may be formulated as a full truckload pickup and delivery problem with time windows. An insertion heuristic is used to determine a feasible starting solution. Finally, several local search operators are embedded in a deterministic annealing algorithm to improve this solution. It is shown that good quality solutions are found in a small amount of run time.

1. INTRODUCTION

In the hinterland of a major seaport, empty containers are transported between consignees, shippers, inland container depots, intermodal terminals and the port. Although transporting empty containers is a non-revenue generating and costly operation, it is an integral part of an efficient container transportation system (Boile et al. 2008). An empty container allocation model is used to determine the optimal repositioning of empty containers based on demand and supply. Unfortunately, such an allocation model does not take into account vehicle routing decisions (Dejax and Crainic 1987). The resulting empty container allocations serve as input for a separate routing model.
Loaded container transports take place on the same network. In practice vehicle routes are often created separately for loaded and empty containers, ignoring cost saving opportunities. (Huth and Mattfeld 2009) Therefore, two opportunities for reducing container transportation costs may be identified: routing loaded and empty container movements simultaneously and integrating empty container allocation with routing decisions, which means not deciding on the origin/destination of an empty container demanded/supplied until determining vehicle routes.
The objective of this research is to solve a problem integrating empty container allocation decisions with loaded and empty container vehicle routing, thus incorporating both cost saving opportunities. A formulation of this problem is given in section two.
Because of the complexity of the problem, the research is divided into two parts. In the first part, described in this paper, a solution approach for routing loaded and empty containers simultaneously is developed. Empty container transport requests are hereby identified beforehand by an allocation problem. The resulting routing problem, which is a full truckload pickup and delivery problem with time windows, is solved by a deterministic annealing algorithm. (Dueck and Scheuer 1990). The second part of the research, integrating empty container allocation and vehicle routing decisions will be addressed in the future.
An overview of related literature is given in section three. The solution approach is discussed in section four. Finally, experimental results are shown in section five and conclusions are drawn in section six.

2. PROBLEM FORMULATION

The formulation of the overall problem, integrating empty container allocation and vehicle routing decisions, is presented in this section. The objective is to first minimize fleet size and then transportation costs. A single vehicle depot and homogeneous vehicles with a single container capacity are assumed. Several container depots or terminals may exist. Based on a set of pickup and delivery locations and their closest container depot, a set of loaded container transport requests is determined. Each request is defined by a pickup and delivery location, of which one is always a container depot. Service for loaded container pickup and deliveries has to take place between a start and end time window. A set of empty container demand and supply locations is also determined. Empty container demands must be fulfilled between the start of the day and a due time, while empty container supplies are available from a certain point in time until the end of the day. Container depots may act as demand and supply points for empty containers. Therefore, for each possible empty container allocation from a container depot to a demand location and from a supply location to a container depot, a separate node is created. For simplicity, travel times include service times in the formulation. The following notation is used:

LR: set of r loaded container requests
S: set of s empty container supply locations
D: set of d empty container demand locations
CD: set of c container depots/terminals

VD: vehicle depot (index 0)
CD_e: set of $c \times (s+d)$ origins and destinations for empty containers
$N = VD \cup LR \cup S \cup D \cup CD_e$: set of all nodes
$V =$ number of vehicles used
$FC =$ fixed cost per vehicle used
$T_{\max}$ = maximum tour duration

$$c_i = \begin{cases} \text{cost of performing request } i \ (i \in LR) \\ 0 \ (i \notin LR) \end{cases}$$

$$t_i = \begin{cases} \text{time to perform request } i \ (i \in LR) \\ 0 \ (i \notin LR) \end{cases}$$

$c_{ij} = c_i +$ cost from i to j
$t_{ij} = t_i +$ time from i to j
$T_i =$ time epoch in i

$$[a_i, b_i] = \begin{cases} \text{opening times of depot } (i \in CD_e \cup VD) \\ \text{time window before service at } i \ (i \in LR \cup S) \\ \text{time window after service at } i \ (i \in D) \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if a vehicle travels from } i \text{ to } j \\ 0 & \text{else} \end{cases}$$

The problem is formulated as follows:

$$Min\ V \times FC + \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \quad (1)$$

Subject to:

$$\sum_{j \in N} x_{ij} = 1 \qquad \forall i \in LR \quad (2)$$

$$\sum_{j \in N} x_{ji} = 1 \qquad \forall i \in LR \quad (3)$$

$$\sum_{j \in N} x_{0j} = V \quad (4)$$

$$\sum_{j \in N} x_{j0} = V \quad (5)$$

$$\sum_{j \in N} x_{ij} \leq 1 \qquad \forall i \in CD_e \quad (6)$$

$$\sum_{j \in N} x_{ji} \leq 1 \qquad \forall i \in CD_e \quad (7)$$

$$\sum_{j \in N} x_{ij} = \sum_{j \in N} x_{ji} \qquad \forall i \in CD_e \quad (8)$$

$$\sum_{j \in N \setminus D} x_{ij} = 1 \qquad \forall i \in D \quad (9)$$

$$\sum_{j \in S \cup CD_e} x_{ji} = 1 \qquad \forall i \in D \quad (10)$$

$$\sum_{j \in D \cup CD_e} x_{ij} = 1 \qquad \forall i \in S \quad (11)$$

$$\sum_{j \in N \setminus S} x_{ji} = 1 \qquad \forall i \in S \quad (12)$$

$$T_i + t_{ij} \leq T_j + (1 - x_{ij})M \qquad \forall i \in N, \forall j \in N \setminus VD \quad (13)$$

$$T_i + t_{i0} \leq T_{max} + (1 - x_{i0})M \qquad \forall i \in N \quad (14)$$

$$a_i \leq T_i \leq b_i \qquad \forall i \in N \quad (15)$$

$$T_0 = 0 \quad (16)$$

$$T_i \geq 0 \qquad \forall i \in N \quad (17)$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in N, \forall j \in N \quad (18)$$

The objective is to minimize both the number of vehicles used and transportation costs (1). By assigning a large fixed cost per vehicle used, minimizing the number of vehicles used is prioritized. Constraints (2) and (3) ensure that each loaded container request is performed exactly once. Constraints (4) and (5) ensure that each vehicle used leaves and enters the vehicle depot. Since an empty container might be transported directly between a supply and a demand location, each possible empty container allocation involving a container depot can be chosen at maximum once (6,7). When a vehicle enters a container depot, it should also leave it, is formulated in constraint (8). Constraints (9), (10), (11) and (12) ensure that each demand and each supply location is visited exactly once. A demand location can only be reached from a supply location or a container depot. When leaving a demand location, a vehicle moves to any location but another demand location. A supply location can be reached from any node but a supply location. A vehicle can move only to a demand location or a container depot when leaving a supply location. Constraint (13) makes sure that when a vehicle reaches a node at a certain time, it cannot reach the following node before this time augmented with the travel time between the nodes. The objective of this constraint is to prevent loops in the tours and to keep track of tour duration. A maximum tour duration is imposed by constraint (14). This maximum tour duration may represent a maximum working shift duration for the drivers. Constraint (15) involves the time window constraints and constraint (16) sets the starting time of each vehicle at the vehicle depot to zero. Finally, constraints (17) and (18) make sure that the timing variables only take on positive values and the linking variables only take on binary values.

3. LITERATURE REVIEW

In the first part of this section, an overview of literature related to full truckload vehicle routing is given. Next, some approaches to integrate allocation and routing decisions are discussed. Finally, literature concerning deterministic annealing is described.

3.1 Full Truckload Vehicle Routing

Gronalt et al. (2003) develop four savings based heuristics for a full truckload pickup and delivery problem with time windows (FT-PDPTW). Goods are transported between distribution centers or depots. Vehicles are based at different depots and may perform several routes during the planning period. A full truckload pickup and delivery problem (FT-PDP) in the context of an intermodal terminal was studied by Imai et al. (2007). The problem is solved by a heuristic based on Lagrangian relaxation. Caris and Janssens (2010) look at a similar problem but include time window constraints. A deterministic annealing algorithm is implemented to solve the problem. Namboothiri and Erera (2008) study the effect of the introduction of an appointment-based access control system at a port on full truckload drayage operations.
Wang and Regan (2002) and Jula et al. (2005) show that a FT-PDPTW may be transformed to an asymmetric multiple vehicle traveling salesmen problem with time windows (*am*-TSPTW) by collapsing each request into a single node.

Wang and Regan (2002) use a time window discretization scheme to solve the problem. The authors iteratively solve an under- and over-constrained version of the problem. Jula et al. (2005) present an insertion heuristic for the *am*-TSPTW. A lower bound on the number of vehicles for the *(a)m*-TSPTW is presented by Mitrovic-Minic and Krishnamurti (2006). Finally, Zhang et al. (2009) use a reactive tabu search algorithm to solve an *am*-TSPTW with multiple vehicle depots.

3.2 Integrating Allocation And Routing Decisions

Dejax and Crainic (1987) suggest that the independent consideration of container allocation and routing neglects possible synergies arising from an integrated view on these problems. However, attempts to integrate allocation and routing decisions for empty containers are scarce.
Smilowitz (2006) formulates a multi-resource routing problem (MRRP) with flexible tasks to integrate allocation and routing decisions for empty trailers in drayage operations. The problem is solved by column generation. Deidda et al. (2008) integrate allocation and routing decisions for empty containers in the hinterland of a port. A truck capacity of two containers is assumed but time windows and possible synergies with the routing of loaded containers are not taken into account. The authors propose an exact algorithm. Finally, the integration of allocation and routing decisions for transporting swap containers between hubs is studied by Huth and Mattfeld (2009). Again, trucks can carry two containers at a time but no time windows are taken into account. A large neighborhood search is used to solve the problem.

3.3 Deterministic Annealing

With deterministic annealing (also known as threshold accepting), a neighboring solution worsening the objective function is accepted if the worsening is smaller than a certain threshold. This threshold may be adapted during the search. (Dueck and Scheuer 1990). Deterministic annealing has been proven more effective than its stochastic counterpart simulated annealing for several problems. Recently, deterministic annealing has been successfully implemented for a number of vehicle routing problems (Bräysy et al. 2003; Bräysy et al. 2008; Caris and Janssens 2010; Nikolakopoulos and Sarimveis 2007; Tarantilis et al. 2004).

4. SOLUTION APPROACH

As mentioned in the introduction, the problem formulated in the second section is very complex. Therefore, in this paper, the problem is simplified by determining empty container transport requests beforehand. This is done by means of an allocation problem which is modeled as a transportation problem. This transportation problem is then solved optimally by the well-known Ford-Fulkerson algorithm (Ford and Fulkerson 1956).
Once the empty container transport requests are known, the routing problem becomes a FT-PDPTW which may be transformed to an *am*-TSPTW. The problem is solved by a deterministic annealing algorithm.

In section 4.1, the insertion heuristic used to find a feasible starting solution for the algorithm is discussed. The local search operators used are described in section 4.2 and the implementation of the deterministic annealing algorithm is described in section 4.3.

4.1 Insertion Heuristic

An initial solution for the routing problem is obtained by a parallel insertion heuristic similar to the one proposed by Jula et al. (2005). Starting from a number of vehicles equal to a lower bound, a node is randomly selected and inserted in its best possible position. This operation is repeated until all nodes are inserted. If no insertion is possible, an additional vehicle is added. The procedure is iterated a number of times to obtain a good starting solution.

4.2 Local Search Operators

Several local search operators are used to improve the initial solution. Three basic operators try to reduce costs: relocate, exchange and 2-Opt*. With the *relocate* operator, a node is removed from its route and reinserted in another route or in another place in its original route. The *exchange* operator swaps a node from two routes. The insertion position of one node is the same as the removal position of the other node or one place before or one place after this position. The *2-Opt** operator removes an arc from two routes and recombines the resulting parts, that is: the first part of route one with the second part of route two and vice versa. For each operator, one route is selected randomly and all combinations of nodes or arcs of this route with those of the other routes are considered. A first improvement solution strategy is used.
Finally, two operators try to reduce the number of routes by reinserting all nodes of respectively one or several routes into the other routes. The first operator tries to insert all nodes of a randomly selected route into the other routes. The second operator tries to insert all nodes of the n shortest routes into all other routes and $n-1$ empty routes. The parameter n is defined as a percentage of the number of routes in the solution. These operators are inspired by the route elimination operators of Bräysy et al. (2008) and Ho and Haugland (2004).

4.3 Deterministic Annealing Algorithm

The implementation of the deterministic annealing algorithm is based on the one in Bräysy et al. (2008) and Caris and Janssens (2010). It is shown in Algorithm 1. Initially, the current best solution is set to the best solution found by the insertion heuristic. The local search operators are used to improve the initial solution. Only changes improving the current best solution are accepted until a local minimum is reached. Next, the threshold T is set to its maximum value T_{max} and the deterministic annealing algorithm is iterated n times. At each iteration, all local search operators are used. Solutions with a higher number of routes than the best solution are never accepted, while all new solutions with a cost $C(S')$ lower than the cost of the current solution $C(S)$ plus the threshold T are accepted. If a solution better than the current best solution is found, this solution is set as best solution. Otherwise, the threshold T is

reduced by the threshold reduction parameter ΔT. Whenever T reaches zero, it is reset to $r \cdot T_{max}$, with r a random number between zero and one. When T reaches zero and no improvement has been found for n_{DA} iterations, the search is restarted from the current best solution.

Algorithm 1: Deterministic Annealing Algorithm

```
Set S_best to the best solution of the insertion heuristic
Set i_last = 0
while i_last < n_LS do
    Apply local search operators
    if C(S') < C(S_best) then
        Set S_best = S' and i_last = 0
    else
        i_last = i_last + 1
    end if
end while
Set T = T_max and i_last = 0
for i = 1 to n do
    Apply local search operators
    if C(S') < C(S_best) then
        Set S_best = S' and i_last = 0
    else
        i_last = i_last + 1
        if T ≤ 0 and i_last ≥ n_DA then
            S' = S_best, i_last = 0,  T = r · T_max
        else if T ≤ 0 then
            Set T = r · T_max
        else
            Set T = T - ΔT
        end if
    end if
end for
```

5. EXPERIMENTAL RESULTS

Results of experiments using the deterministic annealing algorithm described in the previous section are presented in this section. First, the generation of random instances and the calculation of their lower bounds is discussed. Next, experimental results are shown.

5.1 Random Instances

In order to test the robustness of the algorithm, a 2^4 factorial design is set up. Four problem characteristics are identified. For each characteristic a high (+) and low (-) value is determined. The *time window width* for loaded containers (F1) is a random number between 60 and 120 minutes (-) or between 120 and 240 minutes (+). The *number of container depots* (F2) is one (-) or three (+). For each instance the *number of nodes* (F3) is 100 (-) or 200 (+). These nodes exist of an even amount of loaded container delivery locations, loaded container pickup locations, empty container supply locations and empty container demand locations. Finally, the *coordinates of all nodes* (F4) are randomly chosen between 0 and 25 kilometers (-) or between 0 and 50 kilometers (+). This results in 16 problem classes as shown in table 1.
It is assumed that a single vehicle depot is located in the center of the square region and the location of the container depots is the same for each instance. Maximum tour duration is set at 480 minutes. Service time for loading or unloading a container is 10 minutes and vehicle and container depots are open during the whole planning period.

Table 1: Overview of Problem Classes

Class	F1	F2	F3	F4
1	-	-	-	-
2	+	-	-	-
3	-	+	-	-
4	+	+	-	-
5	-	-	+	-
6	+	-	+	-
7	-	+	+	-
8	+	+	+	-
9	-	-	-	+
10	+	-	-	+
11	-	+	-	+
12	+	+	-	+
13	-	-	+	+
14	+	-	+	+
15	-	+	+	+
16	+	+	+	+

5.2 Lower Bound

A lower bound on the number of vehicles is calculated in similar way as the lower bound of Smilowitz (2006). For each problem, the time needed to perform all requests plus the minimum time needed for linking each node with another node, is calculated. The total time needed is then divided by the maximum tour duration to find a lower bound on the number of vehicles. A lower bound on the cost is found by relaxing the binary restrictions of the linking variables.

5.3 Results

The algorithm is tested for one instance of each of the 16 problem classes. The parameter values used are shown in table 2. In a preprocessing step, each loaded container pickup and delivery location is linked to its closest container depot. The transportation problem is solved to find the optimal distribution of empty containers among empty container demand and supply locations and the container depots. This results in a set of loaded and empty container transport requests which serve as input for the routing problem.

Table 2: Overview of Parameters

Parameter	Value
n_{LS}	5 x # vehicles
n	50000
n_{DA}	10 x # vehicles
T_{max}	0.0025 x $C(S)$
ΔT	0.00001 x $C(S)$

The algorithm runs around 7 seconds for 100-node problems and around 15 seconds for 200-node problems. An overview of results is presented in table 3. The costs (C) represent only variable costs related to distance travelled, not including fixed vehicle costs. Instead, the number of

vehicles (V) are presented separately. For each problem, columns two and three show the lower bound on the number of vehicles and the costs. Columns four and five show the average results over five runs of the algorithm. The absolute gap for the number of vehicles is shown in column six, while the relative gap for the costs is shown in column seven.

Table 3: Overview of Results

Class	LB		Average		Gap	
	V	C	V	C	V	C (%)
1	6	1015	7	1034	1	1.86
2	6	978	6	1010	0	3.28
3	6	796	6	837	0	5.17
4	5	657	6	712	1	8.32
5	10	1703	13	1783	3	4.70
6	11	1594	12	1649	1	3.46
7	10	1303	11	1422	1	9.12
8	10	1270	11	1355	1	6.70
9	8	1892	9	1938	1	2.46
10	8	1891	9	1942	1	2.68
11	7	1557	8	1609	1	3.38
12	7	1420	8	1498	1	5.47
13	14	3252	17	3374	3	3.74
14	14	3240	16	3345	2	3.24
15	13	2717	15	2911	2	7.13
16	13	2643	14.6	2798	1.6	5.89

Table 3 shows that the deterministic annealing algorithm is able to find good quality solutions in a short amount of time, especially when taking into account the fact that the linear relaxation of the problem generally provides a rather weak lower bound on the costs.

6. CONCLUSIONS

Two cost saving opportunities in regional container transportation are identified: routing loaded and empty containers simultaneously and integrating allocation and routing decisions. A formulation for a problem integrating both opportunities is presented in section two.

The remainder of the paper focuses on developing a solution strategy for implementing the first cost saving opportunity. A deterministic annealing algorithm for routing loaded and empty containers simultaneously was described. It is shown that this algorithm is able to find good quality solutions relatively fast. Therefore this algorithm may be used as a starting point for integrating allocation and routing decisions.

In future research, adaptations to the deterministic annealing algorithm will be implemented to change the empty container allocations which are currently determined by the transportation problem. Furthermore, a tighter lower bound needs to be determined to offer a better insight in the quality of the algorithm.

REFERENCES

Boile, M.; S. Theofanis; A. Baveja; and N. Mittal. 2008. "Regional Repositioning of Empty Containers: A Case for Inland Depots." *Transportation Research Record: Journal of the Transportation Research Board* 2066, No.1, 31-40.

Bräysy, O.; J. Berger; M. Barkaoui; and W. Dullaert. 2003. "A Threshold Accepting Metaheuristic for the Vehicle Routing Problem with Time Windows." *Central European Journal of Operations Research* 11, No.4, 369-387.

Bräysy, O.; W. Dullaert; G. Hasle; D. Mester; and M. Gendreau. 2008. "An Effective Multistart Deterministic Annealing Metaheuristic for the Fleet Size and Mix Vehicle Routing Problem with Time Windows." *Transportation Science* 42, No.3, 371-386.

Caris, A. and G.K. Janssens. 2010. "A deterministic annealing algorithm for the pre- and end-haulage of intermodal container terminals." *International Journal of Computer Aided Engineering and Technology* 2, No.4, 340-355.

Crainic, T.G.; M. Gendreau; and P.J. Dejax. 1993. "Dynamic and Stochastic Models for the Allocation of Empty Containers." *Operations Research* 41, No.1, 102-126.

Deidda, L.; M. Di Francesco; A. Olivo; and P. Zuddas. 2008. "Implementing the street-turn strategy by an optimization model." *Maritime Policy & Management* 35, No.5, 503-516.

Dejax, P.J. and T.G. Gendreau. 1987. "A Review of Empty Flows and Fleet Management Models in Freight Transportation." *Transportation Science* 21, No.4, 227-248.

Ford, L.R. and D.R. Fulkerson. "Solving the Transportation Problem." *Management Science* 3, No.1, 24-32.

Gronalt, M.; R.F. Hartl; and M. Reimann. 2003. "New savings based algorithms for time constrained pickup and delivery of full truckloads." *European Journal of Operational Research* 151, No.3, 520-535.

Ho, S.C. and D. Haugland. 2004. "A tabu search heuristic for the vehicle routing problem with time windows and split deliveries." *Computers & Operations Research* 31, No.12, 1947-1964.

Huth, T. and D.C. Mattfeld. 2009. "Integration of vehicle routing and resource allocation in a dynamic logistics network." *Transportation Research Part C* 17, No.2, 149-162.

Imai, A.; E. Nishimura; and J. Current. 2007. "A Lagrangian relaxation-based heuristic for the vehicle routing with full container load." *European Journal of Operational Research* 176, No.1, 87-105.

Jula, H.; M. Dessouky; P. Ioannou; and A. Chassiakos. 2005. "Container movement by trucks in metropolitan networks: modeling and optimization." *Transportation Research Part E* 41, No.3, 235-259.

Mitrovic-Minic, S. and R. Krishnamurti. 2006. "The multiple TSP with time windows: vehicle bounds based on precedence graphs." *Operations Research Letters* 34, No.1, 111-120.

Namboothiri, R. and A.L. Erera. 2008. "Planning local container drayage operations given a port access appointment system." *Transportation Research Part E* 44, No.2, 185-202.

Nikolakopoulos, A. and H. Sarimveis. 2007. "A threshold accepting heuristic with intense local search for the solution of special instances of the traveling salesman problem." *European Journal of Operational Research* 177, No.3, 1911-1929.

Smilowitz, K. 2006. "Multi-resource routing with flexible tasks: an application in drayage operations." *IIE Transactions* 38, No. 7, 555-568.

Tarantilis, C.D.; C.T. Kiranoudis; and V.S. Vassiliadis. 2004. "A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem." *European Journal of Operational Research* 152, No.1, 148-158.

Wang, X. and A.C. Regan. 2002. "Local truckload pickup and delivery with hard time window constraints." *Transportation Research Part B* 36, No.2, 97-112.

Zhang, R.; W.Y. Yun; and I. Moon. 2009. "A reactive tabu search algorithm for the multi-depot container truck transportation problem." *Transportation Research Part E* 45, No.6, 904-91.

INVENTORY LOGISTICS AND SUPPLY CHAIN SIMULATION

REQUIREMENTS FOR ELECTRONIC TAGS FROM A BUSINESS PERSPECTIVE

Izabella Lokshina
SUNY Oneonta
324B Netzer Administration Bldg.
Oneonta, NY 13820, USA
E-mail: lokshiiv@oneonta.edu

Evan Schiele
SAP America, Inc
89 Headquarters Plz,
Morristown, NJ 07960, USA
E-mail: evan_schiele@hotmail.com

KEYWORDS
RFID Technology Integrated in Business Processes, RFID Infrastructure, RTLC Model, Risk Analysis, Design Requirements.

ABSTRACT

The importance that businesses and governments have placed on RFID technology in electronic article surveillance; cashless payment from credit cards and public transportation tickets to consumer devices; healthcare and e-government involves reviewing the requirements for electronic tags. This paper defines the RFID Tag Life Cycle (RTLC) model that follows the product life cycle; considers representative business processes and interacting entities; classifies the tag usage methods; identifies the risks; and introduces the design requirements for electronic tags from a business perspective.

INTRODUCTION

Business enterprises and even governments have looked upon the emerging Radio Frequency Identification (RFID) technology as a possible silver bullet in the application of pervasive and ubiquitous computing (Roberts 2006; Smith and Konsynki 2003), along with a further development called the Internet of Things. RFID is a wireless Automatic Identification and Data Capture (AIDC) technology that operates without human intervention. Manufacturers and distributors look to RFID as a savior in handling the impending logistical overload (Furness 2005; Ranky 2006; Wyld 2006).

In fact, the retail industry (with such major retailers like Walmart and Tesco in the US, and Britain's Mark & Spencer and Germany's Metro Group) is the initial driver of RFID adoption. Recently with the push by Wal-Mart and the US Department of Defense to use RFID tags in their supply chain, RFID has re-gained the limelight. Recently RFID market observers have predicted a market growth of 20.7% from 2008 to 2010 (according to reports published by http://www.researchandmarkets.com).

Today RFID technology is used in many applications: manufacturing, healthcare, transport and logistics, defense and other industries. Soon RFID technology will be common for tracking items, products, animals, shipments and even humans.

Associated with the integration of RFID technology in business and in the day-to-day life of consumers are various security and privacy concerns (Weis et al. 2003; Henrici and Muller 2004; Garfinkel et al. 2005; Juels 2006). These concerns are caused by the following factors:

- the potential to reduce purchase anonymity;
- embedded tags in products that can reveal sensitive information;
- violations of location privacy with no-contact, non-line-of-sight through non-conducting material such as cardboard or paper;
- fast read rate at a distance of up to a few meters and invisible identification that can be done indiscriminately.

The importance that businesses and governments have placed on RFID technology in electronic article surveillance, cashless payment from credit cards (Smart cards) and public transportation tickets (Mifare) to consumer devices (such as EZpass and Speedpass), healthcare (tracking items, drugs and patients) and e-government (such as e-driver's license and ePassport) involves reviewing the requirements for electronic tags.

There have been some published papers on integration RFID technology in business, approaching the subject from a position of efficient usage of electronic tags as they were delivered to the RFID market. This paper will take a different approach with introducing the design requirements for electronic tags from a business perspective to be considered by the RFID developers.

ELECTRONIC TAGS AS COMPONENTS OF RFID SYSTEM

A basic RFID infrastructure consists of 3 major components: tags; a reader and its antenna; and middleware and application software together with networking software since the applications may not be collocated with the reader. RFID tags are transponders that contain a chip with active and passive components, more and more integrated into a single module, and an antenna. The antenna receives radio signals from the readers while the chip is used to respond to the signals and to store information. Current tags are small enough to be embedded into physical objects for identification and tracking. Tags can be read only, write once/read many times or read/write capable (Asif and Mandviwalla 2005).

RFID readers are electronic interrogators that emit and receive radio signals through their antennas. They capture data stored and may overwrite the data on the tags. They are also responsible for the information flow between the tags and the host system via the middleware (Castro and Fosso Wamba 2007).

There are 2 main types of RFID tags – active and passive tags. The active tags are used for powered tags and for tags that can respond with more that a predefined ID. The active tag requires a power source which makes it more costly and larger. It is not practical in the retail industry. It operates at 455 MHz, 2.45 GHz or 5.8 GHz and has read ranges from 20 to 100 meters. The passive tag, on the other hand, is cheap, small and has an infinite shelf life. These qualities make it practical for the retail industry. It operates at 128 KHz, 13.6 MHz, 915 MHz or 2.45 GHz and has read ranges from a few inches to 30 feet. There are 2 fundamental design approaches used in passive tags – the near field which uses the magnetic induction and far field which uses electromagnetic wave capture (Want 2006; Weinstein 2005).

The passive Ultra High Frequency (UHF) RFID technology operating from 860 MHz to 960 MHz offers the best read range read rate performance, readability through a wide range of materials, and at a very cost-effective manner; as a result, it is the most prevalent technology across all industries. RFID devices are classified into categories, for example Class 1 Generation 1 devices do not have security features while the Class 1 Generation 2 devices have security features like authentication and encryption. For a more complete classification, see (Phillips et al. 2005).

VALUE AND RISKS OF RFID TECHNOLOGY INTEGRATED IN BUSINESS PROCESSES

RFID tags have advantages over bar codes in that:

- It does not require line-of-sight access.
- The read range is larger.
- Tag readers can communicate with multiple tags simultaneously.
- The tags can store information.
- The tags can be read through non-conducting materials such as cardboard and paper.
- The tags can be read at a much faster rate.

As a result of the benefits of RFID, it is currently used and will be used in more applications in the future. RFID technology is now being embedded in various business processes such as operations management where it is used in asset management, tracking components, work-in-progress and manufactured products; logistics where items are tracked from the inventory or store through the shipment to the retailer; customer service where items are traced to customers to validate defective items that are returned or that are for repairs. RFID information system typically consists of the tag identification (with verification) and wireless flow of data from the tag to the reader through application middleware (SAVANT) to the enterprise information system and may be transmitted via inter-enterprise systems to an external information system.

The popularity of RFID technology has given rise to a number of serious security and privacy concerns. Associated with these concerns, Garfinkel proposed a consumer's bill of rights published through http://www.technologyreview.com in October 2002 .

The RFID Bill of Rights addresses the most obvious RFID abuses. Users of RFID systems and purchasers of products containing RFID tags have:

- The right to know if a product contains an RFID tag.
- The right to have embedded RFID tags removed, deactivated, or destroyed when a product is purchased.
- The right to first class RFID alternatives. Consumers should not lose other rights (e.g. the right to return the product or to travel on a particular road), if they decide to opt-out of RFID or exercise an RFID tag's "kill" feature.
- The right to know what information is stored inside their RFID tags. If this information is incorrect, there must be a means to correct or amend it.
- The right to know when, where and why an RFID tag is being read.

The subsequent EPCglobal guidelines for the use of RFID (http://www.epcglobalinc.org/public_policy/public_policy_guidelines.html) were developed to lessen some of these concerns but did not go far enough.

DESIGN REQUIREMENTS FOR ELECTRONIC TAGS FROM A BUSINESS PERSPECTIVE

This paper defines the RFID tag life cycle model that follows the product life cycle; considers the representative business processes and interacting entities; classifies the tag usage methods; identifies the risks; and introduces the design requirements for electronic tags from a business perspective to be considered by the RFID developers.

The RFID infrastructure is obtained from the EPC Network standard (http://www.epcglobalinc.org/home). Reference (Fosso Wamba et al. 2007) succinctly describes it as follows: "It is composed of five components: (i) the EPC code, which starts as a 64-bit to 128-bit identifier. Once it is incorporated into an RFID chip (also called an EPC tag) and attached to a physical object, product or item, it can provide information such as the manufacturer, the product category and size, the date when the product was made, the expiration date, the final destination, etc. (ii) The RFID reader identifies any EPC tag within its interrogating field, reads the EPC tag and forwards information to the SAVANT. (iii) The SAVANT is the middleware system located between readers and the application systems (AS). Based on configured business rules, it is responsible for data filtering and aggregation and interacts with the EPC Information Service (EPC-IS) and the local Object Name Service (ONS). (iv) The EPC-IS is the gateway between any requester of information and the firm's AS and internal databases. (v) The local ONS is an authoritative directory of information sources available in order to describe all EPC tags used in a supply chain."

RFID devices are used for tracking and transfer of data at different stages or locations to different "owners" in an

open system for later processing. They are different from the standard resources due to the fact that they are "carriers" of the information resources that are useful either because the information is processed directly or indirectly like in the case of tracking. For this reason, RFID have to support two important features: transfer of ownership and multiple authorizations.

Unlike other devices the tag data can change over time; there are different "owners" and variations in the tag usage depending on the product life cycle, business processes and interacting entities.

Reference (Engberg et al. 2004) presents a regular RFID Tag Life Cycle (RTLC) that follows the product life cycle as consisting of 4 phases:

- Supply Chain Management, where the tag delivers a unique ID called ePC, electronic product code (http://www.epcglobalus.org/dnn_epcus/Standards/EPCglobalStandards/tabid/185/Default.aspx) that is used for tracking items through the final manufacture, inventory, storage, distribution, location and transfer of "ownership" processes.
- In-Store and Point-of-Sales, where the tags may be used for verification and prevent counterfeit, track the product and also monitor consumer interaction with the product and if a credit card is used for purchase, the product can be associated with the consumer for purchase services. (After the purchase, the tag can be removed or "killed" but this will prevent post-purchase services such as access to product support, warranty services, and other services in phases 3 and 4).
- Customer Control and after Sales Services, (if the tag is live) the tag can be used as an enabling technology for intelligent applications (for example, an intelligent refrigerator with a reader can flash a message if the date on tagged food items have expired), after sales service where the ePC can be used for verification.
- Recycling and Waste Management where the ePC can be used to sort recyclable materials and identify manufacturers of items that contain hazardous materials.

The RTLC model applies to most current commercial usage of RFID technology – the production of goods for sale. Even though other RFID applications (like eHealth, ePassports, Mifare cards, Smartcards, EZPass, SpeedPass, etc) tags have a different tag life cycle yet the threats derived from the communication mechanism analysis of the tag life cycle in phases 2 and 3 are also applicable here.

It is important to evaluate the RTLC processes that follow the product life cycle and the business entities that interact with the tag, in order to summarize the tag requirements:

- The tag will be embedded on the product by the manufacturer.
- The distributor (storage and transfer) that transport this product to the retailer will rely on the tag for logistical management.
- The retailer will use the tag to verify the product and prevent counterfeit, and for tracking and managing inventory.
- In store the embedded tag will be used to monitor customer interaction with this product as the product can be associated with the consumer for purchase services at the point-of-sales counter.
- The after sales service provider will use the tag for verification and warranty services, and tracking product history.
- The infrastructure service provider will link the ePC to the manufacturer or retailer databases where detailed information is stored.
- The consumer who buys this product may benefit from intelligent applications enabled by the embedded RFID tag.
- The waste manager will use the tag to sort the waste materials and levy charges to the manufacturer (derived from the ePC) based on the volume and type (hazardous, recyclable or garbage) of materials.

The NIST800-98 guideline classifies RFID risks into 4 categories (Karygiannis et al. 2007):

- Business Process (BP) Risk - where a direct attack on the RFID system components will destabilize the business processes it was designed to enable.
- Business Intelligence (BI) Risk – where a competitor could gain unauthorized access to the RFID-generated information and use it against the company.
- Privacy (P) Risk – where personal or private information in an RFID system is used for purposes against privacy regulations or laws. A person in ownership and/or control of a RFID tag can also risk being tracked.
- Externality (EX) Risk – where the RFID system could pose a threat to non-RFID networks, systems, resources or people. There are two types of risks: Hazards of Electro-magnetic Radiation and Computer Network Attacks.

However, there is no the electronic tag risk analysis neither a threat listing in this document.

Threats to RFID systems are typically related to the unauthorized use of readers and happen at the air interface. Reference (Rotter 2008) discusses the following RFID tag risks:

- Eavesdropping happens when attackers secretly monitor the communication between the tag and the reader. This can be minimized by encrypting the data, shielding the tag or limit the tag-reader distance, however attacker can use a non-standard reader to extend the distance.

- Relay attack is done using a fake tag to communicate with the real reader and a fake reader to communicate with the real tag. The information obtained by the fake reader is passed to the fake tag which then communicates with the real reader. This can be minimized by using short range tags, by shielding the tag or implementing the distance bounding protocol.
- Rogue readers can intercept and read tag information. Reader authentication can be used to allow only authorized readers to read the tag information.

Table 1: Requirements for Electronic Tags from a Business Perspective Introduced with the RTLC Model

Representative Business Processes	Interacting Entities	RFID Tag Risks	Risk Categories	Design Requirements for RFID Tags
Supply Chain Management				
Final Manufacturing Process (tagging) Product Movement/Transfer Inventory Storage Inventory Location	Manufacturer (within company premises)	Eavesdropping	BI	Encrypting the data; Shielding the tag; Limiting the tag-reader distance
		Tag Cloning	BP	Tag authenticating
		Replay Attack	EX	Encrypting the data; Shielding the tag; Limiting the tag-reader distance; Tag authentication
		Rouge Reading	BI	Reader authentication
Product Distribution/Transport Warehouse/Product Storage Warehouse/Product Location	Distributors (outside company premises)	Relay Attack	BP, EX	Using short range tags; Shielding the tag; Implementing the distance bounding protocol
		Tracking	BI	Low range tags; Shielding tags; Authenticating the readers; Disabling the tags
		Jamming and Blocking	BP	Early detection and localization; Disabling the tags
Ownership Transfer	Distributors/ Retailers	Eavesdropping	BI	Encrypting the data; Shielding the tag; Limiting the tag-reader distance
		Tag Cloning	PB	Tag authentication
		Replay Attack	EX	Encrypting the data; Shielding the tag; Limiting the tag-reader distance; Tag authentication
In-Store and Point-of-Sales				
Product Verification Inventory Store Storage Inventory Store Items Location Display Shelf Item Location Consumer-Product Interaction	Retailer	Eavesdropping	BI	Encrypting the data; Shielding the tag; Limiting the tag-reader distance
		Jamming and Blocking	BP	Early detection and localization; Disabling the tags
		Tracking	P	Low range tags; Shielding tags; Authenticating the readers; Disabling the tags
Consumer-Product Purchases	Consumer	Blocking	BP	Early detection and localization; Disabling the tags
		Tracking	P	Low range tags or shielding tags; Authenticating the readers; Disabling the tags
Customer Control and After-Sales Services				
Track Product/Consumer that Enables Intelligent Applications Verification for After-Sales Service	After Sales Service Provider/ Retailer	Tag Destruction	BP	Using protective material
		Tracking	EX	Low range tags; Shielding tags; Authenticating the readers; Disabling the tags
Recycling and Waste Management				
Sort Waste Material Identify Manufacturer	Waste Manager	Tag Destruction	BP	Using protective material

- When attackers make a duplicate tag based on a real tag after gaining unauthorized access to the real tag is called tag cloning. The duplicate tag can be used to gain unauthorized access to confidential information or make an electronic transaction. Tag cloning can be minimized using tag authentication.
- Tags can also be used to track people who have in their ownership and /or control a RFID tag by rogue readers and eavesdropping devices. This can be minimized by using low range tags or shielding tags, authenticating the readers or disabling the tags.
- Replay attack happens when attackers eavesdrop on the communication between a tag and reader, and then use a cloned tag to repeat the authentication sequences. This can be minimized using the same countermeasures for eavesdropping and tag cloning.
- Physical tag destruction is when a tag is physically destroyed like microwaving in an oven or hitting it with a hammer.
- Blocking is when attackers use a blocker tag to stimulate the existence of many tags and cause a denial of service as the reader tries to interrogate these non-existence tags.
- Jamming means paralyzing the communication system by generating radio noise at the same frequency as that of the system. Blocking and jamming devices can be detected early and localize so that appropriate actions is taken.

The RTLC model that follows the product life cycle; the representative business processes related to the tag usage and the interacting entities; the associated electronic tag risks; and the design requirements for electronic tags are presented in Table 1. Besides, we expect that RFID technology will continue to develop and be further integrated in business processes. Therefore, more requirements for electronic tags will be introduced from a business perspective, and this will drive the RFID developers to come up with better tag designs.

CONCLUSIONS

The prominence that businesses and governments have placed on RFID technology in electronic article surveillance, cashless payment from credit cards and public transportation tickets to consumer devices, healthcare and e-government required reviewing the design requirements for electronic tags. This paper defined the RFID tag life cycle model that followed the product life cycle; considered the representative business processes and interacting entities; classified the tag usage methods; identified the risks; and introduced the design requirements for electronic tags from a business perspective to be considered by the RFID developers.

REFERENCES

Asif, A. and M. Mandviwalla. 2005. "Integrating the Supply Chain with RFID: a Technical and Business Analysis." *Communications of the Association for Information Systems*, Vol. 15, 393-427.

Castro, L. and S. Fosso Wamba, 2007, "An Inside Look at RFID Technology." *Journal of Technology Management and Innovation*, Vol. 2, No. 1, 128-141.

Engberg, S., M. Harning and C. Damsgaard Jensen. 2004. "Zero-knowledge Device Authentication: Privacy & Security Enhanced RFID Preserving Business Value and Consumer Convenience." *In Proceedings of the Conference on Privacy, Security and Trust (PST),* New Brunswick, Canada.

Fosso Wamba, S., Lefebvre, L.A. and Lefebvre, E. 2007. "Integrating RFID Technology and EPC Network into a B2B Retail Supply Chain: A Step Toward Intelligent Business Processes." *Journal of Technology Management and Innovation*, Vol. 2(2), 114-124.

Furness, V. 2005. "The RFID Market Outlook: New Applications, Best Practice and Future Profit Opportunities." *Business Insight*. www.globalbusinessinsights.com

Garfinkel, S., A. Juels and R. Pappu. 2005. "RFID Privacy: An Overview of Problems and Proposed Solutions". *IEEE Security and Privacy,* 3(3), 34–43.

Henrici, D. and P. Muller. 2004. "Tackling Security and Privacy Issues in Radio Frequency Identification Devices". *In Proceedings of the International Conference on Security in Pervasive Computing (SPC 2004)*, Vienna, Austria, 219–224.

Juels, A. 2006. "RFID Security and Privacy: A Research Survey". *IEEE Journal on Selected Areas in Communications*, 24(2), 381–394.

Karygiannis, T., B. Eydt, G. Barber, L. Bunn and T. Phillips. 2007. "Guidelines for Securing Radio Frequency Identification (RFID) Systems." National Institute of Standards and Technology Special Publication 800-98.

Ranky, P. 2006. "An Introduction to Radio Frequency Identification (RFID) Methods and Solutions." *Assembly Automation*, 26(1), 28-33.

Roberts, C. M. 2006. "Radio Frequency Identification (RFID)." *Computers & Security*, Vol.25, No.1, 18-26.

Rotter, P.. 2008. "A Framework for Assessing RFID System Security and Privacy Risks." *IEEE Pervasive Computing*, 7(2), 70–77.

Smith, H. Y. and B. Konsynki. 2003. "Developments in Practice X: Radio Frequency Identification (RFID) – an Internet for Physical Objects." *Communications of the Association for Information Systems*, Vol. 12, 301-311.

Weis S, S. Sarma, R. Rivest and D. Engels. 2003. "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems." *In Proceedings of the International Conference on Security in Pervasive Computing (SPC 2003)*, Boppard, Germany, 454–469.

Wyld, D. 2006. "RFID 101: The next big thing in management." *Management Research News*, 29(4), 154-173.

BIOGRAPHY

IZABELLA LOKSHINA, PhD is Associate Professor of MIS and chair of Management, Marketing and Information Systems Department at SUNY Oneonta, USA. Her main research interests are Intelligent Information Systems, Communication Networks, Modeling and Computer Simulation; lokshiiv@oneonta.edu

EVAN SCHIELE, PMP, MBA is a Program Delivery Manager with SAP America. His main research interests are Enterprise Resource Planning, Utilization of Information Technology in Business and Project Management; evan_schiele@hotmail.com

ROBUSTNESS OF THE PERFORMANCE OF AN UNRELIABLE PRODUCTION-INVENTORY SYSTEM AGAINST VARIOUS DEMAND PATTERNS

Roongrat Pisuchpen
Industrial Engineering, Kasetsart University
50 Phahon Yothin, Chatuchak, Bangkok 10900, Thailand
e-mail: fengros@ku.ac.th

Gerrit K. Janssens
Transportation Research Institute (IMOB)
Hasselt University - campus Diepenbeek
Wetenschapspark 5, 3590 Diepenbeek, Belgium
e-mail: gerrit.janssens@uhasselt.be

KEYWORDS

Production-inventory system, unreliable machines, transfer line, performance evaluation.

ABSTRACT

A production-inventory model aims to help the decision-maker in running the production process in relation to the demand process in such a way that shortages are kept at a low level and no excessive inventory is built up. The production system itself might be unreliable, introducing an additional uncertainty factor. Given characteristics of the unreliable behaviour in terms of failure times and repair times, the decision-maker decides to which level of inventory to produce in order to secure the system against all types of uncertainty. In literature, an analytical model of such a system has been introduced assuming a compound Poisson process for the demand process in which the individual order sizes follow an exponential distribution in order to obtain some customer service performance measures. The performance of inventory decisions depends on the shape of the demand distribution in a certain period. In order to evaluate the assumption of an exponential order size against other demand distributions, a discrete event simulation model has been developed in the Arena software. The influence of the shape of the demand distribution is tested using the simulation model.

INTRODUCTION AND LITERATURE REVIEW

Production–inventory models link the production process to the demand process to ensure that shortages are kept at a low level and no excessive inventory is built up. The models vary in terms of the characteristics of the production process, the demand pattern, the customer response to shortages and the inventory review procedures. For example it can deal with single or multiple products, with deterministic or stochastic demand, with a finite or infinite planning horizon, with linear, convex or concave cost functions, with various cost elements, and with various types of constraints like nonnegative inventory, backordering, initial or terminal constraints (Tzafestas, 1982). The trade-off is made either in a cost-oriented approach, in which shortage penalties have to be defined, or in a service-oriented approach in which some predetermined customer service performance measures need to be obtained.

In this paper, a production–inventory system is considered, in which the production system is unreliable: the production facility can fail resulting in disrupted production plans, reduced inventory and shortages. A single production stage is considered: a line lay-out or a network lay-out introduce a higher level of complexity to the model. In order to improve the efficiency of such a system, there are two approaches. One is the utilisation of a stand-by facility, the other is to allocate an inventory between the facility and the market. The stand-by facility is put into operation when the facility fails. The inventory avoids lost sales or backorders. The decision problem relates to the size of the buffer inventory at which production has to be stopped. The decision problem becomes a difficult problem when a series of machines (i.e. a transfer line) is concerned. If no intermediate buffers are provided, failures at any stage result in the failure of the entire production system and consequently the overall production rate is affected. Given characteristics of the unreliable behaviour in terms of failure times and repair times, the decision-maker has to decide to which level of inventory to produce in order to secure the system against all types of uncertainty.

In the literature on multi-stage lines with finite intermediate buffers, De Koster (1989) distinguishes four classes of models. The production system in our model is a continuous flow-line, which belongs to the third class, i.e. the class which deals with continuous flow models and where machine speeds are deterministic but machines may fail.

Continuous flow-line systems operate on a flow of product through the system. Many models on such flow lines consider only one performance measure, called the *availability*, i.e. the production rate of the whole system and do not take into consideration the demand process.

Articles, which model these systems without considering demand have been produced over the last decades, for example, among many others, by Buzacott and Hanifin (1978), Malathronas et al. (1983), Dallery et al. (1989), Van Oudheusden and Janssens (1994), and Sörensen and Janssens (2001).

Our model is based on a study of a one-machine system which operates as long as the accumulated inventory is below a preset threshold. The machine or facility is subject to failures after which it is repaired. Both the times between failures and the repair times are assumed to follow exponential distributions. The demand process is assumed to be a compound Poisson process. Orders arrive according to a Poisson process. The size of an individual order is also of stochastic nature. Such a system with Poisson order arrivals which have order sizes of an exponential distribution has been studied by Posner and Berg (1989). An approximate model for a series of unreliable machines with similar characteristics has been developed by Sörensen and Janssens (2003).

The performance of inventory decisions depend on the shape of the demand distribution in a certain period. The compound Poisson distribution can take various shapes from close to the normal and Gamma distributions to bimodal distributions (which look similar to the bimodal Beta-distributions). In an investigation on the relevance of the demand shape Bartezzaghi, Verganti, and Zotteri (1999) find out that the shape is very relevant. In extreme cases the impact of different demand distribution shapes on inventories is comparable to the effect of doubling the coefficient of variation. An extensive overview of the distribution types used to model period demand, lead time and/or demand during lead time is given in Vernimmen et al. (2008).

The next section introduces an analytical model from literature for the single facility production-inventory system with its assumptions and notations. Also a simulation model is presented and validated for this particular system. Then an experimental design is proposed to test the influence of the shape of the compound Poisson distribution on buffer size decisions. Simulation results are presented and analysed. The final section states the conclusions of this research.

AN ANALYTICAL AND SIMULATION MODEL FOR THE PRODUCTION-INVENTORY SYSTEM

The production-inventory system under study is the one described in Posner and Berg (1989). The system consists of a single production machine with constant production rate which operates as long as the inventory level is below a pre-set parameter value, and is idle otherwise. The system is unreliable in such a way that the machine fails from time to time and has to be repaired: both the operating times of the machine and the repair times of the failed machines follow exponential distributions. Demand depletes the inventory according to a compound Poisson process. A demand that cannot be satisfied completely from the inventory on hand takes whatever amount is available and the rest of the demand is lost. The authors obtain the steady-state distribution of the inventory level as well as some operational performance measures like the customer service level and machine utilization, making use of fhe generalised technique of the 'system point' in level-crossings analysis (Brill and Posner, 1977; Brill and Posner, 1981).

The model uses a number of assumptions regarding the production rate (set equal to 1 without loss of generalization), the operating time until failure (follows an exponential distribution), the repair time (follows an exponential distribution). Demand orders arrive according to a Poisson process and order sizes follow an exponential distribution. An order is filled up from inventory up to the level available. The part of the order which cannot be satisfied is *lost*. Production is stopped when the inventory reaches a pre-specified level. It is resumed when the inventory level drops under this value, it means after the next demand order epoch.

Using these assumptions, a number of performance measures can be calculated. Some examples are: the *availability* of the machine (i.e. fraction of time the machine is engaged in active production), the *fraction of time the machine is down due to failure*, the *machine shutdown rate due to overstocking,* the *fraction of loss due to shortages,* the *fraction of demand satisfied* and the *average inventory level.*

One of the assumptions introduces the exponential distribution for modeling the individual order size. As the performance of an inventory management policy might depend on the shape of the demand distribution, it is appropriate to develop a discrete event simulation model for the investigation of the influence of the shape on the policy parameters. The model, which has been implemented in the Arena-software (Kelton et al., 2009), is briefly discussed in the following paragraphs.

The model is shown in Appendix 1. Variables include the production rate, the uptime rate, the repair rate, the initial inventory, and the value of *M* indicating the level at which production resumes. The *Create* module creates customer orders as entities according to a Poisson process. The *Assign* module defines the order size. The *Separate* module separates each order into two parts of the decision process.

The first decision is to decide whether the on-hand inventory is sufficient to supply the customer order. If the condition is true, the customer receives the full order. The subsequent *Assign* module updates the inventory level. In case of the on-hand inventory is insufficient to supply the customer order in full, the customer receives the order equal to the on-hand inventory. As a result, the inventory level becomes zero. The *Record* module is used to count the number of customers (un)satisfied and the Statistic Spreadsheet module is used to determine the percentage of demand satisfied.

The second decision considers whether manufacturing should be resumed. At any time when the on-hand inventory is less than a level *M*, the production process is triggered. If the condition is true, the order sheet is sent to the *Hold for condition* module and waits for production until there is no work-in-process on the machine. A unit production time of (1/ Production rate) is used.

The *Process* module represents the production process. The machine is unreliable with the failure rate specified in *Failure Spreadsheet* module and *Resource Spreadsheet* module of this machine. The *Failure* module shows the machine uptime with an exponential distribution with mean (1/Uptime rate). The machine downtime follows an exponential distribution with mean (1/Repair rate) minute. Failure appears in the busy state only. When a production unit is produced, the on-hand inventory is incremented using the *Assign* module. Also then the inventory level is compared with the value *M* using the *Separate* module to copy the entity into two separate paths. In the first path, the manufactured unit leaves the system. In the other path, the entity is needed to check the decision logic by comparing the inventory level with the level *M*. The production terminates when the inventory level reaches level *M* using the *Pickup* module to pick the production order from the *Hold* module out of the system.

Before designing experiments for this model, the model needs to be validated. In order to allow the system to reach the steady-state and reduce the impact of the warm-up period, a run length of 30 days is specified. The system operates 24 hours per day and 30 replications are made to reduce the variance of the estimates. With this number of replications, the half width of percent demand satisfied is less than 1 percent.

The validation experiment is run by keeping the order arrival process unchanged (i.e. a Poisson arrival process with arrival rate 0.3 and with an order size which follows an exponential distribution with mean 3.1 units). The value of *M* is also fixed at 100 units. The simulation experiments vary the failure rate and the repair rate versus the order arrival rate. The ratio failure rate/order arrival rate takes values from 0.1 till 1.0 in steps of 0.1. The ratio repair rate/order arrival rate takes values {10,5,1,0.5,0.1}. The combination of both ratios leads to fifty experimental points for validation.

As a measure of performance for validation the percentage difference between the mean value of ten replications and the value obtained from the Posner-Berg formulae for the percentage of demand satisfied is taken. The difference ranges from less than 0.04% till 17% in the set of 50 experiments. In an exploratory analysis by means of classification trees no experimental parameter shows a significant influence on the validation performance measure. However, the difference relates to the absolute value of the measure 'percent of demand satisfied'. The results are shown in Table 1. The column *Range* indicates the range in the 'percent of demand satisfied'. The column '*%Difference'* is defined as 100*(the value obtained by the Posner-Berg formulae – the value obtained by the simulation) / (the value obtained by the Posner-Berg formulae). The column *'Number of points'* refers to the number of experimental points (out of 50) which fall in the specified ranges. The split into five ranges has been realized by the categorizing algorithm.

Range	*% Difference*	*Number of Points*
12-17	-13.57	5
17-53	-7.30	9
53-78	-4.01	10
78-95	-1.05	10
95-99	-0.09	16

Table 1: Simulation validation results

As good performing real production systems are supposed to be designed to obtain customer service levels of 90% or higher, we are confident that the simulation model results are within less than 1% of the analytical model results.

EXPERIMENTAL DESIGN AND DISCUSSION OF THE SIMULATION RESULTS

Inventory management systems are mostly based on the demand distribution during lead-time by means of two characteristics: mean and standard deviation. Just like in the Posner and Berg article, we assume that the demand is generated by individual orders and that each order has a size according to probability distribution. This means that the same mean and standard deviation for demand in a fixed time period (the lead-time) can be obtained by various combinations of order frequency and size.

In the experiments, we assume that the frequency of demand is modeled by a Poisson process, which means that the time between individual orders follows an exponential distribution. This assumption follows the Posner and Berg article. However these authors assume that the order size follows an exponential distribution, while our experiments deal with a very different type of distribution: the uniform distribution. The uniform distribution is chosen as a reference point as it serves the case where nothing is known about the demand except for lower and upper bounds.

Under these assumptions, the total demand during a finite time period follows a compound Poisson distribution. Given a mean and standard deviation for the total demand, the compound Poisson distributions can take very different shapes from close to the Normal distributions, to J-shaped or bimodal distributions (Ramaekers and Janssens, 2007). It is the objective of this research to investigate in how far these shape characteristics might influence the performance of the production-inventory system. With first and second moments fixed, the shape of the distributions is characterized by their normalized third and fourth moments.

The following paragraphs recall the moments for the compound Poisson distributions, related to our experiments.

Let X be the variable representing the size of an individual demand for which the probability distribution function is written as $F_X(x)$ and let S be the variable representing the total demand for which the probability distribution is written as $F_S(x)$. The distribution of the number of orders in a finite time period is described by a Poisson distribution with parameter λ. Then, the compound Poisson distribution has a density function:

$$F_S(x) = \sum_{k=0}^{\infty} e^{-\lambda} \frac{\lambda^k}{k!} F_X^{*k}(x) \qquad [1]$$

The mean and variance of the compound Poisson distribution are:

$$\mu_S = \lambda \mu_X \text{ and } \sigma_S^2 = \lambda\left(\sigma_X^2 + \mu_X^2\right) \qquad [2]$$

Further let μ'_{Si} denote the i-th central moment of the compound Poisson distribution.

Consider first the case of the Posner and Berg article, when X follows an *exponential distribution*, The knowledge of μ_S and σ^2_S completely determines the values of λ and μ_X. These values will be used as points of comparison with results obtained from the simulations.

In case the demand size follows a *uniform distribution* with density function

$$f_X(x) = \frac{1}{b-a} \text{ if } a \le x \le b, \text{ and = 0 elsewhere,} \qquad [3]$$

With this order size distribution, the second, third and fourth central moment of the compound Poisson distribution are:

$$\mu'_2 = \lambda \frac{b^3 - a^3}{3(b-a)} \qquad [4]$$

$$\mu'_3 = \lambda \frac{b^4 - a^4}{4(b-a)} \qquad [5]$$

$$\mu'_4 = \lambda \frac{b^5 - a^5}{5(b-a)} + 3\lambda^2 \left[\frac{b^3 - a^3}{3(b-a)}\right]^2. \qquad [6]$$

Given the mean and variance of the aggregated demand, denoted as μ_S and σ^2_S, the lower and upper bounds of the uniform distribution can be expressed in their values as:

$$a = \frac{\frac{2\mu_S}{\lambda} - \sqrt{\frac{12\sigma_S^2}{\lambda} - \frac{12\mu_S^2}{\lambda^2}}}{2} \qquad [7]$$

and

$$b = \frac{\frac{2\mu_S}{\lambda} + \sqrt{\frac{12\sigma_S^2}{\lambda} - \frac{12\mu_S^2}{\lambda^2}}}{2} \qquad [8]$$

From the knowledge that both a and $b \ge 0$ and that $\sigma_S^2 \ge 0$, a valid lower and upper bound for the value of λ can be determined as:

$$\frac{\mu_S^2}{\sigma_S^2} \le \lambda \le \frac{4\mu_s^2}{3\sigma_S^2}. \qquad [9]$$

(see Ramaekers and Janssens, 2007).

Simulation experiments are run for the same compound Poisson distribution with the combinations of repair rates and failure rates, which are earlier mentioned. This means we use a demand during lead time with mean equal to 0.93 and a variance equal to 5.766. Three combinations of the order arrival rate λ and both parameters of the uniform distribution for individual order size a and b are defined including two extreme combination values (the case where a = b and the case where a = 0). The data are shown in Table 2.

λ	*a*	*b*
0.150	6.200	6.200
0.175	1.557	9.072
0.200	0.000	0.300

Table 2: Experimental parameter values

The results of the simulation experiments are shown in Appendix 2. Table 3 summarises the *%Difference* according to the range of the Posner-Berg formula (similar definitions as in Table 1). It can be recognised that using the uniform distribution instead of the exponential hardly has any influence.

Range	*% Difference*	*Number of Points*
77-89	1.00	3
89-95	0.35	6
95-99	0.04	15

Table 3: Simulation results analysis

CONCLUSIONS

The analytical model by Posner and Berg seems to be very robust again the type of distribution of the individual demand. This has been shown by using uniform order size distributions which have a totally different shape than the exponential distributions which were used in their analysis. The generalization of this statement for a wider range of mean and variance of the compound Poisson distribution needs further investigation.

REFERENCES

Bartezzaghi, E., Verganti, R., Zotteri G. (1999), Measuring the impact of asymmetric demand distributions on inventories. International Journal of Production Economics, 60-61:395-404.

Brill, P. and Posner, M. (1977). Level crossings in point processes applied to queues: Single server case, Operations Research 25: 662–674.

Brill, P. and Posner, M. (1981). The system point method in exponential queues: A level crossing approach, Mathematics of Operations Research 6: 31–49.

Buzacott, J.A. and L.E. Hanifin (1978). Model of automatic transfer lines with inventory banks: a review and comparison. AIIE Transactions 10(2):197-207.

Dallery, Y., R. David and X.-L. Xie (1989). Approximate analysis of transfer lines with unreliable machines and finite buffers. IEEE Transactions on Automatic Control 34(9): 943-953.

De Koster, M.B.M. (1989). Capacity Oriented Analysis and design of Production Systems. Lecture Notes in Economics and Mathematical Systems 323, Springer Verlag, 245

Kelton, W.D., R.P. Sadowski and D.T. Sturrock (2006), Simulation with Arena (4th ed.), McGraw Hill.

Malathronas, J.P., J.D. Perkins and R.L. Smith (1983). The availability of a system of two unreliable machines connected by an intermediate tank. AIIE Transactions 15(3):195-201.

Posner, M. and Berg, M. (1989). Analysis of a production–inventory system with unreliable production facility, Operations Research Letters 8: 339–345.

Ramaekers, K. and G.K. Janssens (2007). On the choice of a demand distribution for inventory management models, European Journal of Industrial Engineering 2(4): 479-491.

Sörensen, K. and G.K. Janssens (2001). Buffer allocation and required availability in a transfer line with unreliable machines. International Journal of Production Economics 74:163-173.

Sörensen, K. and G.K. Janssens (2003). A production-inventory system of an unreliable continuous transfer line. Journal of Systems Science and Systems Engineering 12(3): 298-306.

Tzafestas S.G. (1982). Optimal and modal control of production-inventory systems. In: S.G. Tzafestas (ed.), Optimisation and Control of Dynamic Operational Research Models, North-Holland Publishing Company, 1-71.

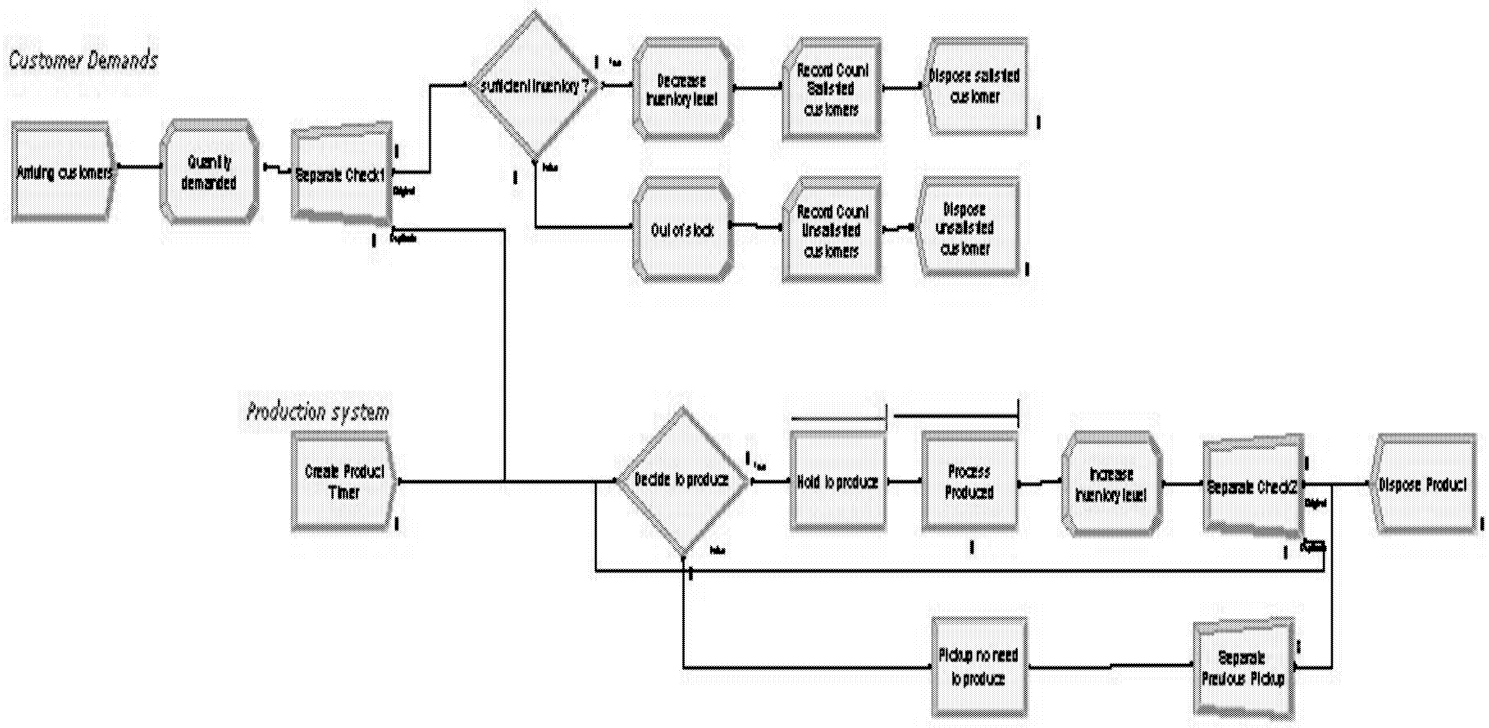
Van Oudheusden, D.L. and G.K. Janssens (1994). Availability of three-machine two-buffer systems. JORBEL (Belgian Journal of Operations Research, Statistics and Computer Science) 34(2):17-37.

Vernimmen, B., Dullaert W., Willemé P. and Witlox F. (2008), Using the inventory-theoretic framework to determine cost-minimizing supply strategies in a stochastic setting, International Journal of Production Economics, 115(1):248-259.

BIOGRAPHY

ROONGRAT PISUCHPEN received her Bachelor of Engineering in Industrial Engineering from Kasetsart University (Thailand), Master of Engineering in Industrial and Systems Engineering and PhD. in Industrial and Systems Engineering from Asian Institute of Technology (Thailand). Currently, she is an assistant professor in Industrial Engineering at Engineering Faculty, Kasetsart University in Bangkok. She is also a visiting professor and program committee member in Supply Chain Management, Turakitbundit University, Thailand. She is author of "The Handbook of Simulation Modeling with ARENA" published in Thai language. In the Arena Simulation Contest year 2009, she was an advisor for the winning team. Her main research interests include Petri Net models, simulation modeling and analysis, and applications of operations research.

APPENDIX 1



APPENDIX 2

Repair rate	Failure rate	PB_pds	Lambda	a	b	%DemSatis	HalfWidth
3	0.03	99.010	0.150	6.200	6.200	99.089	0.119
1.5	0.03	98.796	0.150	6.200	6.200	98.920	0.134
0.3	0.03	95.565	0.150	6.200	6.200	95.769	0.349
0.15	0.03	89.090	0.150	6.200	6.200	89.609	0.394
3	0.06	98.800	0.150	6.200	6.200	98.839	0.145
1.5	0.06	98.266	0.150	6.200	6.200	98.317	0.185
0.3	0.06	89.237	0.150	6.200	6.200	89.760	0.348
0.15	0.06	76.785	0.150	6.200	6.200	77.534	0.390
3	0.03	99.010	0.175	1.557	9.072	99.090	0.132
1.5	0.03	98.796	0.175	1.557	9.072	98.834	0.155
0.3	0.03	95.565	0.175	1.557	9.072	95.697	0.270
0.15	0.03	89.090	0.175	1.557	9.072	89.366	0.452
3	0.06	98.800	0.175	1.557	9.072	98.813	0.138
1.5	0.06	98.266	0.175	1.557	9.072	98.247	0.198
0.3	0.06	89.237	0.175	1.557	9.072	89.316	0.379
0.15	0.06	76.785	0.175	1.557	9.072	77.688	0.439
3	0.03	99.010	0.200	0.000	9.300	99.007	0.141
1.5	0.03	98.796	0.200	0.000	9.300	98.829	0.157
0.3	0.03	95.565	0.200	0.000	9.300	95.501	0.358
0.15	0.03	89.090	0.200	0.000	9.300	89.247	0.320
3	0.06	98.800	0.200	0.000	9.300	98.751	0.162
1.5	0.06	98.266	0.200	0.000	9.300	98.195	0.181
0.3	0.06	89.237	0.200	0.000	9.300	89.558	0.392
0.15	0.06	76.785	0.200	0.000	9.300	77.433	0.444

SIMULATION STUDY OF A BASIC INTEGRATED INVENTORY PROBLEM WITH QUALITY IMPROVEMENT AND LEAD TIME REDUCTION

Sofie Van Volsem, Johannes Verdiesen
Department of Industrial Management
Ghent University
Technologiepark 903
BE-9052 Zwijnaarde, Belgium
e-mail: Sofie.VanVolsem@UGent.be

KEYWORDS
Supply chain, quality modeling, inventory model, discrete event simulation

ABSTRACT

In this paper, we investigate the trade-off between investing in product quality improvement and lead time reduction in a B2B single-supplier, single-buyer environment. To this end, a discrete event simulation model is proposed, based on the integrated inventory model defined in Ouyang et al. (2006). Furthermore, the possibility of using alternative failure mechanisms and capital investment functions are studied.

INTRODUCTION

Supply chain management -as defined by the Council of Supply Chain Management Professionals- encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities. Importantly, it also includes coordination and collaboration with channel partners, which can be suppliers, intermediaries, third party service providers, and customers. In essence, supply chain management integrates supply and demand management within and across companies.

Supply chains have existed since man began to trade, but only during the last decades have they grown globally and have they taken the enormous proportions they do today. This of course has to do with economic globalization. An overview of the impact and effects of economic globalization is depicted in Friedman's bestseller "The World is Flat". Globalization brings opportunities as well as threats: in the context of supply chains we could state that the possible competitive advantage to be gained from efficient supply chaining becomes more important, but global supply chains are also more vulnerable. This heightened vulnerability stems from increased security risks, increased product quality risks and increased probabilities of late delivery.

Lee and Whang (2005) comment on the impact of increased security measures after 9/11 on supply chain safety and cost, while Robinson and Malhotra (2005) define the concept of supply chain quality management. Batson and McGough (2007) introduce the concept of supply chain quality modeling. They state that on one hand there has been a lot of attention for supply chain optimization, but mostly from an operational or financial point of view and not from a quality point of view; and on the other hand there are lots of quality management models but they are all geared towards optimizing quality in one entity in the supply chain and not in the supply chain as a whole. But as the quality of a firm's product depends not only on its own actions but also on the actions of its suppliers, Batson and McGough (2007) suggest a supply chain quality model to measure, predict and follow up the evolution of product quality throughout a supply chain.

In this paper the trade-off between investing in product quality improvement and lead time reduction in a B2B single-supplier, single-buyer environment is investigated. Thereto the integrated inventory model proposed by Ouyang et al. (2006), is transformed into a simulation model which is used to analyze the dynamics of the system and to study the effect of several modifications to the model. Two types of modifications to Ouyang's original model are introduced: modifications concerning the failure mechanism of the production process and modifications concerning the quality improvement (investment) function.

THE BASIC MODEL

Following Baiman et al. (2000), we assume a risk neutral buyer and a risk neutral supplier in a B2B single supplier-single buyer network. The buyer sells units to the market, which it first purchases from a supplier. This kind of situation is encountered whenever a buyer owns the brand and/or designs the product, but outsources the production to one or more suppliers. From the customer's point of view, the ultimate goal is to get the right quantity of products in the highest possible quality at the right time. Capital investment to shorten the lead time and/or to improve product quality are considered effective ways to reach this goal. This paper

investigates the trade-off between these two possible investments options under different conditions.

The proposed basic model is an integrated inventory model i.e. the goal is to minimize the joint total expected cost (sum of buyer's costs and supplier's costs) per time unit. The integrated system is triggered by the market demand. As this demand is considered stochastic, the buyer holds a certain level of stock. Whenever his stock level drops below the reorder point, the buyer places an order with the supplier, who thereupon immediately starts with the production. Each order is delivered to the buyer in a certain number of deliveries. During production it is possible that the production process gets out-of-control and thus produces defective items. These defective items can be repaired at the expense of an additional cost. The chance of the production process getting out-of-control can be influenced by investments in production quality.

There is a certain lead time before the buyer can get hold of his ordered products. Because of this lead time and the stochastic nature of the market demand, it is possible that the buyer experiences stock-outs. In this case it is assumed that a known proportion is backordered while the other part are lost sales. It is assumed that lead time can be shortened at the expense of an additional cost. Shortening the lead time implies that safety stock can be lower, stock-out losses can be reduced, and ultimately, service levels can be improved.

Our basic model is based on the work of Ouyang et al. (2006) and has 5 decision variables: the order size (Q), the reorder point (r), the lead time (L), the number of deliveries for each order (m) and the probability that the production process goes out-of-control each time another unit is produced (θ). Based on these five variables, Ouyang theoretically calculated a total expected cost for the buyer (TEC_b), consisting of inventory cost, ordering cost, delivery cost, backordering cost and crash cost, and for the vendor (TEC_v), consisting of inventory cost, set-up cost, rework cost and capital cost. The cost efficiency of the vendor-buyer system was analyzed through an evaluation of the joint total expected cost ($JTEC$), defined as:

$$JTEC = TEC_b + TEC_v \tag{1}$$

with

$$TEC_b(Q, r, L, m) = \tag{2}$$
$$\frac{D}{Q}\left\{A_b + mF + m\left[\pi + \pi_0(1-\beta)\right]E(X-r)^+\right\}$$
$$+h_b\left[\frac{Q}{2m} + r - DL + (1-\beta)E(x-r)^+\right] + \frac{mD}{Q}R(L)$$

and

$$TEC_v(Q, \theta, m) = \tag{3}$$
$$\frac{\alpha}{\delta}\ln\left(\frac{\theta_0}{\theta}\right) + \frac{A_v D}{Q} + \frac{sDQ\theta}{2} + \frac{h_v DQ}{m}\left(\frac{1}{P} + \frac{m-1}{2D} - \frac{m}{2P}\right)$$

Table 1: The basic model's decision variables

Q	Buyer's order quantity
r	Buyer's reorder point
L	Buyer's lead time
m	Number of deliveries per order
θ	Probability of the supplier's production process to go out-of-control ($\theta \leq \theta_0$)

Table 2: Ouyang's model parameters

D	average demand per period
σ	standard deviation on D
P	Production speed of the vendor (P > D)
A_b	Order cost for the buyer per order
A_v	Set-up for the vendor per set-up
F	Transportation cost per delivery
h_b	Buyer's holding cost per unit of time
h_v	Vendor's holding cost per unit of time
s	repair cost for the vendor per defect
π	Unit cost per late delivery
π_0	Unit marginal profit for the buyer
β	fraction of backorders ($0 \leq \beta \leq 1$)
θ_0	Original out-of-control probability
α	Vendor's capital cost per unit of time
δ	Percentage reduction of θ per invested dollar
X	unknown lead time distribution with mean: DL, stdev: $\sigma\sqrt{L}$

THE SIMULATION MODEL

Verification

We developed a discrete event simulation program in Matlab to analyze the dynamics of this model. To implement the program we had to step away from the original distribution-free approach and therefore could not directly use Ouyang's results to verify our simulation model. To overcome this problem, we also adapted Ouyang's equations and solved them using AMPL/CPLEX/MINOS.

When we compared the results from our program and the AMPL-model we found that there was a difference in interpretation for the lead time L. Ouyang consideres L fixed, so the buyer always receives its delivery L time units later than he ordered it. In certain cases however, it is possible that the production time is longer than L, rendering a delivery after exactly L time units infeasible. In Ouyang's theoretical model this possibility is not accounted for; but it could, however, disrupt our simulation model. Therefore we adapted Ouyang's equations again; these changes made the AMPL-results match the results from our simulations.

Table 3: Best results for the original model

Q	r	L	m	θ $\times 10^{-4}$	TECb	TECv	JTEC	stdev
500	220	42	2	0.1	1225	1288	2513	22.82
500	220	42	2	0.01	1221	1353	2573	5.36
500	170	28	2	0.1	1268	1313	2581	58.71
700	220	42	3	0.1	1249	1350	2599	7.55
700	170	28	3	0.1	1272	1333	2605	1.56
700	220	42	3	0.01	1245	1376	2621	1.57
500	220	42	3	0.1	1216	1416	2632	2.26
500	170	42	3	0.1	1264	1371	2636	4.32
500	170	28	3	0.1	1259	1389	2647	5.17
500	170	28	2	0.01	1280	1373	2652	3.66

Simulation Results

Using the simulation program we performed a 3^5 full-factorial experiment over the five variables. We chose three values for each variable and performed two simulations for each of the 243 possible combinations of the variables. We analyzed the results (Table 3), looking for relations between the variables and the total cost.
The driving factor of the total cost seemed to be the cost occurring for the buyer when his inventory is empty: lost profit due to lost sales and additional cost of back-orders. This cost is due to the stochastic nature of the demand. To minimize this cost, it is necessary to reduce the probability of an empty stock. Therefore we need to balance the size of a delivery (defined by Q/m), the reorder point r and the lead time L. It is not easy to identify one global optimum for this situation, but there are several combinations of these variables that all give satisfying results.
Although the 4 variables Q, r, L and m will be responsible for the order of magnitude of the total cost, the variable θ is not unimportant. By choosing a good value for θ it is still possible to further reduce the costs, in most cases even by more than 10%.

MODIFICATIONS TO THE FAILURE MECHANISM

In the original model there is a probability θ that the production process goes out-of-control each time another unit is produced. Once the process is out of control, all remaining products of that production run will be damaged and they will need rework to be fixed. This failure mechanism is simple but not always very realistic. Freimer et al. (2006) describes three other failure mechanisms (see Figure 1) that are frequently used in other models. We will implement these three mechanisms in our simulation program and analyze the (cost) effects on the vendor-buyer system.

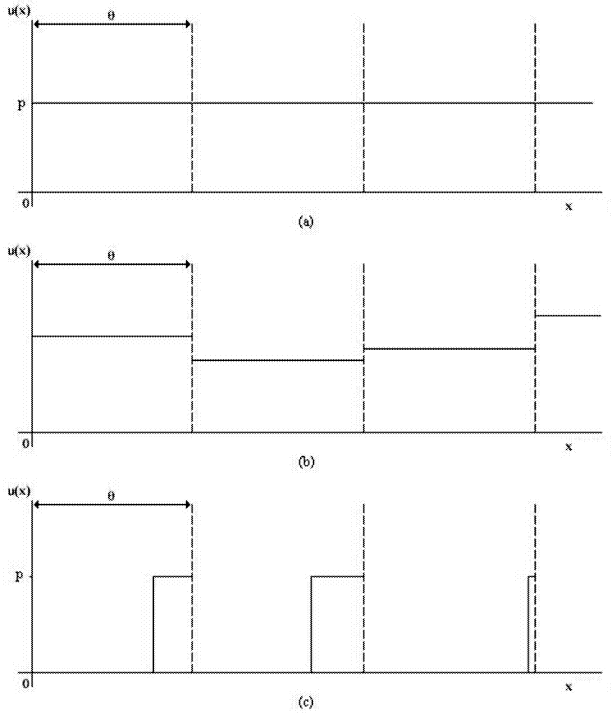


Figure 1: Freimer's different failure mechanisms:(a) Bernouilli failure, (b) Variable Bernouilli failure, (c) time to failure geometrically dictributed

(a) Bernoulli failure mechanism

For the Bernoulli-mechanism there is no longer a special out-of-control state for the production process. Every unit that is produced has the same probability p to be defective. The main implication is that the fraction of damaged products in one production run is no longer dependent on the length of the production run, because the expected fraction of defect goods will always be equal to p.

(b) Variable Bernoulli failure mechanism

The variable Bernoulli failure mechanism is very similar to the normal Bernoulli failure mechanism. The only difference is that the fraction of defective goods, p, is not the same for every production run. At the beginning of each production run the value for p is selected using a stochastic process based on a normal distribution with mean $\bar{p}$. When we consider multiple production runs the expected fraction of damaged goods will then be $\bar{p}$, but the variance of this fraction will be higher than in the previous case.

(c) Geometrically distributed time until failure

This last failure mechanism is a hybrid form between the original mechanism and the Bernoulli mechanism.

Table 4: Best results for the model with Bernoulli-failure

Q	r	L	m	p	TECb	TECv	JTEC	stddev
500	220	42	2	0.002675	1225	1302	2528	4.73
500	220	42	2	0.0005	1223	1339	2562	0.84
700	220	42	3	0.002675	1246	1322	2568	3.19
500	170	28	2	0.002675	1275	1304	2579	13.32
700	170	28	3	0.002675	1271	1326	2597	1.00
500	220	42	3	0.002675	1217	1382	2598	0.39
700	220	42	3	0.0005	1248	1361	2609	0.85
500	170	28	2	0.0005	1280	1338	2619	3.67
500	220	42	3	0.0005	1216	1420	2636	1.97
700	170	28	3	0.0005	1280	1360	2640	4.42

Table 5: Best results for the model with variable p Bernoulli-failure

Q	r	L	m	p	TECb	TECv	JTEC	stddev
500	220	42	2	0.002675	1230	1299	2529	1.08
500	220	42	2	0.0005	1229	1338	2568	2.67
700	220	42	3	0.002675	1248	1320	2568	1.99
500	170	28	2	0.002675	1273	1302	2575	4.76
500	220	42	3	0.002675	1216	1382	2598	0.77
700	170	28	3	0.002675	1279	1323	2601	8.67
500	170	28	2	0.0005	1267	1338	2605	3.20
700	220	42	3	0.0005	1246	1360	2605	1.81
700	170	28	3	0.0005	1272	1361	2633	1.96
500	220	42	3	0.0005	1215	1421	2636	0.79

Before the production process goes out-of-control, none of the products will be defective and after the process goes out-of-control a fraction p of the products will be defective. The probability of the process going out-of-control each time another unit is produced is θ'.

Simulation Results

We also performed a full-factorial experiment for each of these three new failure mechanisms. We chose the values for the new variables (p, $\bar{p}$ and θ') in such a way that the expected total number of defect units over time would be comparable to the original model.
The results (Tables 4, 5 and 6) show that the behavior of the system does not change. The best results are still obtained for the combinations of the variables that are corresponding to the best results for the original model. The only difference we observe is the variance of the total cost. The variance is lowest for the Bernoulli processes, higher for the hybrid mechanism and highest for the original mechanism.

MODIFICATIONS TO THE INVESTMENT MODEL

In the original model a continuous logarithmic capital investment function, $I(\theta) = \delta^{-1} \ln(\theta_0/\theta)$, is used to model the quality improvement on the vendor's side. This function is a good choice from a theoretical point of view, but the logarithmic function has only two parameters: θ_0, defined by the original quality of the production process and δ, which is essentially a scale parameter.

Table 6: Best results for the model with geometrically distributed time until failure

Q	r	L	m	θ' $\times 10^{-3}$	TECb	TECv	JTEC	stddev
500	220	42	2	0.0175	1226	1296	2522	18.3656
700	220	42	3	0.0175	1247	1293	2540	10.2122
500	170	28	2	0.0175	1289	1288	2578	22.8749
500	220	42	2	0.0018	1230	1355	2585	4.8385
700	170	28	3	0.0175	1276	1313	2589	11.4437
500	220	42	3	0.0175	1216	1389	2606	14.9262
700	220	42	3	0.0018	1250	1375	2626	0.6856
500	170	28	2	0.0018	1277	1354	2631	11.8734
500	170	28	3	0.0175	1259	1375	2634	10.1493
700	170	28	3	0.0018	1271	1378	2649	4.7341

Therefore the fitting capabilities are very limited in a real situation. To overcome this we introduce two new investment functions in our simulation program.

Piecewise linear function

If we use a piecewise linear function, we can divide the interval of θ in different parts and use a linear approximation in each part. The more pieces are used, the better the fit will be. In figure 2, we let θ vary between 10^{-4} and 10^{-5} in steps of 10^{-5}.

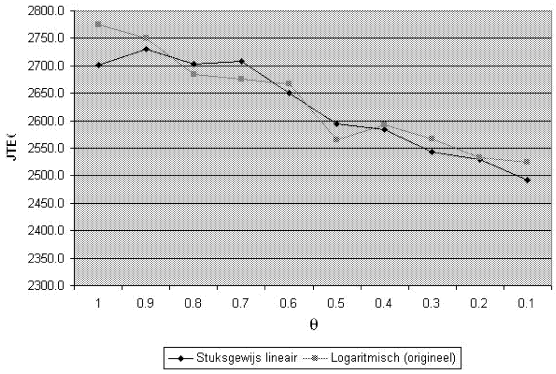


Figure 2: Comparison of the results for the logarithmic and piecewise linear investment function

Discrete function

In a real situation it could be very difficult to deduce a continuous investment function. In many cases quality improvement is done on a project basis. A certain project has a certain price and results in a corresponding quality improvement. This is no longer a continuous function but a discrete one. For the comparison in Figure 3, eight different discrete values of θ were chosen, ranging from 1.7 10^{-4} to 6 10^{-6}.

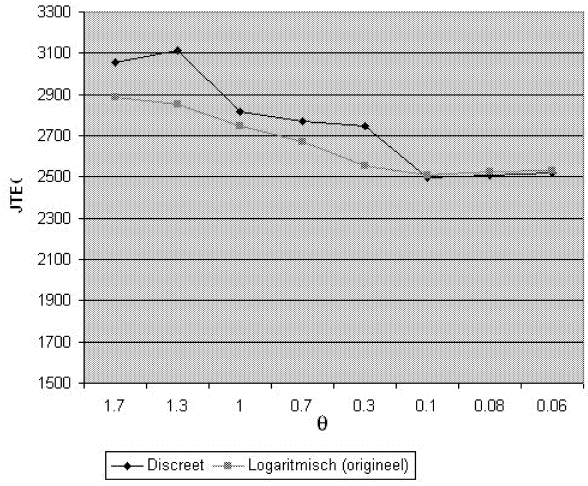


Figure 3: Comparison of the results for the logarithmic and discrete investment function

Simulation Results

We also did experiments with these new investment functions. We let θ change and kept the other variables constant. In both cases we saw the expected differences in total cost, but these were very small (Figure 2, Figure 3).

CONCLUSION

We have introduced several modifications to the original integrated inventory model. From our experiments we can conclude that the total cost is influenced mainly by the buyer in case he can not supply timely to the end customer. From the results it can be seen that it is necessary to carefully balance the buyer's order quantity Q, the buyer's reorder point r, the buyer's lead time L and the number of deliveries per order m in order to keep the total costs low. The experiments with the different failure mechanism and investment functions showed the model capable of providing meaningful results and therefore its useability in practical situations, where e.g. piecewise linear investment functions will be more readily available from historical data than the logarithmic function from the original theoretical model.

REFERENCES

Baiman S.; Fischer P.; and Rajan M., 2000. *Information, contracting, and quality costs. Management Science*, 46, no. 6, 776–789.

Batson R. and McGough K., 2007. *A new direction in quality engineering: supply chain quality modelling. International Journal of Production Research*, 45, no. 23, 5455–5464.

Freimer M.; Thomas D.; and Tyworth J., 2006. *The value of setup cost teduction and process improvement for the economic production quantity model with defects. European Journal of Operational Research*, 173, 241–251.

Lee H. and Whang S., 2005. *Higher supply chain security with lower cost: lessons from total quality management. International Journal of Production Economics*, 96, 289–300.

Ouyang L.Y.; Wu K.S.; and Ho C.H., 2006. *The single-vendor single-buyer integrated inventory problem with quality improvement and lead time reduction - minimax distribution-free approach. Asia-Pacific Journal of Operational Research*, 23, no. 3, 407–424.

Robinson C. and Malhotra M., 2005. *Defining the concept of supply chain quality management and its relevance to academic and industrial practice. International Journal of Production economics*, 96, 315–337.

AUTHOR BIOGRAPHY

SOFIE VAN VOLSEM received a MSc degree in Chemical Engineering from Ghent University in 1998 and a PhD in Engineering Sciences from the same institution in 2006. She worked in industry as a process & quality engineer before returning to academia. After being with the University of Antwerp for 6 years and the University College of West-Flanders for nearly 2 years, she currently holds a post-doc position at Ghent University. She teaches Quality & Industrial Statistics. Her research interests are quality management, quality and reliability issues in supply chains, management applications of metaheuristics.

JOHANNES (Joost) VERDIESEN graduated as Master in Engineering Sciences: Industrial Engeneering and Operational Research at Ghent University in 2009 an currently works as an associate at the Boston Consulting Group.

GREEN SUPPLY CHAIN EVENT MANAGEMENT

Matthias Klumpp
Sascha Bioly
Institute for Logistics and Service Management (ild)
FOM University of Applied Sciences
Leimkugelstraße 6
D-45141 Essen, Germany
E-Mail: matthias.klumpp@fom-ild.de

Alexandra Saur
Stephan Zelewski
Institute for Production and Industrial Information Management (PIM)
University of Duisburg-Essen
Universitätsstraße 9
D-45141 Essen, Germany
E-Mail: alexandra.saur@pim.uni-due.de

KEYWORDS
Green Supply Chain Event Management, Sustainability, Green Logistics, Supply Chain Event Management.

ABSTRACT

Sustainable logistics concepts mainly focus on strategic decision levels for example in supply chain design, warehouse location and standard transport mode selection decisions. This paper is addressing a more operative approach and analyzing green concepts for day-to-day logistics decisions, commonly discussed as supply chain event management concepts in order to speed up and improve such decisions e.g. by semi-automated software and decision systems. As the authors show in such automated operative decisions systems a green approach at least in information providing has to be discussed and can bring huge benefits for sustainable logistics in practice. As decisions will include provided information about environmental impacts operative logistics solutions are assumed to be more environmentally friendly than before.

1. INTRODUCTION

In general green and sustainable logistics concepts are focused on *strategic* decisions as e.g. location and general transport mode decisions. But nevertheless for future improvements and concept development also the dimension of *operative* logistics decisions is an important field of improvement. As practical example day-to-day situations as e.g. a production delay at an Asian factory often leads to hurried short-term transports via airplane. Thus a significant environmental impact in this decision arena can be assumed as in most cases environmental impact worsens as time pressure is increasing and speed has to be enhanced.
Therefore the concepts of supply chain event management have to be evaluated regarding sustainable information and decision making. As a guiding principle such day-to-day decisions (even automated ones in SCEM systems) should consider sustainable aspects as for example the implicated CO_2 emission by changes in transport modes (e.g. air instead of seaway). Therefore existing calculation models addressing transport mode comparisons with emission criteria (see Klumpp et al., 2009; Zelewski et al., 2009) have to be merged with existing SCEM concepts. This is the core topic of the following research paper.

2. LITERATURE REVIEW

The development of concepts in supply and logistics management towards more sustainability is driven by a *multitude* of factors:

- *Political influences* as e.g. the Kyoto Protocol of 1997 (to be followed by the Copenhagen Protocol) and emissions trading (UNFCCC, 2009).
- *Media influences* expecting data, concepts and reactions from companies in order to prove their sustainable management concept (especially regarding PR concepts).
- *Management influences* integrating the expected future raw material prices driven by shortages in raw materials due to restricted resources.

But nevertheless relevant management concepts are at best in an *early stage* of development, described by the fact that most of them describe a single perspective instead of holistic management views.
It can be recognized that the term *Green Logistics* itself already is a trend in global logistics research and is also present in agendas of companies worldwide for more than 15 years with the first authors starting to publish from 1990 onwards. Therefore the following literature review is taking into account the last 15 years.
Literature regarding logistics, supply management and supply chain management is in many cases *cost* driven (Wiedmann et al., 2008, p. 63), *quality* (Bogaschewsky et al., 2008, p. 244) and *risk* oriented (Goll et al., 2008, p. 150). Sustainability concepts are to date only implemented as *sub-factors* in concepts within these three specific perspectives or for a specified industry sector (e.g. the food sector; Hamprecht, 2005, p. 2). Even optimization models with a per se *integrated* approach are missing sustainable parameters in their objectives (Kohler, 2008, p. 10). The following literature survey is describing this in table 1. In this structure *four perspectives* were analyzed if addressed in the selected literature (by key word “green logistics”, “sustainable logistics” and “CSR” depending on availability). If a paper addressed the named perspective this is marked in the table. In order to specify the four perspectives the following description is given:

- The *Operational Event Perspective* is addressed in articles concerning logistics process management, event management and decision making as e.g. routing, scheduling or exception management.
- The *Customer Quality Perspective* is a topic in articles concerning customer and industry needs, quality aspects as e.g. cost effectiveness and low goods damage levels.

- The *Logistics Perspective* is addressed in articles with the topics like e.g. transport, network and warehousing concepts, fleet management and bundling strategies, location decisions and concepts aimed at reducing delays.
- The *Green Perspective* is addressed in articles coping with improving the sustainability and CSR in general – with subtopics as e.g. CO_2 and emissions reduction or minimizing energy consumption by transport and warehousing facilities.

In a specific overview therefore table 1 shows that until today there is no broad discussion about the *operational* event management perspective in sustainable logistics which we term green supply chain event management. There clearly is a need for an integrated concept bringing together operational, customer, logistics and sustainability expectations

Table 1: Literature Synopsis Green Logistics

Author(s)	Specific Perspective			
	Operational Event Perspective	Customer Quality Perspective	Logistics Perspective	Green Perspective
Al-Mansi et al. (2008)	X		X	
Anderson et al. (2009)		X		X
Archel et al. (2008)				X
Bowen et al. (2001)			X	X
Carter et al. (2008)				X
Darnall et al. (2008)			X	X
Fleischmann et al. (1997)			X	
Fabig et al. (2008)				X
Hamprecht (2005)		X	X	
Hussain (1999)		X		X
Kohler (2008)	X	X	X	
Koplin et al. (2007)		X		X
Krause et al. (2009)	X			X
Middendorf (2008)			X	X
Rodrigue et al. (2001)			X	
Seuring et al. (2008)	X			X
Siepermann et al. (2009)				X
Steven (2004)			X	X
Straube et al. (2008)		X	X	X
Tate (1996)		X	X	
Vieira et al. (2008)			X	X
Walton et al. (1998)			X	X
Wu et al. (1995)			X	X

3. TRANSPARENCY IN SUPPLY CHAIN OPERATIONS: SCEM

Existing supply chain event management concepts focus on real-time information and decision management aided by computer systems. Basic functions are existing *tracking and tracing tools* as transparency can be seen as a prerequisite for semi-automated or automated supply chain event management systems (SCEM). The task of SCEM mostly lies in gradually realizing all data along a supply chain. Those develop in an enterprise and have to be exchanged between enterprises (Nissen, 2002, p. 477). OKHRIN describes the task of SCEM systems as follows (Okhrin, 2008, p. 111): „Event oriented systems enable the monitoring of stocks, orders and goods deliveries along the entire delivery chain. They identify expected events as well

as unplanned disturbances and inform the decision maker about their status with the goal of a punctual identification of the disturbances and the states of emergency to guarantee."
Supply chain management (SCM) needs a smooth information flow for an efficient functionality within a delivery chain. The event oriented process management reaches this goal by giving information about the real commodity flow to all involved. SCEM has to realize thereby a permanent monitoring of material and goods flows along the entire chain and additionally has to make coordinated management action possible in case of supply disturbances and exceptional cases (Beckmann, 2003, p. 113). An example SCEM model shows *figure 1* (Schenk et al., 2007, p. 224).

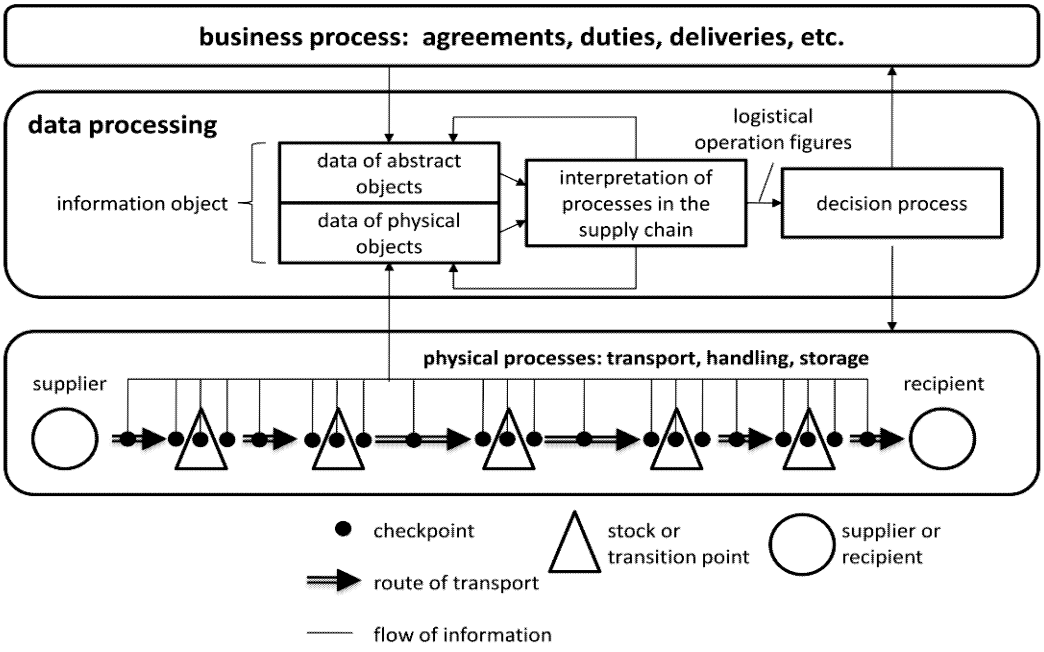


Figure 1: SCEM Strategy Model

The task of SCEM is an active and customer oriented monitoring of the delivery chain to recognize disturbances and give possible solutions. Thus SCEM increases the flexibility and capacity of reaction of the entire supply chain.
The first theoretical bases of SCEM were already compiled in the form of elaboration about management by exception (MBE) in the middle of the last century, whereby beginnings of practical field use can be found in the bases of the tracking and tracing (Hunewald, 2005, p. 9; Wildemann, 2007, p. 13). Characteristic for the MBE approach is the fact of reducing control and steering activities of the responsible person. An intervention is only necessary if an event cannot be processed and/or settled independently by the SCEM system (Bittel et al., 1964, p. 5). With SCEM an interface is provided between the pre-defining supply chain planning (SCP), the planned process and the real operational sequence along the supply chain execution (Arnold et al., 2008, p. 481; Nissen, 2002, p. 477).
If deviations between the current actual condition of the process and the planned process are observed, SCEM introduces immediately a set of reaction measures, which serve for the recovery of the disturbances and a regular continuation of the process and/or suggest alternative solution types. Things like tracking and tracing, the traceability and on-line arrangement of goods, charge carriers and vehicles are bundled with additional functions (e.g. warning functions) for the better control and decision support provided. SCEM cannot replace the fundamental SCP but builds further on this (Bretzke, 2002, p. 28).
A SCEM system can extend the functionality of tracking and tracing applications. The generated status messages are passed on to the decision maker in real time. These of all participants of the supply chain collected data are supervised and interpreted by the event management system. In the further process SCEM accomplishes a comparison of nominal and actual values and designates the signals in planned events or unplanned disturbances (Klaus, 2004, p. 13). This is the main task of the SCEM system: If the system registers an incident, thus a plan deviation from the defined specified condition, it tries to make a rapid reorganization of the process available on the basis of pre-defined solution alternatives. However this is only possible if a potential scenario is programmed in the event system and possible alternative solutions are implemented (Karrer, 2003, p. 188).
The biggest advantage of SCEM is transparency over several stages of the delivery chain, since ideally all individual shipments can be constantly supervised and steered. High complexity and dynamics of the processes in a value chain, however, make the effective implementation of SCEM difficult. Thus it is required that all processes are integrated along a supply chain in the event management system, because only with completely integrated and not partly omitted processes can an optimal reaction of the event management take place (Wildemann, 2007, p. 41). As a condition for this a greatest possible information transparency along the entire supply chain has to be ensured (Wildemann, 2007, p. 44). In the course of constantly growing requirements of participants in the supply chain (shipper - service provider - customer) gains in particular the use of SCEM increasingly in meaning. Tracking and tracing, warning functions and further telematic components increase the quality and topicality of information about the whole

supply chain. In direct consequence these tools permit optimal planning extending reactivity and effectiveness with deviations and exceptional cases. SCEM can improve the efficiency and security of logistics processes. SCEM deployment optimizes the yield situation and customer satisfaction. The information gain concerning business processes secures a positive prognosis for the enterprise used by SCEM.

4. GREEN SUPPLY CHAIN EVENT MANAGEMENT MODEL

In order to *integrate* all the specified perspectives needed in a Green SCEM system a holistic management model has to be drafted – drawing structure from the perspectives described in the literature review above. First there has to be a *sustainability* or *green perspective* as described above containing the following:

- *Input Reduction* objective calling for lower inputs of non-renewable materials as e.g. energy and raw materials needed for transport equipment and transport and logistics services.
- *Safety* objective describing the absence of harmful events such as oil and other dangerous goods spills in natural habitats or human injuries.
- *Pollution efficiency* objective determining a reduction of emissions of e.g. greenhouse gases or other pollutants in relation to logistics service outputs.

A *logistics perspective* is underpinned by the following three important factors:

- *Availability* objective describing the basic function of logistics to ensure availability of the right goods at the right place and on time.
- *Quality* objective addressing the need for unharmed goods transport and smoothness of logistics *services* (service orientation, security awareness).
- *Transparency* objective in logistics depicting the aim to provide accurate and real-time information about transport, goods status and overall logistics performance for customers and other partners in the supply chain.
- The *customer perspective* is described further by the following three objectives:
- *Value* objective addressing the ration of costs and *product* quality in purchasing to be guarded and improved.
- *Risk* objective defining an overall risk management approach in order to avoid situations threatening company existence.
- *Process efficiency* objective determining the process time and internal process costs to be reduced in supply management e.g. by E-Procurement.

The *event and flexibility perspective* is outlined by the following three factors:

- *Speed* objective according to standard events in supply chains as usually disruptions in the transport chain create a need for higher speed.
- *Acceptance* objective mentioning that with increasing technology impact on all steps, persons and companies in a supply chain there has to be more emphasizing of acceptance.
- *Implementation* objective determining the fact that future technologies will need even more education and training efforts in order to fledge their full potential in supply chain event management (*implementation hurdle*).

Altogether these factors build an integrated view and would enable a "Green SCEM Cockpit" for logistics decisions – for the first time including sustainable decision facts. This could lead to more sustainable operational logistics decisions e.g. in transport modes. A simulation of this planned data is currently run by the athors with example data from DACHSER in Cologne/Germany.

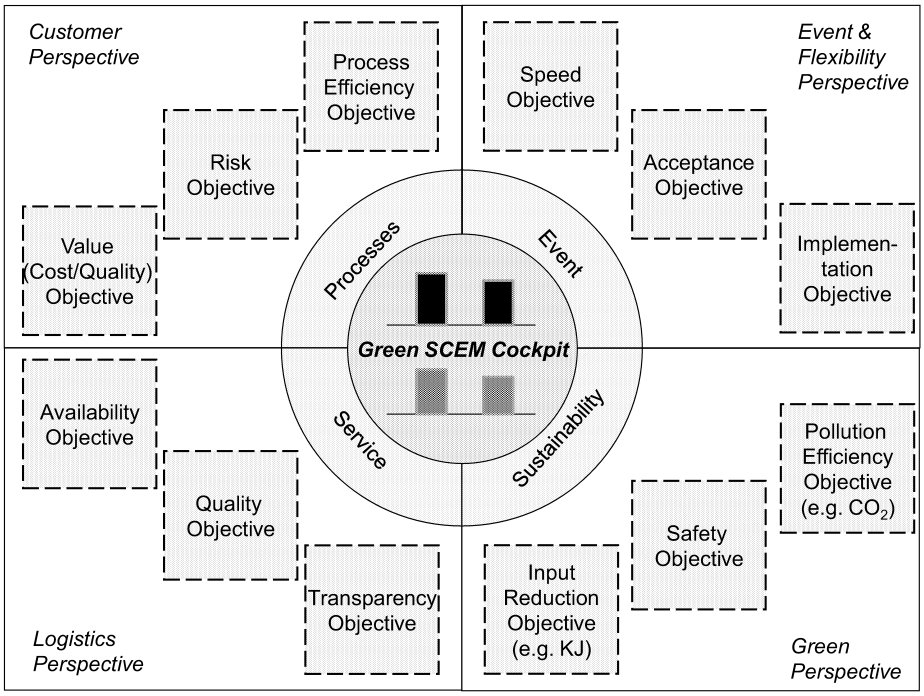


Figure 2: Green Supply Chain Event Management Model

5. OUTLOOK AND FUTURE DEVELOPMENT

Future developments will bring a *further interest in sustainability* concepts as for example the visitors of the largest European logistics fair in Munich predicted (DB Schenker Laboratories, 2009, p. 1). These experts see green and sustainable logistics concepts as the *most important trend* for 2015. Therefore research has to adapt to this trend and also to an increasing number of questions asked by practitioners in supply and logistics management regarding green concepts. This research demand will have to be answered in three areas:

- *Measurement* and indicator concepts have to be established and validated as suggested in this article.
- *Management* concepts for Green SCEM have to be developed in order to guarantee for an integrated perspective on green supply and logistics.
- *Implementation and education* concepts have to be detailed in order to secure the impact of such research and management theory concepts.

If research is fostering these challenges there will be enough chances to contribute to the global task of sustainability also by operative logistics and SCEM.

Grant Support Notice
The research for this article was supported by research funding in the projects *LOGFOR* (German NRW Department MWME as well as EU - ERDF) and *MAEKAS* (German Federal Department BMWi).

REFERENCES

Al-Mansi, A., Aldarrat, H., Rhoma, F., Goudz, A., Lorenz, M., and Noche, B. (2008). "Design of an Environmental Supply Chain Network: A Biosolid Waste Case Study". In *Global Logistics Management*, Kersten, W., Blecker, T., and Fläming, H. (Eds.). ESV: Berlin, 87-98.

Anderson, M., and Skojett-Larsen, T. (2009). "Corporate Social Responsibility in Global Supply Chains". In *Supply Chain Management - An International Journal*, 14 (2), 75-86.

Archel, P., Fernandez, M., and Larringa, C. (2008). "The Organizational and Operational Boundaries of Triple Bottom Line Reporting: A Survey. In *Environmental Management*, 41, 106-117.

Arnold, D., Isermann, H., Kuhn, A., Furmans, K., and Tempelmeier, H. (Eds.) (2008). *Handbuch Logistik*. Springer: Berlin.

Beckmann, H. (Eds.) (2004). *Supply Chain Management*, Springer: Berlin.

Bittel, L. R., and Maynard, H. B. (1964). *Management by Exception: Systematizing and simplifying the managerial job*. McGraw-Hill: New York.

Bogaschewsky, R., and Müller, H. (2008). „Stand und Weiterentwicklung des E-Procurement in Deutschland". In *Best Practice in Einkauf und Logistik, 2nd edition*. BME (Eds.). Gabler: Wiesbaden, 237-254.

Bowen, F. E., Cousins, P. D., Lamming, R. C., and Faruk, A. C. (2001). "The role of supply management capabilities in green supply". In *Production and Operations Management,* 10 (2), 174-189.

Bretzke, W.-R. (2002). "SCEM – Entwicklungsperspektive für Logistikdienstleister". In *Supply Chain Management,* 2 (3). 27-31.

Carter, C. R., and Rogers, D. S. (2008). "A framework of sustainable supply chain management: Moving toward new theory". In *International Journal of Physical Distribution & Logistics Management,* 38 (5), 360-387.

Darnall, N., Jolley, G. J., and Handfield, R. (2008). "Environmental Management Systems and Green Supply Chain Management: Complements for Sustainability?". In *Business Strategy and the Environment,* 18, 30-45.

DB Schenker Laboratories (2009). Ergebnisse der Messebefragung "transport logistic 2009", Eigenverlag: Berlin.

Fabig, K,. and Pinnau, H. (2008). "Wassereinsparung im Sanitärbereich öffentlicher Einrichtungen in Hamburg". In *Best Practice in Einkauf und Logistik, 2nd edition*. BME (Eds.). Gabler: Wiesbaden, 304-316.

Fleischmann, M., Bloemhof-Ruwaard, J. M., Dekker, R., van der Laan, E. A., van Nunen, J. A. E. E., and van Wassenhove, L. N. (1997). *Quantitative models for reverse logistics: A review, European Journal of Operational Research*, 103, 1-17.

Goll, L., and Haupt, S. (2008). "Corporate Governance, Risk- and Compliance Management in der Beschaffung". In *Best Practice in Einkauf und Logistik*. BME (Eds.). Gabler: Wiesbaden, 149-168.

Hamprecht, J. (2005). *Sustainable Purchasing Strategy in the Food Industry*. DiFo: Bamberg.

Hunewald, C. (2005). *Supply Chain Event Management – Anforderungen und Potentiale am Beispiel der Automobilindustrie*. DUV: Wiesbaden.

Hussain, S. S. (1999): "The Ethics of ‚Going Green': The Corporate Social Responsibility Debate". In *Business Strategy and the Environment,* 8, 203-210.

Karrer, M. (2003). "Supply Chain Event Management – Impulse zur ereignisorientierten Steuerung von Supply Chains". In *Innovationen im E-Business, series Innovative Produktion und Logistik*. Dangelmaier, W. (Eds.). Vol. 10, ALB-HNI: Paderborn, 187-198.

Klaus, O. (2004). "Geschäftsregeln im Supply Chain Event Management". In *Supply Chain Management,* 4 (2), 13-19.

Klumpp, M., Zelewski, S., and Saur, A. (2009). "Increasing Rail Cargo Transport Performance". In *High-Performance Logistics*. Blecker, T., Kersten, W., and Meyer, M. (Eds.). ESV: Berlin, 17-30.

Kohler, K. (2008). *Global Supply Chain Design*, CfSM: Estenfeld.

Koplin, J., Seuring, S., and Mesterharm, M. (2007). "Incorporating sustainability into supply management in the automotive industry – the case of the Volkswagen AG". In *Journal of Cleaner Production,* 15 (2007), 1053-106.

Krause, D. R., Vachon, S., and Klassen, R. D. (2009). "Special Topic Forum on Sustainable Supply Chain Management: Introduction and Reflections on the Role of Purchasing Management". In *Journal of Supply Chain Management,* 45 (4), 18-25.

Middendorf, K. (2008). "Logistik im Spannungsfeld zwischen Globalisierung und Nachhaltigkeit". In *Das Beste der Logistik*. Baumgarten, H. (Eds.). Springer: Berlin, 405-414.

Nissen, V. (2002). „Supply Chain Event Management". In *Wirtschaftsinformatik*, 44 (5), 477-480.

Okhrin, I. (2008). "Supply Chain Event Management". In *Handbuch Logistik*. Arnold, D., Isermann, H., Kuhn, A., Furmans, K., Tempelmeier, H. (Eds.) Gabler: Berlin, 481-485.

Rodrigue, J.-P., Slack, B., and Comtois, C. (2001). "Green Logistics (The Paradoxon of)". In *The Handbook of Logistics and Supply-Chain Management*. Brewer, A. M., Button, K. J., and Hensher, D. A. (Eds.) Pergamon: London, 339-350.

Schenk, M., Tolujew, J., Barfus, K., and Reggelin, T. (2007). "Grundkonzepte zu logistischen Echtzeitsystemen: Monitoring, Event Management und Frühwarnung". In *Jahrbuch der*

Logistik 2007. Wolf-Kluthausen, H. (Eds.). Korschenbroich, 222-226.

Schneider, C. (2004). *Controlling für Logistikdienstleister – Konzepte, Instrumente, Anwendungsbeispiele, Trends*. DVV: Hamburg.

Seuring, S., and Müller, M. (2008). "From a literature review to a conceptual framework for sustainable supply chain management". In *Journal of Cleaner Production,* 16/2008, 1699-1710.

Steven, M. (2004). "Network in Reverse Logistics". In *Supply Chain Management and Reverse Logistics*. Dyckerhoff, H., Lackes, R., and Reese, J. (Eds.). Springer: Berlin, Heidelberg, New York, 163-180.

Straube, F., and Borkowski, S. (2008). *Global Logistics 2015+ - How the world´s leading companies turn their logistics flexible, green and global and how this affects logistics service providers*. BVL: Berlin.

Tate, K. (1996). "The elements of a successful logistics partnership". In *International Journal of Physical Distribution & Logistics Management,* 26 (3), 7-13.

UNFCCC (2009). http://www.unfccc.int, date 12.02.2010.

Vahrenkamp, R., and Siepermann, C. (Eds.) (2007). *Risikomanagement in Supply Chains – Gefahren abwehren, Chancen nutzen, Erfolg generieren*. ESV: Berlin.

Vieira, D. R., da Silva, H. A. and dos Santos, A. (2008). "Integrated Engineering – The Case of Sustainable Development of Packaging in the Food Industry". In *A New Wave of Innovation in Collaborative Networks*. Pawar, K., Thoben, K. D., and Goncales, R. (Eds.). Nottingham University: Nottingham, 325-332.

Walton, S. V., Handfield, R. B., and Melnyk, S. A. (1998): "The Green Supply Chain: Integrating Suppliers into Environmental Management Processes". In *International Journal of Purchasing and Materials Management,* 1998 (1), 2-11.

Wiedmann, H., and Teichmann, J. (2008). "Next Level Purchasing: Erfolgsfaktor eines aktiven Kostenmanagements". In *Best Practice in Einkauf und Logistik, 2nd edition*. BME (Eds.). Gabler: Wiesbaden, 56-65.

Wildemann, H. (2007). *Event Management in der Supply Chain – Leitfaden zur Steuerung geplanter und zufälliger Ereignisse entlang der Supply Chain*. TCW: München.

Wu, H.-J., and Dunnin, S. C. (1995). "Environmentally responsible logistics systems". In *International Journal of Physical Distribution & Logistics Management,* 25 (2), 20-35.

Zelewski, S., Klumpp, M., and Saur, A. (2009). "Leerfahrtenoptimierung und Kapazitätserweiterung durch Kooperationen von Eisenbahnverkehrsunternehmen". In *Höchstleistung im spurgeführten System*. Schütte, J. (Eds.). Eigenverlag: Dresden, 1-29.

TELECOM NETWORKS SIMULATION

Achievable Cell Throughput in 3G Systems

Péter Fazekas

Department of Telecommunications
Budapest University of Technology and Economics
Budapest, Hungary
Email: fazekasp@hit.bme.hu

***ABSTRACT*—In this paper a method is shown that enables the estimation of average useful carrier throughput in a 3G cellular system. The method considers multiple bearer types and investigates systems with or without HSDPA service deployed. This analysis allows a fast and effective capacity modelling and dimensioning of the network. The effects of network layout, system parameters, channel and user spatial distribution on carrier throughput is investigated in terms of numerical results. The applicability of the modelling method is verified by means of snapshot mobile network simulations.**

INTRODUCTION

In the past few years the WCDMA based 3G systems were rapidly deployed throughout Europe. Release 6 of the standards already contained HSDPA (High speed Downlink Packet Access) allowing the theoretical maximum gross physical transmission rate of 14.4 Mbps, hence the use of cellular 3G networks for Internet access is quickly spreading today.

As on the radio interface WCDMA (Wideband Code Division Multiple Access) is applied, due to interference reasons the offered capacity of a cell is not pre-determined, rather it itself depends on traffic and propagation issues, thus radio capacity, coverage and carried traffic are dependent on each other and are strongly coupled. This is an inherent property of all systems using CDMA technology, it causes that a single measure of cell capacity (e.g. in kbps units, meaning the amount of traffic that can be served by the cell) cannot be directly determined, since it depends on constitution of the traffic mix, the volume of it, user positions and requirements and the carried traffic of neighboring cells.

However, obtaining this measure, that is able to estimate the achievable useful throughput (that can be viewed as capacity) of a cell under realistic circumstances has enormous importance during network dimensioning process, where the rough number and placement of cells to cover a given are has to be determined. The goal of this study is to develop a method for obtaining the average useful throughput of a radio cell.

The novelty of the approach presented in this paper is the method of handling the multiservice nature of UMTS, in contrast with the fact that most literature deals with capacity only in case of the presence of a single connection type. Moreover, this modelling captures the effect of multi-path propagation by means of the use of distance-dependent orthogonality factor (OF). The effect of users' spatial distribution and cell size is also incorporated and investigated, as well as the co-existence of traditional Release'99 UMTS traffic and HSDPA traffic on the same carrier frequency in a cell. Last, but not least, the fact that there are several HSDPA terminal types with different capabilities is also incorporated.

Although in the past decade numerous studies were published about WCDMA and 3G performance (e.g. [1],[2],[3]), but these capacity evaluations either lack the realistic assumptions above or based on detailed simulations but not provide simple analytic formulae.

SYSTEM DESCRIPTION

Basically the analysis concentrates on a given cell, however the effect of surrounding cells will be considered. I suppose omni directional transmit antennas and terrain without mayor obstacles, resulting in circular cell areas, with the base station in the middle of the circle. The radius of the examined cells is denoted with $R\mathrm{cell}$. The placement of neighboring base stations are determined according to standard hexagonal layout, the influence of two rings of neighboring cells (18 base stations) will be considered.

The interference effect of neighbors are calculated based on channel attenuation model. Channel is considered to influence the signal power with exponential path loss, generally

$$\mathrm{PL}(r) = \beta \cdot r^{-\gamma}. \tag{1}$$

This exponential model incorporates well known channel models, such as COST 231, Okumura-Hata or Hata models, where different propagation environments (e.g. rural, suburban, urban), carrier frequencies and topological parameters (base station height, mobile antenna height) are considered via the parameters β and γ in (1). Multipath effect is causing the loss of orthogonality among channelization codes in the downlink. This is taken into account by means of the orthogonality factor. As suggested in [4], this depends on the receiver terminal's distance from the base station (the further the terminal is, the more severe the loss of orthogonality is)

as:

$$\rho(r) = \frac{1}{1+\beta \cdot r}, \tag{2}$$

Neighboring base stations are modelled as source of interference power, transmitted at constant level, but unlike most of the literature, here base station powers may be different.

I assume that UMTS users and HSDPA users are present in the cell and a single carrier accommodates both traffic. Users of UMTS might use any of K service classes (radio bearers). In practical systems just a few services (radio bearers) are used. The typical services have useful data rate (not including the redundancy induced by forward error correction and necessary signalling bits) of 12.2 kbs (maximum rate of the variable bitrate AMR speech codec), 64 kbps (low resolution video, high quality audio or low rate data), 144 and 384 kbps bearers (video or data).

Each service class is characterised by the signal to interference ratio required for acceptable performance. This is more precisely expressed in terms of required bit energy per interference spectral density ratio $\frac{E_b}{I_0}$, for now on the $\frac{E_b}{I_0}$ requirement is denoted by ε_k for service type k or user k. The values of ϵ_k are assumed to be known. In the literature, several sets of values can be found, based on link-level simulations or just estimations.

User distribution is assumed to be known over the cell area and given in terms of the pdf of user positions, different distribution for different service class users may be assumed.

CAPACITY OF RELEASE'99 RADIO INTERFACE

In order to analyse the 3G operation with HSDPA deployed, first a cell with only conventional UMTS traffic is analysed. The following investigation is based on the basic downlink SIR (Signal to Interference Ratio) equation of UMTS systems (e.g. [1], but naturally all papers use some form of these basic equations). The bit energy over interference spectral density ratio should be over a given threshold to achieve the necessary bit error ratio for a service. Given that user i is placed on polar coordinates (r_i, ϕ_i) with the serving base station (number 0) in the origo this relationship is shown in (3). In this expression $P_i^0(r_i, \phi_i)$ is the transmission power targeted to user i, $L^0(r_i)$ is the channel gain between the serving base station and the user, depending on the user's distance r_i, P_{inst}^0-s is the total instantaneous transmitted power of the serving base station, P_{inst}^b denotes the instantaneous power radiated by the bth neighboring base station, $L^b(r_i, \phi_i)$ is the path gain between this and user i. The power of the pilot and control channels is included in P_{inst}^0, thus its power is $P_{\text{Pil}}^0 = P_{\text{inst}}^0 - \sum_i P_i^0$, $\rho(r_i)$ is the orthogonality factor, P_{noise} is the power of the thermal noise. Rc and Rb_i are the chiprate and bitrate of the given service used by the customer (and their ratio is the processing gain).

Inequality (3) must hold for all active connections. The amount of traffic is bounded by the finite transmission power of the base station.

The multi-service nature of UMTS is taken into account by means of different ε requirements and the different processing gains of services, hence the power required for a service depends not only on its distance from the base station.

I define the average capacity of a UMTS carrier as following. Let the useful bitrate of the kth bearer type be denoted by Rb_k and the average number of active bearer k is N_k. Given these circumstances the average capacity of a UMTS carrier is defined as

$$\overline{R}_{\text{UMTS}} = \sum_{k=1}^{K} N_k \cdot Rb_k = N \cdot \sum_{k=1}^{K} n_k \cdot Rb_k \tag{5}$$

where N is the average number of total scheduled connections and the ratio of type k bearers is n_k.

This quantity characterizes the average amount of useful throughput carried over a UMTS carrier, therefore the term capacity is used. In the following, the term capacity and throughput will be used interchangeably for this quantity.

To obtain carrier throughput, the following calculations should be performed. After rearranging (3) and considering that power control forces the system to achieve, but not to exceed the SIR requirement (i.e. the equality is supposed) and using that P_{inst}^0 is the sum of all used power, we arrive to the linear system of equations in (4). In this equation $\eta_i = \frac{P_{\text{noise}}}{L^0(r_i)}$ is the relative noise power and we define $f^b(r_i, \phi_i) = \frac{L^b(r_i, \phi_i)}{L^0(r_i)}$, the geometry factor. Let us concentrate on the mean of transmitted powers in the system. If we take the expectation of (4), this result in the average power transmitted to a user. This average will be the same for all terminals using a given bearer type k. Consequently, the following expression is viewed as the average transmitted power to a type k connection:

$$\overline{P}_k^0 = \varepsilon_k \cdot \frac{Rb_k}{Rc} \cdot \left((1 - \overline{\rho_k}) \cdot P_{\text{avg}}^0 + \sum_{b \neq 0} P_{\text{avg}}^b \cdot \overline{f_k^b} + \overline{\eta_k} \right). \tag{6}$$

In the expression above, P_{avg}^0 and P_{avg}^b denote the average output powers of the examined base station and that of the interfering base stations. The average of those parameters that depend on user position (relative noise, orthogonality factor, geometry factor) are determined by simply integrating their distance (and direction) dependent expression multiplied with user density function, over the cell. In case of evenly distributed users and exponential pathloss the average OF and relative noise power can be determined in closed form. After determinating the required mean parameters, equation (6) must be solved for all K service classes with a supposed level of average used power (P_{avg}^0). As result, we have average power levels of all K bearer types, namely $\overline{P}_k^0$.

The average number of simultaneous transmissions at the radio interface was earlier introduced and denoted by N. As stated earlier, this means that on a long time average, there is $N \cdot n_k$ active connections of type k on the radio interface.

$$\left[\frac{E_b}{I_0}\right]_i = \frac{Rc}{Rb_i} \cdot \frac{P_i^0(r_i,\phi_i)\cdot L^0(r_i)}{(1-\rho(r_i))\cdot P_{\text{inst}}^0 \cdot L^0(r_i) + \sum_{b\neq 0} P_{\text{inst}}^b \cdot L^b(r_i,\phi_i) + P_{\text{noise}}} \geq \varepsilon_i \tag{3}$$

$$P_i^0(r_i,\phi_i) = \varepsilon_i \cdot \frac{Rb_i}{Rc} \cdot \left((1-\rho(r_i)) \cdot \left(P_{\text{Pil}}^0 + \sum_j P_j^0(r_j,\phi_j) \right) + \sum_{b\neq 0} P_{\text{inst}}^b \cdot \frac{L^b(r_i,\phi_i)}{L^0(r_i)} + \eta_i \right), \tag{4}$$

$$\overline{P}_k^0 = \varepsilon_k \cdot \frac{Rb_k}{Rc} \cdot \left((1-\overline{\rho_k}) \cdot (N \cdot \sum_{l=1}^{K} n_l \overline{P}_l^0 + P_{\text{Pil}}^0) + \sum_{b\neq 0} P_{\text{avg}}^b \cdot \overline{f_k^b} + \overline{\eta_k} \right). \tag{7}$$

$$\text{SIR}_{\text{HS}}(r,\phi) = 16 \cdot \frac{P_{\text{HS}}^0 \cdot L_i^0(r,\phi)}{(1-\rho(r)) \cdot P_{\text{inst}}^0 \cdot L_i^0(r,\phi) + \sum_{b\neq 0} P_{\text{inst}}^b \cdot L_i^b(r,\phi) + P_{\text{noise}}}. \tag{8}$$

This inherently means, that the average used output power is

$$N \cdot \sum_{k=1}^{K} n_k \cdot \overline{P}_k^0 + P_{\text{Pil}}^0 = P_{\text{avg}}^0. \tag{9}$$

After the average power levels are calculated from (6), N has to be determined from (9). After having N, the average useful capacity (or average useful throughput using a given average power level) of the UMTS bearer is simply calculated as (5).

For the further capacity evaluation of 3G radio interface, with HSDPA enabled, we need another measure of the system. This is the average used power in case of average traffic of R_{UMTS} kbps. The former idea can be used again in the reverse direction. Namely, for obtaining the used power, N is expressed from (**??**), with the given R_{UMTS} traffic. Then, since in this case P_{avg}^0 is unknown, it's expression from (9) with the previously determined N should be substituted into (6). Then we arrive to a slightly modified version of the basic equations, namely (7). This linear system has to be solved for all $\overline{P}_k^0$-s, then the resultant average used power is given as in (9).

CAPACITY AND PERFORMANCE OF HSDPA

To evaluate the average capacity of a 3G carrier when HSDPA is deployed, the basic SIR equation is the starting point, written for the HSDPA channel. For a HSDPA user i, served by base station 0 this is according to (8). In this expression the term 16 is the fixed spreading factor used in the HSDPA channel. Given that HSDPA uses multicode transmission, each physical channel (spreading code) causes interference to the other ones due to multipath propagation, hence the power of HSDPA also appear in the denominator of (8) as the part of the total output power. Considering this, the only change regarding the former equations is that the HS-DSCH channel appears as if it was a new type of UMTS service class, but it *does not have an explicit SIR requirement*, rather it chooses transport format according to the SIR of the channel.

During operation, the HSDPA terminal reports it's perceived SIR to the base station by means of the CQI (Channel Quality Indicator) parameter, that – along with thew terminal category – explicitly determines the transport format it can receive. Thus, in order to evaluate HSDPA in terms of transmission capacity, a relationship is needed which connects SIR with CQI. Most of the literature rely on [5], where an empirical relation was given, based on detailed simulations. Another expression found for this relationship is given in [6], that is simply $CQI = \lfloor SIR + 3.5 \rfloor$.

The standard contains tables that define the applicable transport format when a given CQI is reported, for each terminal category. The transport format explicitly determines the transmitted useful bits during a frame, hence the instantaneous useful transmission rate. Thus, from (8) the achievable throughput of a terminal of category i can be directly derived, using the mapping

$$\text{SIR}_i(r,\phi) \Rightarrow \text{CQI}_i(r,\phi) \Rightarrow R_i(r,\phi), \tag{10}$$

where $R_i(r,\phi)$ is the achievable transmission rate of a category i terminal at position (r,ϕ).

The average throughput achievable by the category (R_i) is then calculated by integrating over the area with the user density pdf.We assume that there is information on the ratio of different terminal categories among all the HSDPA devices, that is α_i, $i \in [1..12]$ for category i. The average cell capacity of HSDPA is then given by averaging over the device categories:

$$R_{\text{HSDPA}} = \sum_{i=1}^{12} R_i \cdot \alpha_i. \tag{11}$$

To take deeper insight into system performance we have to handle the case when HSDPA service uses the power and code resource that is left unused by the Release'99 traffic. That is, the HSDPA resource is not supposed to be known in advance, but it is calculated from the Release'99 traffic amount.

In this case the following method is applicable. The aim is to determine the average available power and with this, the achievable average throughput of HSDPA services, given that the cell is carrying an average of R_{UMTS} kbps UMTS traffic. The used power of UMTS has to be determined basically according to the method described above. The

difference is, that HSDPA traffic also means intra-cell interference for Release'99 traffic and vice versa, thus the linear system of equations (7) is changed. As we are interested in HSDPA capacity and HSDPA utilizes the total remaining power of the Node B, the modified expression contains all the maximum possible output power of the Node B, namely

$$\overline{P}_k^0 = \varepsilon_k \cdot \frac{Rb_k}{Rc} \cdot \left((1 - \overline{\rho_k}) \cdot (P_0^0) + \sum_{b \neq 0} P_{\text{avg}}^b \cdot \overline{f_k^b} + \overline{\eta_k} \right), \tag{12}$$

and the average HSDPA power is coming from the last equation of the system, that is the one that assures that all achievable power is allocated, namely

$$\overline{P}_{\text{HS}}^0 = P_0^0 - N \cdot \sum_{k=1}^{K} n_k \cdot \overline{P}_k^0 - P_{\text{Pil}}^0. \tag{13}$$

As earlier, during this task the average number of "virtual" Release'99 connections is given through the average useful traffic R_{UMTS}, namely

$$N = \frac{R_{\text{UMTS}}}{\sum_{k=1}^{K} n_k \cdot Rb_k}. \tag{14}$$

It is easy to see from the (7) and (12) what was explained earlier: including HSDPA traffic will increase the necessary power level for UMTS, thus allowed HSDPA power is is not just the difference between the total output power and UMTS power.

Obtaining average HSDPA throughput and mean achievable HSDPA user throughput on a location again simply means determining the SIR from (8), with the average $\overline{P}_{\text{HS}}^0$ and then applying the method described above.

NUMERICAL EVALUATION

In this Section the average capacity of 3G systems is evaluated, based on the numerical methods investigated in previous sections. The analytical examinations are compared with results of snapshot simulations. The latter is based on generating uncorrelated random snapshots of the possible traffic scenarios. During the simulations, a customer is placed randomly over the examined radio cell area, according to given distributions (three possible distributions are examined, as explained later). Then, according to traffic mix distribution, its requested service is chosen. This generation of customers repeats until the system runs out of resources (radio power). As result, the sum of useful bitrates the instantaneous cell capacity is derived, the average capacity is determined as the average of capacities resulting from large number of such snapshot simulations.

Capacity of UMTS system without HSDPA service

The first set of investigations target a UMTS cell without HSDPA services. The effect of used power, user spatial distribution, service class ratios, cell size and distance-dependent orthogonality factor are revealed.

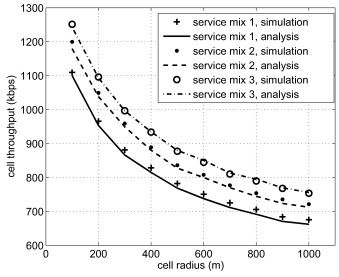


Fig. 1. Average cell throughput with distance-dependent orthogonality factor, even user distribution

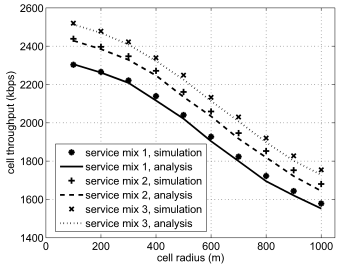


Fig. 2. Average cell throughput in hotspot scenario

In general, three basic service mixes are investigated, namely the ratio of 12.2, 64, 144 and 384 kbps connections are assumed to be 0.4 0.3 0.2 and 0.1 in service mix 1; in service mix 2 and service mix 3 the ratio of 384kbps bearer is increased to 0.2 and 0.3 at the voice bearer's expense. Figure 1 displays the average achievable cell throughput as cell radius is increased, supposing even user distribution over the cell. It is apparent, that calculating with the distant variable nature of the OF reflects the experience that smaller cells have higher capacity, in contrast with the constant OF resulting in cell size independent capacity (not shown in the Figure). The effect of different usage ratios of different radio bearers is also reflected. The proposed capacity evaluating method is also useful in determining the effect of user distribution in the cell. To capture this effect, besides the even user distribution over the cell (referred to in the following as "even scenario") the so-called "hotspot scenario" is evaluated. In this case user distribution over the plane follows a symmetric two dimensional normal distribution (with independent normally distributed x and y coordinates, with means 0 and equal variances), this is used to model the case when users are mainly placed around the base station.

Figure 2 shows the effect of user distribution: it is apparent that in the hotspot case cell capacity is more than the double of the capacity in the evenly distributed scenario. Bigger capacity of the hotspot scenario is not a surprise, as users generally enjoy higher SIR values as they generally dwell around the base station.

With regards to the planning of the cellular UMTS network, the following statements can be concluded. If customers are generally condensed over some distinct locations (e.g.

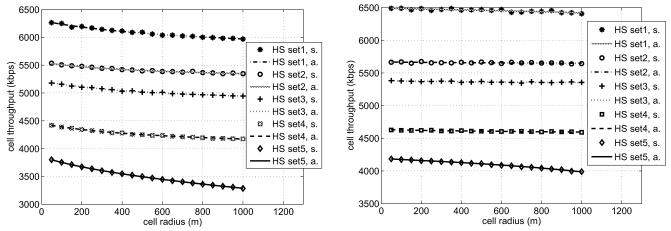


Fig. 3. Average cell throughputs in even (left), hotspot (right) scenarios as function of cell size

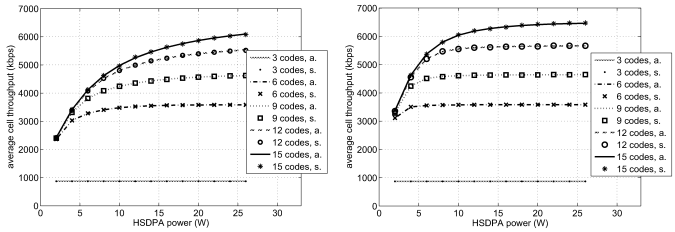


Fig. 4. Average cell throughputs in even (left), hotspot (right) scenarios as function of HSDPA power

residential blocks with rarely visited areas in between), it is worth installing base stations to the center of these locations as this result in significant increment of cell capacity. Cell size (by means of the pilot channel power) can be set to be higher: this allows the fulfillment of coverage requirements with smaller number of base stations!

HSDPA capacity results

This section investigates the capacity of the 3G system, if HSDPA service is deployed. Currently available 3G equipment of the market not always support the dynamic sharing of resources between conventional Release'99 traffic and HSDPA service, but requires the operator to previously configure the available resources (power, number of channelization codes) for the two types of traffic. This facilitates the capacity evaluation of HSDPA, as the effect of existing R'99 traffic can be taken into account by means of the reduced, but constant power level and spreading codes allowed for HSDPA. It is straightforward to account self interference caused by R'99 traffic with the highest power level allowed.

Figure 3 shows the average cell capacity as the function of cell radius, calculated by the proposed analytical method (curves denoted by "a.") and also simulated values (dots denoted by "s."). Results were obtained for the following settings.

Five sets of radio resources (denoted by "HS set1" ... "HS set5") were defined for HSDPA, modeling situations from abundant to scarce HSDPA resources (power changing from 25 to 5 Watts along with the number of codes changing from 15 to 7). Total power of 28 Watts was considered, with constant 3 Watts pilot and signalling channel power, consequently self-interfering power was rising from 3 to 23 Watts in the five sets. Only three types of terminals were considered among the 12 possible types (type 6, 7 and 10, with assumed ratios of 0.5, 0.3 and 0.2 respectively).

It can be stated that as we anticipated it, hotspot scenario results in the highest HSDPA capacity, and even scenario provides the least HSDPA throughput. Capacity is not very sensitive to cell size, except in case of scarce radio resources. Average throughput does not exceed 6.5 Mbps even in case of abundant resources.

Figure 4 reveals average cell throughput in case of 500m cell radius, as the function of HSDPA power. Other settings are as in the previous scenario, except the number of HSDPA codes, that is also a parameter that labels the different curves. It is apparent that in case of hotspot scenario the HSDPA capacity approaches its maximum steeply as HSDPA power is increased, while in concentrated and even scenarios capacity grows less quickly.

CONCLUSIONS

In this paper a fast and efficient method was shown to estimate achievable average throughput of 3G radio interface. The resultant measure is applicable during network dimensioning process. This procedure takes into account multiple bearer types, user distribution, multipath propagation and realistic interference issues.

ACKNOWLEDGEMENT

This work was supported by the Mobile Innovation Center, Hungary.

REFERENCES

[1] K. Sipila, K. C. Honkasalo, J. Laiho-Steffens, and A. Wacker, "Estimation of capacity and required transmission power of WCDMA downlink based on a downlink pole equation," in *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, vol. 2, 2000, pp. 1002–1005.

[2] K. I. Pedersen, T. F. Lootsma, M. Stttrup, F. Frederiksen, T. E. Kolding, and P. E. Mogensen, "Network performance of mixed traffic on high speed downlink packet access and dedicated channels in wcdma," in *Proc. of IEEE Vehicular Technology Conference, VTC 2004 Fall*, September 2004, pp. 4496–4500.

[3] M. Wrulich, W. Weiler, and M. Rupp, "HSDPA performance in a mixed traffic network," in *Proc. of IEEE Vehicular Technology Conference, VTC 2008 Spring*, May 2008, pp. 2056–5060.

[4] K. I. Pedersen and P. E. Mogensen, "The downlink orthogonality factors influence on WCDMA system performance," in *Proceedings of the 56th IEEE Vehicular Technology Conference, VTC 2002-Fall*, vol. 4, 2002, pp. 2061–2065.

[5] F. Brouwer, I. de Bruin, J. Silva, N. Souto, F. Cercas, and A. Correia, "Usage of link-level performance indicators for HSDPA network-level simulations in E-UMTS," *Proceedings of IEEE ISSSTA'04, Sydney*, 2004.

[6] Motorola and Ericsson, "Revised CQI proposal. 3GPP TSG-RAN-WG1 HSDPA, R1-02-0675, April 9-12," 2002.

MODELS AND ALGORITHMS FOR SELF-SIMILAR IP NETWORK TRAFFIC SIMULATION AND PERFORMANCE ANALYSIS

Dimitar Radev
University of Rousse
8 Studentska Street
7017 Rousse, Bulgaria
E-mail: dradev@abv.bg

Izabella Lokshina
SUNY Oneonta
324B Netzer Administration Bldg.
Oneonta, NY 13820, USA
E-mail: lokshiiv@oneonta.edu

KEYWORDS
Modeling and Simulation in Communication Networks, Teletraffic, Stochastic Self-similar Processes, Synthetic Self-Similar Sequence Generators.

ABSTRACT

The paper examines self-similar (or fractal) properties of real communication network traffic data over a wide range of time scales. These self-similar properties are very different from the properties of traditional models based on Poisson and Markov-modulated Poisson processes. Advanced fractal models of sequentional generators and fixed-length sequence generators, and efficient algorithms that are used to simulate self-similar behavior of IP network traffic data are developed and applied. Numerical examples are provided; and simulation results are obtained and analyzed.

INTRODUCTION

The growth of broadband networks and the Internet has been exponential in terms of users and user-end systems as well as in traffic in recent years. High-speed communications networks are able to support a wide range of multimedia applications, such as audio, video and computer data that differ significantly in their traffic characteristics and performance requirements (Giambene, 2005).

One possible goal for telecommunications developers is to build a unified high-speed communication network platform capable of carrying diverse traffic and supporting diverse levels of quality of service (QoS). Recent studies of real communication network traffic data have shown that teletraffic (technical term, identifying all phenomena of control and transport of information within the telecommunications networks) exhibits self-similar (or fractal) properties over a wide range of time scales (Boxma & Cohen, 2005).

The properties of self-similar IP network traffic are very different from properties of traditional models based on Poisson, Markov-modulated Poisson, and related processes (Trivedi, 2001). The use of traditional models in networks characterized by self-similar processes can lead to incorrect conclusions about the performance of analyzed networks (Faraj, 2000). Traditional models can lead to over-estimation of the network performance, insufficient allocation of communication and data processing resources, and consequently difficulties in ensuring the QoS (Rubinstein & Melamed, 1998).

Self-similarity can be classified into two types: deterministic and stochastic (Jeong, 2002). In the first type, deterministic self-similarity, a mathematical object is assumed to be self-similar (or fractal) if it can be decomposed into smaller copies of itself (Park & Willinger, 2000). That is, self-similarity is a property, in which the structure of the whole is contained in its parts.

This work is focused on stochastic self-similarity. In that case, probabilistic properties of self-similar processes remain unchanged or invariant when the process is viewed at different time scales. This is in contrast to Poisson processes that lose their burstiness and flatten out when time scales are changed. However, the time series of self-similar processes exhibit burstiness over a wide range of time scales. Self-similarity can statistically describe teletraffic that is bursty on many time scales.

One can distinguish two types of stochastic self-similarity. A continuous-time stochastic process $\mathbf{Y}_t$ is strictly self-similar with a self-similarily parameter H ($1/2<H<1$), if $\mathbf{Y}_{ct}$ and $c^H\mathbf{Y}_t$ (the rescaled process with time scale ct) have identical finite-dimensional probability for any positive time stretching factor c (Kushner, 2001). This definition, in a sense of probability distribution, is quite different from that of the second-order self-similar process, observed at the mean, variance and autocorrelation levels. The process $\mathbf{X}$ is asymptotically second-order self-similar with $0.5<H<1$, if for each k large enough $\rho_k^{(m)} \rightarrow \rho_k$, as $m\rightarrow\infty$, where $\rho_k=E[(X_i-\mu)(X_{i+k}-\mu)]/\sigma^2$.

In this work the exact or asymptotic self-similar processes are used in an interchangeable manner, which refers to the tail behavior of the autocorrelations (Heyes & Ganesh Babu, 2004). Modeling and simulation of self-similar IP network traffic is performed with advanced generators of synthetic self-similar sequences, divided into two practical classes: the sequential generators and the fixed-length sequence generators. Both classes of generators are considered in this study.

PROPERTIES OF LONG-RANGE DEPENDENT SELF-SIMILAR PROCESSES

In recent communication networks the most striking feature of some second-order self-similar processes is that the accumulative functions of the aggregated processes do

not generate as the non-overlapping batch size m increasing to infinity. Such processes are known as Long-Range Dependent (LRD) processes (Bobbio et al, 2003). This is in contrast to traditional processes used in modeling IP network traffic, all of which include the property that the accumulative functions of their aggregated processes degenerate as the non-overlapping batch size m increasing to infinity, i.e., $\rho_k^{(m)} \to 0$ or $\rho_k^{(m)}=0(|k|>0)$, for $m>1$. The equivalent definition of long-range dependence is given as (1).

$$\sum_{k=-\infty}^{\infty} \rho_k = \infty \tag{1}$$

Another definition of LRD is presented as (2),

$$\rho_k \sim L(t)k^{-(2-2H)}, \quad \text{as} \quad k \to \infty \tag{2}$$

where $1/2<H<1$ and $L(\cdot)$ slowly varies at infinity, i.e. for all $x>0$ it could be determined as (3).

$$\lim_{t\to\infty} \frac{L(xt)}{L(t)} = 1 \tag{3}$$

The Hurst parameter H characterizes the relation in (2), which specifies the form of the tail of the accumulative function. One can show that is true for $1/2<H<1$, as given in (4).

$$\rho_k = \frac{1}{2}\left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right] \tag{4}$$

For $0<H<1/2$ the process is Short Range Dependent (SRD) and could be presented as (5).

$$\sum_{k=-\infty}^{\infty} \rho_k = 0 \tag{5}$$

For $H=1$ all autocorrelation coefficients are equal to one, no matter how far apart in time the sequences are. This case has no practical importance in real telecommunications network traffic modeling. If $H>1$, then (6) is true.

$$\rho_k = \begin{cases} 1 & \text{for} \quad k = 0 \\ \frac{1}{2}k^{2H}g(k^{-1}) & \text{for} \quad k > 0 \end{cases} \tag{6}$$

where

$$g(x) = (1+x)^{2H} - 2 + (1-x)^{2H} \tag{7}$$

One can see that $g(x)\to\infty$ as $H>1$. If $0<H<1$ and $H\neq 1/2$, then the first non-zero term in the Taylor expansion of g(x) is equal to $2H(2H\text{-}1)x^2$. Therefore, (8) is true.

$$\rho_k / (H(2H-1)k^{2H-2}) \to 1, \quad \text{as} \quad k \to \infty \tag{8}$$

In the frequency domain, an essentially equivalent definition of LRD for a process X with given spectral density (9),

$$f(\lambda) = \frac{\sigma^2}{2\pi} \sum_{k=-\infty}^{\infty} \rho_k e^{ik\lambda} \tag{9}$$

is that in the case of LRD processes, this function is required to satisfy the following property (10),

$$f(\lambda) \sim c_{f_1} \lambda^{-\gamma}, \quad \text{as} \quad \lambda \to 0 \tag{10}$$

where c_{f_1} is a positive constant and $0<\gamma<1$, $\gamma = 2H\text{-}1<1$.

As a result, LRD manifests itself in the spectral density that obeys a power-law in the vicinity of the origin. This implies that $f(0)=\Sigma_k \rho_k=\infty$. Consequently, it requires a spectral density, which tends to $+\infty$ as the frequency λ approaches 0.

For a Fractional Gaussian Noise (FGN) process, the spectral density $f(\lambda,H)$ is given by (11),

$$f(\lambda, H) = 2c_f(1-\cos(\lambda))B(\lambda, H) \tag{11}$$

with $0<H<1$ and $-\pi\leq\lambda\leq\pi$, where (12) is true,

$$c_f = \sigma^2(2\pi)^{-1}\sin(\pi H)\Gamma(2H+1)$$
$$B(\lambda, H) = \sum_{k=-\infty}^{\infty} |2\pi k + \lambda|^{-2H-1} \tag{12}$$

and $\sigma^2 = \mathrm{Var}[X_k]$ and $\Gamma(\cdot)$ is the gamma function.

The spectral density $f(\lambda,H)$ in (11) complies with a power-law at the origin, as shown in (13),

$$f(\lambda, H) \to c_f \lambda^{1-2H}, \quad \text{as} \quad \lambda \to 0 \tag{13}$$

where $1/2< H <1$.

SEQUENTIAL GENERATORS OF SELF-SIMILAR IP NETWORK TRAFFIC

We could construct a sequential Markovian model that imitates a self-similar sequence; however this approach would have a disadvantage because of connection between the model parameters and its self-similar properties that is difficult to understand.

Markovian models for self-similar traffic require including several control parameters with a wide range of input values. As a result, controlling these values in sequential generators is much more complex than in generators of fixed-length sequences of self-similar processes with a given Hurst parameter (Stathis & Maglaris, 2000).

We considered the following efficient candidate sequential generators, based on:

- Fractal-Binomial-Noise-Driven Poisson processes;
- Superposition of Fractal Renewal Processes (SFRP);
- M/G/∞ Processes (MGIP);
- Pareto-Modulated Poisson Processes (PMPP);
- Spatial Renewal Processes & Fractional Gaussian Noise;
- Superposition of Autoregressive Processes (SAP).

Model based on Fractal-Binomial-Noise-Driven Poisson Processes

For the standard Fractal Renewal Process (FRP), inter-event times are independent random variables (Bobbio et al, 2003). The marginal Probability Density Function (PDF) of such a fractal renewal process can be defined as (1),

$$f(t) = \begin{cases} 0, & t \leq A, \\ \delta A^{\delta} t^{-(\delta+1)}, & t > A \end{cases} \tag{14}$$

where $0<\delta<2$. Selecting δ in this interval proves far superior to $0<\delta<1$ for the same required values that the inter-event time PDF can be further improved (Borst & Mitra, 1998). The improved PDF of the FRP decays as a power law, as shown in (15),

$$f(t) = \begin{cases} \delta A^{-1} e^{-\delta t/A}, & 0 < t \leq A, \\ \delta e^{-\delta} A t^{-(\delta+1)}, & t > A, \end{cases} \tag{15}$$

which is continuous for all t, producing smoother spectral density function (Radev & Lokshina, 2007). A method based on the Fractal-Binomial-Noise-Driven Poisson Processes (FBNDP) adds M independent and identically distributed (iid) alternating FRPs to generate a fractal binomial noise process that serves as the rate function for a Poisson process. The FBNDP requires five input parameters to generate self-similar sequences: A, δ, R, Δt, and M. The resulting Hurst parameter H assumes the value $(\alpha+1)/2$. The suggested algorithm is advancing with the intervals Δt. If S is a simulation clock, which advances in time, and $S^{(j)}$ is the elapsed time of the j-th FRP sequence, then $S^{(j)}=\tau_0^{(j)} + \tau_1^{(j)} +\ldots+\tau_k^{(j)}$ for some k and j=1,2,…,M, where $\tau_k^{(j)}$ is the inter-arrival time. The sequence of self-similar pseudo-random numbers $X_0,X_1,\ldots$ is generated through the following process.

Algorithm 1:

***Step 1*:** *For each j=1,2,…,M, generate $\tau_0^{(j)}$ from (16),*

$$\tau_0^{(j)} = \begin{cases} -\delta^{-1} A \log[U(\delta^{(V-1)})(\delta^{(V-U)})^{-1}, & V \geq 1, \\ AV^{1/(1-\delta)}, & V < 1, \end{cases} \quad (16)$$

where

$$V \equiv \frac{1+(\delta-1)e^{\delta}}{\delta} U \quad (17)$$

and U is an iid uniformly distributed random variable over the unit interval [0,1); *set $S^{(j)}=\tau_0^{(j)}$.*

***Step 2*:** *Find j^* and $S^{(j)}$ such that j^*=argmin$_j\{S^{(j)}\}$.*

***Step 3*:** *Calculate (18).*

$$x = \begin{cases} 0, & if \quad S^{(j^*)} < A, \\ 1, & if \quad S^{(j^*)} \geq A. \end{cases} \quad (18)$$

***Step 4*:** *If x=1, then X_0 should be drawn from a Poisson probability distribution with λ=1. If x=0, then X_0=0.*

***Step 5*:** *Set i=1, and y=0. Advance the simulation clock, i.e., $S \leftarrow S^{(j^*)}$.*

***Step 6*:** *Construct a new inter-event time $\tau_i^{(j)}$ from (19),*

$$\tau_i^{(j)} = \begin{cases} -\delta^{-1} A \log[U], & U \geq e^{-\delta}, \\ e^{-1} A U^{-1/\delta}, & U < e^{-\delta}, \end{cases} \quad (19)$$

and set $S^{(j^*)} \leftarrow S^{(j^*)}+\tau_i^{(j)}$.

***Step 7*:** *Find a new j^* such that j^*= argmin$_j\{S^{(j)}\}$, and compute $S^{(j)}$-S.*

***Step 8*:** *Repeat Step 6 through Step 8 to obtain x as in (18).*

***Step 9*:** *Advance the simulation clock, i.e., $S \leftarrow S^{(j^*)}$, and set $y=y+x$.*

***Step 10*:** *Repeat Step 6 through step 10 within time slot of length Δt.*

***Step 11*:** *Compute X_i=POISS(y), set y=0, and $i=i+1$.*

***Step 12*:** *Repeat Step 6 through step 11 until $i=n$, where n is the number of sample points.*

The self-similar traffic sequence $\{X_1,X_2,\ldots\}$ of 1,048,576 numbers was generated in 1 min. 38 sec. on the Centrino-based HP PC (1.76 GHz) using the FBNDR algorithm.

Model based on Superposition of Fractal Renewal Processes

A method based on a Superposition of the Fractal Renewal Processes (SFRP) uses a group of independent and identical fractal renewal processes (FRP). This method requires three parameters, i.e., α and A from the individual FRPs, and M, which is the number of FRPs superposed. The resulting Hurst parameter H, and the mean μ and the variance σ^2 of the marginal output distribution in related counting process during the unit time interval, can be defined according to (20):

$$\begin{aligned} H &= (\alpha+1)/2 \\ \mu &= \mathrm{E}[X_n] = \lambda \\ \sigma^2 &= Var[X_n] = (1+(1/t_0))^{\alpha}\lambda \end{aligned} \quad (20)$$

where

$$\lambda = M\delta[1+(\delta-1)^{-1}e^{-\delta}]^{-1}A^{-1} \quad (21)$$

is an aggregate arrival rate of events in the unit time interval. The sequence of self-similar pseudo-random numbers $X_0,X_1,\ldots$ is generated with the following procedure.

Algorithm 2:

***Step 1*:** *For each j=1,2,…,M, and i=0, generate $\tau_0^{(j)}$ from equations (3) and (4); set $S^{(j)}=\tau_0^{(j)}$.*

***Step 2*:** *Find j^* such that j^*=argmin$_j\{S^{(j)}\}$ and set $X_0=S^{(j^*)}$.*

***Step 3*:** *Advance the simulation clock, i.e. $S \leftarrow S^{(j^*)}$.*

***Step 4*:** *Set $i=i+1$. Construct a new inter-event time $\tau_i^{(j)}$ from equation (6) and set $S^{(j^*)} \leftarrow S^{(j^*)}+\tau_i^{(j)}$.*

***Step 5*:** *Find a new j^* such that j^*= argmin$_j\{S^{(j)}\}$, and compute $S^{(j)}$-S.*

***Step 6*:** *Advance the simulation clock, i.e. $S \leftarrow S^{(j^*)}$.*

***Step 7*:** *Repeat Step 4 through Step 6 until $i=n$ is reached, where n is the number of sample points.*

The self-similar traffic sequence $\{X_1,X_2,\ldots\}$ of 1,048,576 numbers was generated in 1 min. 12 sec. on the Centrino-based HP PC (1.76 GHz) using the SFRP algorithm.

Model based on Superposition of Autoregressive Processes

A method based on the superposition of autoregressive processes (SAP) generates asymptotically self-similar sequences when aggregating several independent autoregressive processes. In the simplest case this can be presented as the sum of two autoregressive processes of the first order, as shown in (22):

$$\begin{aligned} z_{1i} &= A_{1i} z_{1,i-1} + y_{1i} \\ z_{2i} &= A_{2i} z_{2,i-1} + y_{2i}, \quad i = 1,2,... \end{aligned} \quad (22)$$

where A_{1i} and A_{2i} are randomly chosen from a beta-distribution $B(\alpha_1,\alpha_2)$ on [0,1] with the shape parameters α_1 and α_2, where $\alpha_1>0$, $\alpha_2>0$; y_{1i} and y_{2i} are a pair of independent and identically distributed sequences of random variables with mean of zero and variance $\sigma^2=1$.

Using the least-square fitting we can find that α_2=7.7929∗log(H)+4.9513. Then, the Hurst parameter H is linearly dependent on the shape parameter α_2 of a beta-distribution, while α_1 can be selected arbitrary; for example, α_1=1 in all cases that are investigated.

The PDF $f(x)$ of the beta-distribution is defined as (23),

$$f(x)=\begin{cases}\dfrac{x^{\alpha_1-1}(1-x)^{\alpha_2-1}}{\beta(\alpha_1,\alpha_2)} & 0<x<1\\ 0, & \text{otherwise}\end{cases} \quad (23)$$

where $\beta(\alpha_1,\alpha_2)$ is given as (24).

$$\beta(\alpha_1,\alpha_2)=\int_0^1 x^{\alpha_1-1}(1-x)^{\alpha_2-1}dx=\frac{\Gamma(\alpha_1)\Gamma(\alpha_2)}{\Gamma(\alpha_1+\alpha_2)} \quad (24)$$

This method, also based on a superposition of the autoregressive processes, consists of the following steps.

Algorithm 3:

***Step 1*:** *Set i=i+1. Determine* z_{1i} *and* z_{2i} *using* (*9*).

***Step 2*:** *Calculate the sum* $X_i = z_{1i} + z_{2i}$, $i=1,2,\ldots..$

***Step 3*:** *Repeat Step 1 and Step 2 until i=n, where n is the number of sample points.*

The self-similar traffic sequence $\{X_1,X_2,\ldots\}$ of 1,048,576 numbers was generated in 3.5 seconds on the Centrino-based HP PC (1.76 GHz), using the SAP algorithm.

Performance Analysis

Unlike other sequential generators, the SAP generator does not require an aggregation level to be assumed as an input parameter. But it requires the shape parameter α_1 as input parameter. Mean values of estimated *H* obtained with algorithms for sequential generators are shown in Table 1 and Table 2, respectively, for Hurst parameter, equal to 0.6, 0.7 and 0.8, 0.9.

Table 1: Accuracy of Generated Sequences (H = 0.6; 0.7)

Methods	Mean Values of Estimated *H* and Δ*H*			
	0.6		0.7	
	$\hat{H}$	Δ*H*	$\hat{H}$	Δ*H*
FBNDP	0.6086 (0.581, 0.636)	+1.440	0.6875 (0.660, 0.715)	-1.789
SFRP	0.6166 (0.589, 0.644)	+2.759	0.7091 (0.682, 0.737)	+1.296
SAP	0.5989 (0.595, 0.603)	-0.182	0.6852 (0.680, 0.690)	-2.112

Table 2: Accuracy of Generated Sequences (H = 0.8; 0.9)

Methods	Mean Values of Estimated *H* and Δ*H*			
	0.8		0.9	
	$\hat{H}$	Δ*H*	$\hat{H}$	Δ*H*
FBNDP	0.7827 (0.755, 0.810)	-2.157	0.8502 (0.823, 0.878)	-5.538
SFRP	0.7986 (0.771, 0.826)	-0.174	0.8686 (0.841, 0.896)	-3.491
SAP	0.7845 (0.781, 0.788)	-1.937	0.8971 (0.894, 0.900)	-0.325

The cumulative relative error for the FBNDP method was less then 5%, except for *H*=0.9. For the SEFR method the cumulative relative error was +6.21%, +1.78%, -1.43% and -5.37%, for H=0.6, 0.7, 0.8 and 0.9, respectively. Similar to the FBNDP method, estimated *H* values ranged from positively biased to negatively biased, as the *H* value increased. For α_2=2.9 and 8.1 all values of the Hurst parameter from the traffic sample sequence of the SAP method were higher than the required values. The relative error was +24.14% and +10.45%, respectively; in that case, the results were overestimated. For α_2=21.3 and 71.5, the relative error was -0.01% and -8.45%.

FIXED-LENGTH SEQUENCE GENERATORS OF SELF-SIMILAR IP NETWORK TRAFFIC

The most frequently studied discrete-time models of self-similar IP network traffic belong to fractional autoregressive integrated moving-average (F-ARIMA) and fractional Gaussian noise (FGN) processes, and possible fixed-length sequence generators are based on:

- Fractional autoregressive integrated moving-average processes (F-ARIMA);
- Fast Fourier transform (FFT);
- Fractional Gaussian noise and Daubechies wavelets;
- Random midpoint displacement (RMD);
- Successive random additions.

Model based on Fractional Autoregressive Integrated Moving-Average Processes

Let us consider F-ARIMA(0,*d*,0) method for generating self-similar sequences, where *d* is the fractional differencing parameter, 0<*d*<1/2, and let generate the process $\mathbf{X}=\{X_i: i=0,1,2,\ldots,n\}$ with a normal marginal distribution, the mean of zero and the variance σ_0^2, and the autocorrelation function (ACF), $\{\rho_k\}$ (k=0,±1,…) that defined as (25),

$$\rho_k=\gamma_k/\gamma_0=\frac{\Gamma(1-d)\Gamma(k+d)}{\Gamma(d)\Gamma(k+1-d)}, \quad (25)$$

where

$$\gamma_k=\sigma_0^2\frac{(-1)^k\Gamma(1-2d)}{\Gamma(k-d+1)\Gamma(1-k-d)}$$

The fixed-length sequence of self-similar pseudo-random numbers is generated through the following process.

Algorithm 4:

***Step 0*:** *Set* $N_0=0$ *and* $D_0=1$. X_0, *the firs pseudo random element in the output self-similar sequence, is generated from the normal distribution* $N(0,\sigma_0^2)$, *where* σ_0^2 *is the required variance of the* X_i.

***Step i*:** ($i=1,\ldots,n-1$). *Compute* $mean_i$ *and* var_i *of* X_i *recursively, using the following equations:*

$$N_i=\rho_i-\sum_{j=1}^{i-1}\phi_{i-1,j}\rho_{i-j},$$

$$D_i=D_{i-1}-N_{i-1}^2/D_{i-1},$$

$$\phi_{ii}=N_i/D_i,$$

$$\phi_{ij}=\phi_{i-1,j}-\phi_{ii}\phi_{i-1,i-j}, \quad j=1,\ldots,i-1,$$

where ϕ_{ij}, $i=0$, $j=0,\ldots,n-1$, is given by

$$\phi_{ii}=-\binom{i}{j}\frac{(j-d-1)!(i-d-j)!}{(-d-1)!(i-d)!},$$

$$mean_i=\sum_{j=1}^{i}\phi_{ij}X_{i-j},$$

$$\mathrm{var}_i=(1-\phi_{ii}^2)\,\mathrm{var}_{i-1},$$

Generate X_i *from* $N(mean_i, var_i)$. *Increase i by* 1. *If i=n, then stop.*

The self-similar traffic sequence $\{X_1,X_2,\ldots\}$ of 1,048,576 numbers was generated in 4 hours, 1 min. and 24 sec. on the Centrino-based HP PC (1.76 GHz) using the F-ARIMA algorithm. Therefore, this algorithm is too rigorous to be used to generate long sample sequences (Radev & Lokshina, 2009).

Model based on Fractional Gaussian Noise and Daubechies Wavelets

In this work we propose a new advanced generator of pseudo-random self-similar sequence based on Fractional Gaussian Noise (FGN) and Daubechies Wavelet (DW), called the FGN-DW method. The generators based on the DW produce more accurate self-similar sequences than those based on Haar Wavelets (HW) (Daubechies, 1992). In that case not only estimates of H obtained from the DW are closer to the true values than those from the HW, but also variances obtained from the DW are much lower. The reason behind is that the DW produce smoother wavelet coefficients that are used in the Discrete Wavelet Transform (DWT), than the HW. The HW are discontinuous, and they do not have good time-frequency localisation properties, since their Fourier transforms decay as $|\lambda|^{-1}$, $\lambda\to\infty$, meaning that resulting decomposition has a poor scale. Therefore, the DW produce more accurate coefficients than the HW. The suggested approach for generating synthetic self-similar FGN sequences in a time domain is based on a DWT. Wavelets can provide compact representations for a class of FGN processes, because the structure of wavelets naturally matches the self-similar structure of the LRD processes.

This paper confirms that the FGN-DW method is sufficiently fast being used for practical generation of synthetic self-similar sequences as simulation input data. The wavelet analysis transforms a sequence onto a time-scale grid, where the term scale is used instead of frequency, because the mapping is not directly related to frequency as in the Fourier transform. The wavelet transform generates the wavelet coefficients $d_x(i,j)$ from a sequence of given numbers. For an LRD process, the variance of the wavelet coefficients at each level i is defined by (26),

$$E[d_x(j,\cdot)^2 = CP_f 2^{i(2H-1)} \tag{26}$$

where $C>0$ and (18) is true,

$$P_f = 2(2\pi)^{1-2H} c_\gamma E(2H-1)\sin((1-H)\pi) \tag{27}$$

where $E(\cdot)$ is the Euler function and c_γ is a positive constant. The power parameter P_f that plays a major role in fixing the absolute size of the LRD process and generates effects in the applications is an independent quantitative parameter with the dimensions of variance.

A spectral estimator can be obtained though calculating time average Q_i of the $d_x(i,\cdot)$ at a given scale that could be described as (28),

$$Q_i = \frac{1}{n_i}\sum_{j=1}^{n_i} d_x^2(i,j), \tag{28}$$

where n_i is the number of wavelet coefficients at scale i, i.g. $n_i=2^{-i}n$, where n is the number of data points. The estimator uses a weighted linear regression as the variance of $\log_2(Q_i)$ vary with $\log_2(2^i)$. An estimated Hurst parameter $\hat{H}$ could be obtained with the linear regression (29),

$$y_i = \log_2(Q_i) = (2\hat{H}-1)i + c + 1/(n_i \ln 2) \tag{29}$$

where c is a constant.

The wavelet transform delivers good resolution in both time and scale, as compared to the Fourier transform, which provides only good frequency resolution. The fixed-length sequence of self-similar pseudo-random numbers can be generated through the recently developed process that consists of the following steps.

Algorithm 5:

***Step 1*:** *Given: H, Start for i=1 and continue until i=n. calculate a sequence of values* $\{f_1,f_2,\ldots,f_n\}$ *using* (30),

$$f(\lambda,H) = c_f\,|\lambda|^{1-2H} + O(|\lambda|^{\min(3-2H,2)}), \tag{30}$$

where $c_f = \sigma^{2\cdot}(2\pi)^{-1}\sin(\pi H)\Gamma(2H+1)$, $O(\cdot)$ *represents the residual error and* $f_i = \hat{f}(\pi i/n;H)$,

the value of the frequencies f_i *corresponds to the spectral density of an FGN process for* f_i *ranging between* $\pi/n \div \pi$.

***Step 2*:** *Multiply* $\{f_i\}$ *by realizations of the independent exponential random variable with the mean of one to obtain* $\{\hat{f}_i\}$, *because the spectral density estimated for a given frequency is distributed asymptotically as the independent exponential random variable with the mean* $f(\lambda,H)$.

***Step 3*:** *Generate a sequence* $\{Y_1,Y_2,\ldots,Y_n\}$ *of complex numbers such that* $|Y_i| = \sqrt{\hat{f}_i}$ *and the phase of* Y_i *is uniformly distributed between 0 and* 2π. *This random phase technique preserves the spectral density corresponding to* $\{\hat{f}_i\}$. *It also makes the marginal distribution of the final sequence normal and produces the requirements for FGN.*

***Step 4*:** *Calculate two synthetic coefficients of orthonormal Daubechies wavelets. The output sequence* $\{X_1,X_2,\ldots,X_n\}$ *representing approximately self-similar FGN process is obtained by applying the inverse Daubechies wavelets transformation of the sequence* $\{Y_1,Y_2,\ldots,Y_n\}$.

It took only 2 seconds on Centrino-based HP PC (1.76 GHz), to generate a sequence of 1,048,576 numbers using this outstanding algorithm.

Performance Analysis

The Hurst parameter estimates for the wavelet-based H estimator are shown in Table 3 (H=0.6 and 0.7) and Table 4 (H=0.8 and 0.9) with different wavelet coefficients. The obtained results are averaged over 30 sequences, and they show that for all input H values the F-ARIMA and the FGN-DW methods produce sequences with less biased H values than other methods.

Table 3: Accuracy of Sequences Generated with FGN-DW (H = 0.6; 0.7)

Wave-let coeff.	Mean Values of Estimated H and ΔH			
	0.6		0.7	
	$\hat{H}$	ΔH	$\hat{H}$	ΔH
2	0.6019 (0.574, .630)	+0.323	0.6984 (0.671, 0.726)	-0.229
4	0.6026 (0.575, 0.630)	+0.433	0.7039 (0.676, 0.731)	+0.554
8	0.6026 (0.575, 0.630)	+0.430	0.7031 (0.676, 0.731)	+0.445
16	0.6013 (0.574, 0.629)	+0.214	0.6987 (0.671, 0.726)	-0.185

Table 4: Accuracy of Sequences Generated with FGN-DW (H = 0.8; 0.9)

Wave-let coeff.	Mean Values of Estimated H and ΔH			
	0.8		0.9	
	$\hat{H}$	ΔH	$\hat{H}$	ΔH
2	0.7943 (0.767, 0.822)	-0.709	0.8898 (0.862, 0.917)	-1.137
4	0.8055 (0.778, 0.833)	+0.684	0.9074 (0.880, 0.935)	+0.821
8	0.8039 (0.776, 0.831)	+0.486	0.9049 (0.877, 0.932)	+0.545
16	0.7962 (0.769, 0.824)	-0.474	0.8938 (0.866, 0.921)	-0.694

The means of estimated H values, obtained with the suggested wavelet-based algorithm, are shown in Table 5.

Table 5: Variances of Hurst Parameter

Wave-let coeff.	Mean Values of Estimated H and ΔH			
	0.6	0.7	0.8	0.9
2	1.949e-04	2.154e-04	2.393e-04	2.670e-04
4	2.508e-04	2.540e-04	2.586e-04	2.645e-04
8	2.440e-04	2.432e-04	2.438e-04	2.456e-04
16	2.116e-04	2.055e-04	2.007e-04	1.973e-04

Analysis of the mean times required to generate sequence of a given length demonstrates that the sequential generators are more attractive for the practical simulation studies of computer networks than the F-ARIMA-based generator, since they are much faster. However, these generators require more input parameters, and selecting appropriate values is a problem that remains. In addition, the problem of how to define the relationship between the Hurst parameter and two shape parameters of a beta-distribution in the case of SAP also remains.

CONCLUSIONS

Communication network traffic exhibits self-similar properties over a wide range of time scales that are very different from the properties of traditional models. Self-similar models provide capacity to estimate the network performance and quality, allocate the resources, and ensure the QoS. One of the problems that telecommunications researchers face during simulation is how to generate long synthetic sequential self-similar sequences. To solve the problem three aspects must be considered: how accurately self-similar process can be generated; how quickly the methods generate long self-similar sequences, and how appropriately self-similar processes can be used in sequential simulation.

REFERENCES

Bobbio, A., Horváth, A., Scarpa, M., Telek, M. (2003). Acyclic discrete phase type distributions: Properties and a parameter estimation algorithm. - Performance Evaluation, **54**(1), 1-32.

Borst, S.C., Mitra, D. (1998). Virtual Partitioning for Robust Resource Sharing: Computational Techniques for Heterogeneous Traffic. - *IEEE Journal on Selected Areas in Communications*, **16**, (5), 668–678.

Boxma, O., Cohen, J. (2000). Self-Similar Network Traffic and Performance Evaluation. John Wiley & Sons, New York.

Daubechies, I. (1992) *Ten Lectures on Wavelets, vol.61 of CBMS-NFS Regional Conference Series in Applied Mathematics*. SIAM Press, Philadelphia, Pennsylvania.

Faraj. R. (2000). Modeling and analysis of self-similar traffic in ATM networks. PhD thesis.

Giambene,G.(2005). Queueing Theory and Telecommunications: Networks and Applications. Springer, NY. 585 p.

Hayes, J., Ganesh Babu, T. (2004), *Modeling and Analysis of Telecommunications Networks*, NY, John Wiley and Sons.

Jeong, H.-D. (2002). Modelling of Self-Similar Teletraffic for Simulation. PhD Thesis. University of Canterbury. USA.

Kushner. H., (2001). Heavy Traffic Analysis of Controlled Queueing and Communications Networks. Springer, NY.

Mehdi, J. (2003), *Stochastic Models in Queueing Theory*, Academic Press.

Park, K., Willinger, W. (2000). Self-Similar Network Traffic and Performance Evaluation. John Wiley & Sons. New York.

Radev, D., Lokshina, I. (2007). Modeling and Simulation of Self-Similar Teletraffic, *Proceedings of the Industrial Simulation Conference (ISC) 2007*, Delft, The Netherlands, 355-359.

Radev, D., Lokshina, I. (2009). Modeling and Simulation of Self-Similar Wireless IP Network Traffic, *Proceedings of the IEEE Wireless Telecommunications Symposium (WTS) 2009*, Prague, Czech Republic, 1-6.

Rubinstein, R., Melamed, B., (1998), *Modern Simulation and Modeling,* New York, John Wiley & Sons.

Stathis, C., Maglaris, B. (2000). Modeling the Self-Similar Behaviour of Network Traffic. *Journal of Computer Networks*, **34** (1): 37 – 47.

Trivedi, K. (2001), *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*, New York, John Willey & Sons.

BIOGRAPHY

DIMITAR RADEV, D.Sc. is Professor at Department of Communication Technique and Technologies at University of Rousse, Bulgaria. His main research interests are Teletraffic Theory, Modeling and Simulation in Communication Networks, and Performance Analysis of Queuing Systems; dradev@abv.bg

IZABELLA LOKSHINA, PhD is Associate Professor of MIS and chair of Management, Marketing and Information Systems Department at SUNY Oneonta, USA. Her main research interests are Intelligent Information Systems, and Modeling and Simulation in Communication Networks; lokshiiv@oneonta.edu

PERFORMANCE PREDICTION OF CONSERVATIVE PARALLEL DISCRETE EVENT SIMULATION

Gábor Lencse
Széchenyi István University
Egyetem tér 1.
H-9026 Győr, Hungary
lencse@sze.hu

András Varga
Opensim Ltd.
Szőlő köz 11.
H-1032 Budapest, Hungary
andras@omnetpp.org

KEYWORDS

performance analysis, parallel discrete event simulation, conservative synchronisation method, speed-up, OMNeT++

ABSTRACT

In a previous paper, a quantitative criterion has been proposed for efficient execution of the Null Message Protocol, the best- known conservative parallel discrete event simulation (PDES) protocol. The criterion is based on a novel concept of the coupling factor, and allows one to use intuitive and easy-to-measure input parameters. The criterion can be used to assess simulation models' potential for parallel execution as well as the maximum partitioning that may still potentially yield good performance.

This paper deals with experimental verification of the proposed rule that a large coupling factor is a necessary precondition for getting a good speed-up with conservative parallel simulation. A closed queueing network is used as simulation model, and it is run on up to 24 CPU cores. It is shown that the criterion provides a quick and convenient way to determine whether a simulation model has a potential for speed-up before actually investing work in the parallelization.

INTRODUCTION

Because of the rapid development and growth of information and communication technology (ICT) systems in the last decades, the analysis of their performance is an important issue. Event-driven ***discrete event simulation*** (DES) is a powerful tool for this task. Nevertheless, the systems are often so large and complex and the number of events is so high that the execution requires unacceptably long time even using a supercomputer. Let us review first the solutions that were proposed for this problem so far.

Parallelisation can be a natural solution. However, it is not an easy task and the achievable speed-up is often limited. The reason is the algorithm of the event-driven DES. When doing ***parallel discrete event simulation*** (PDES), the model of the system is divided into partitions (called Logical Processes), and the partitions are assigned to processors that are executing them. To maintain causality, the virtual times of the partitions must be synchronised. There are three different methods for synchronisation. The first two are described in (Fujimoto 1990).

The ***conservative method*** ensures that causality is never violated. An event can be executed only if we are certain that no events with smaller timestamp exist (and also will not be generated) anywhere in the model. Unless the simulated system has a special property that the so called *look-ahead* is large enough, the processors executing the partitions need to wait for each other in the majority of time, so the achievable speed-up is poor.

The ***optimistic method*** allows and detects causality errors and uses *roll-backs* to recover from them. However, there are some problems associated with roll-backs. First, the ability of the roll-backs requires extra functionality (e.g. state saving and restoration) from both the simulation kernel and the models[1]. Even though a simulator may support optimistic parallel execution, model authors may not be willing to invest the extra work in the model building that is needed to prepare the model for the optimistic method. Second, above a certain number of processors the resource consumption of the roll-backs may hinder the good speed-up and the algorithm may not scale well. Third, state saving and restoration itself has CPU and memory costs, decreasing the gain from parallel execution.

The ***statistical synchronisation method*** (SSM) has been proposed by György Pongor (Pongor 1992). SSM does not exchange individual messages between the partitions, but rather the statistical characteristics of the message flow. In its original form, SSM was applicable for the analysis of steady state behaviour of systems. It was further developed (as SSM-T) by Gábor Lencse (Lencse 1998). The method can produce excellent speed-up, but has a limited area of application (Lencse 1999). For more information about the three methods, see (Lencse 2002) and its references.

In the paper (Varga et al. 2003), the authors propose a method for assessing available parallelism in a simulation model for conservative synchronization. The method requires only a small number of parameters that can be easily measured on a sequential simulation.

---

[1] Assuming that C or C++, the usual languages for high-performance simulations, are used to implement the models. Models written in idiomatic C/C++ cannot be automatically save/restored by the simulation kernel.

In this paper, we intend to check the results of the aforementioned work for a higher number of processors, and also examine how different parameters of the model influence achievable speed-up.

The remainder of this paper is organised as follows: first, we briefly summarize the method for assessing the available parallelism: we describe the necessary parameters and present the method; then we describe the hardware and software environment of our experiments; next, we repeat some of the experiments of Varga et al. for higher number of processors; then we examine how the speed-up depends on some of the parameters; finally, we present our conclusions.

This topic was identified as being of importance in the parallel simulation of large systems.

ASSESSING AVAILABLE PARALLELISM

The paper (Varga et. al. 2003) uses the notations *ev* for events, *sec* for real world time in seconds and *simsec* for simulated time (model time) in seconds. The paper uses the following quantities for the assessing of available parallelism:

- *P performance* represents the number of events processed per second (*ev*/*sec*). *P* depends on the performance of the hardware and the amount of computation required for processing an event. *P* is independent of the size of the model.
- *E event density* is the number of events that occur per simulated second (*ev*/*simsec*). *E* depends on the model only, and not on the hardware and software environment used to execute the model. *E* is determined by the size, the detail level and also the nature of the simulated system.
- *R relative speed* measures the simulation time advancement per second (*simsec*/*sec*).
  Note that $R = P/E$.
- *L lookahead* is measured in simulated seconds (*simsec*). When simulating telecommunication networks and using link delays as lookahead, *L* is typically in the *microsimsec–millisimsec* range.
- τ *latency* (sec) is the latency of sending a message from one Logical Process (LP) to another. This value is usually in the µs-ms range, and is largely determined by the hardware and software on which the simulation runs.
- λ *coupling factor* can be calculated as the ratio of LE and τP:

$$\lambda = \frac{L \cdot E}{\tau \cdot P} \qquad (1)$$

The paper (Varga et. al. 2003) states that the chance of the good speed-up of the PDES using the *conservative synchronisation method* can be predicted on the basis of the magnitude of λ. The work supports its theoretical result by experiments performed on clusters of 2 or 4 (single core) CPUs. The aim of this paper is to verify those results by experiments performed on larger clusters up to 12 dual core CPUs (that is, 24 cores).

HARDWARE AND SOFTWARE ENVIRONMENT

For our experiments, we used a cluster of 12 PCs with *AMD Athlon 64 X2 Dual Core 4200+* processor, *2*1GB DDR2 667MHz* (dual channel) RAM and *NVIDIA nForce® 500 SLI™ MCP built-in Gigabit Ethernet* NIC. The hosts were interconnected by a *3Com 2948-SFP* Gigabit Ethernet switch.

The hosts were running Debian Squeeze GNU/Linux operating system and LAM/MPI 7.2.1 cluster software.

The communication latency between the hosts (the PCs of the cluster) over MPI was about *25µs*.

We have created user accounts called "mpi" on all the hosts. Its home directory was stored on a *SUN Fire X4200 M2* NFS server connected by the same Gigabit Ethernet switch.

We used private IP addresses, and there was no other load on the cluster and NFS server during the experiments.

We used OMNeT++ version 4.0p1.

THE SIMULATION MODEL

For the experiments, we used the Parallel CQN (Closed Queueing Network) simulation sample program of OMNeT++ – the same model that the original work used. This model consists of *M* tandem queues where each tandem consists of a switch and *k* single-server queues with exponential service times (Figure 1).

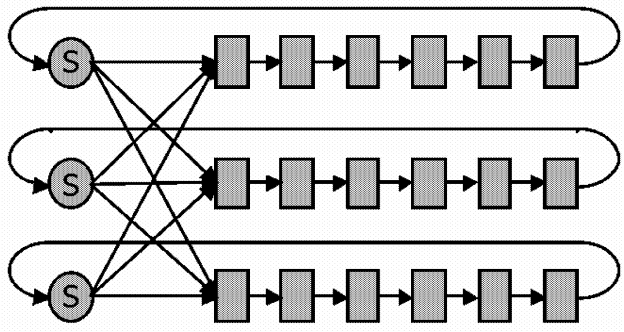


Figure 1. *M*=3 Tandem Queues with *k*=6 Single Server Queues in Each Tandem Queue

The last queues are looped back to their switches. Each switch randomly chooses the first queue of one of the tandems as destination, using uniform distribution. The queues and switches are connected with links that have nonzero propagation delays. The OMNeT++ model for CQN wraps tandems into compound modules.

To run the model in parallel, we assign tandems to different LPs (Figure 2.). Lookahead is provided by delays on the marked links.

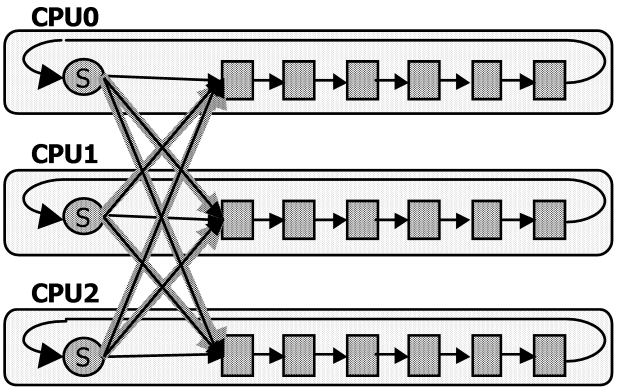


Figure 2. Partitioning the CQN Model

As for the parameters of the model, we have used the preset values shipped with the model. We chose configuration B, the one that promised good speed-up. The main parameters were: M=24 tandem queues, k=50 queues in each tandem queue, exponential service time of the queues with expected value of 10 seconds. As for the delay between the tandem queues, first we used the preset value of L=100 seconds, and later we used this parameter to achieve the values of λ, which we were interested in.

THE EXPERIMENTS

First, we ran a single processor simulation with L=100s to measure the E and P variables necessary for the calculation of λ. It is a very convenient feature of the OMNeT++ simulator that it displays these variables both in its GUI and command line user interfaces. We used the latter for the measurement, because at the time of the parallel simulation we needed to use the command line environment to facilitate convenient experimenting. For the value of λ, we got:

$$\lambda = \frac{L \cdot E}{\tau \cdot P} = \frac{100 \cdot 156}{25 \cdot 10^{-6} \cdot 250000} \approx 2500 \qquad (2)$$

This is an appropriate value to expect good speed-up according to (Varga et. al. 2003). The value of λ decreases with the number of LPs. If we use N number of LPs, then:

$$\lambda_N = \frac{\lambda}{N} \qquad (3)$$

We performed a series of experiments for the following values of N: 1, 2, 4, 6, 8, 12 and 24. N=1 means that there was only a single LP. From 2 to 24, we included N/2 number of dual core hosts in the MPI virtual machine, each running 2 LP's. Each LP contained M/N tandem queues. (E.g. N=6 means that there were 6 LP's executed by the 6 cores of 3 hosts, and each LP contained 4 tandem queues.)

With a few exceptions, we executed every experiment n=11 times, and calculated average and standard deviation for the t_i execution times.

Initial Results

Our first results are shown in Table 1.

Table 1. Execution Time and Speed-up in the Function of the Number of Logical Processes with L=100s Lookahead

number of LP's	1	2	4	6	8	12	24
avg. ex. time [sec]	516.55	291.27	140.00	94.09	72.27	50.18	33.91
exec time std. dev.	5.47	11.31	7.46	1.14	1.62	0.60	0.30
speed-up	1.00	1.77	3.69	5.49	7.15	10.29	15.23
relative speed-up	1.00	0.89	0.92	0.91	0.89	0.86	0.63

Note that except the case of N=24 LPs, the value of λ_N is always higher than 200, and there is an excellent speed-up. λ_{24}=104 corresponds to an acceptable but significantly lower speed-up (the relative speed-up is only 0.63).

To examine the effect of λ being one order of magnitude smaller, we carried out the same series of experiments for L=10s, too.

Table 2. Execution Time and Speed-up in the Function of the Number of Logical Processes with L=10s Lookahead

number of LP's	1	2	4	6	8	12	24
avg. ex. time [sec]	523.09	307.64	157.73	14.64	99.27	91.45	119.27
exec time std.	11.78	20.72	1.19	.9633	2.57	0.69	0.79
speed-up	1.00	1.70	3.32	4.56	5.27	5.72	4.39
relative speed-up	1.00	0.85	0.83	0.76	0.66	0.48	0.18

Table 2 shows the results for an L lookahead 10 times smaller than in Table 1. The changed lookahead also means that the values of λ_N are also approximately[2] 10 times smaller. For N=4 LPs, the calculated value of λ_4 =2500/10/4=62.5 still results in a 3.32 times speed-up and a relative speed-up of 0.83. For the values of N=6, 8 and 12, the relative speed-up is visibly degrading, and for N=24 it becomes poor: 0.18. (The execution time *increases* if we use 24 LPs instead of 12 LPs.)

Vacationing Jobs and their Influence on λ

Given that our simulation model is relatively simple, it would be tempting to think that its performance characteristics (P and E) do not depend on the L lookahead, and so λ is directly proportional to L. However, this assumption is incorrect due to the phenomenon of "vacationing jobs", explained below. The consequence is that λ must be calculated independently from P and E measured for each value of L.

[2] But not exactly, see explanation later.

If we compare the execution times of the sequential simulations we can observe a slight difference: it is 523s and 517s for the lookahead values of 10s and 100s, respectively. The difference is even more radical for L=1000s; then the execution time is 416s. The explanation is that as L increases, a higher proportion of the jobs will be "buffered" in the long-delay links among the tandems, that is, they are effectively removed from the queueing inside the tandems. This phenomenon results in a lower load for higher values of L. Let us refer to these jobs as "vacationing" jobs.

Our model is a closed queueing network, so the number of the jobs in the system is constant; in our model this number is always 2400, as there are 24 tandems, 50 queues in all tandems, and initially 2 jobs are placed in each queue. When the switch makes a decision about a job, with the probability of 23/24≈95.8% the job will be sent to a different tandem, that is, the job will be vacationing for L simulated seconds before it arrives at the first queue of the destination tandem. A tandem consists of k=50 queues, and each of them has an exponential service time with a mean of 10s and an output latency of 1s (modelled as link delay between adjacent queues). As a rough estimation, we can say that at the beginning of the simulation (when the jobs are distributed evenly and there are no vacationing jobs) the complete travel of a job through the tandem will last:

$$50*(10s+10s+1s)=1050s \quad (4)$$

This is an estimated service time of the tandem, valid only at the beginning of the simulation. For L=100s and 1000s, the vacationing time is comparable to the roughly estimated service time of the tandem. As a very rough[3] approximation, we can say that 10% of the jobs are vacationing for the L=100s lookahead. Because of the capabilities of the OMNeT++ simulator, the number of vacationing jobs can be easily determined. The jobs are represented by messages in OMNeT++, and the messages representing vacationing jobs are residing in the Future Event Set (FES). The number of the events in the FES is displayed in the GUI of the simulator, and even the contents of the FES can be interactively examined during the simulation. We have found that on average, 245 messages representing jobs (≈10%) were present in the FES in the case of the sequential simulation with L=100s lookahead.

The consequence of vacationing jobs is that the value of λ cannot be calculated for the different lookahead values from the parameters that we measured with L=100s lookahead, but the parameters should be measured for all the different lookahead values.

[3] The vacationing events are missing from the tandems, so the service time of the tandems will be less, than it was in the beginning of the simulation.

We measured the value of λ for the different values of L to see how much the difference is. Now, we did not use samples from the P and E values displayed by the simulator during the experiments (as we did it for the initial estimation of λ), rather made complete runs and used the measured full event number (*ev*) and execution time (*sec*) for the calculation of P and E. The elapsed virtual time was always 10 days, that is, 864000 *simsec*. Table 3. shows the measured values, and the λ values calculated (incorrectly) using linear extrapolation from the L=100s case. It is apparent that in the L=1000s case there is a significant difference between the measured and linearly extrapolated λ values.

Table 3. The Values of λ in the Function of the Lookahead (measured vs. calculated from the initial estimation)

L [simsec]	**0.1**	**1**	**10**	**100**	**1000**
# events	138122606	138091806	137816386	134885378	102957082
exec. t.*[sec]*	524.18	521.36	523.09	516.54	415.73
P [ev/sec]	263502.24	264868.43	263465.92	261132.49	247653.72
E [ev/simsec]	159.86	159.83	159.51	156.12	119.16
meas.'d λ	**2.43**	**24.14**	**242.17**	**2391.39**	**19246.76**
λ_0*L/L_0	2.50	25.00	250.00	2500.00	25000.00

The Effect of the Magnitude of λ on the Available Speed-up

To fully explore the effect of the magnitude of λ on the available speed-up, we conducted a series of experiments for other values of *L:* L=100ms, 1s, 10s, 100s and 1000s. Figures 3-7 show the results.

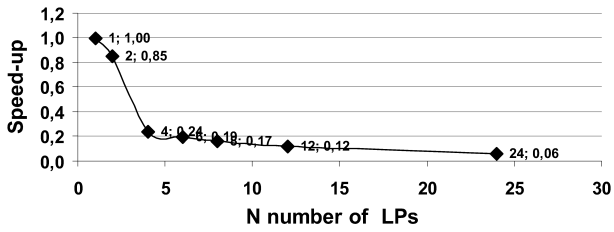


Figure 3. Speed-up in the Function of N for L=0.1s

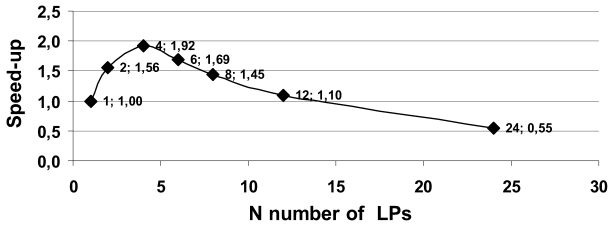


Figure 4. Speed-up in the Function of N for L=1s

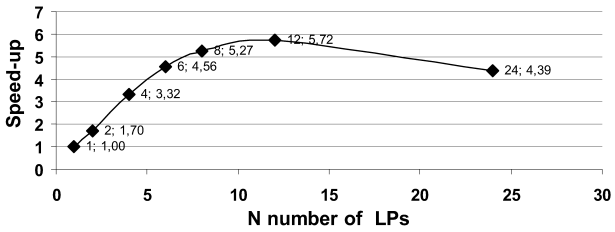


Figure 5. Speed-up in the Function of N for L=10s

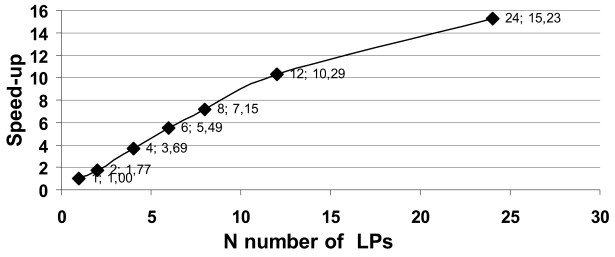


Figure 6. Speed-up in the Function of N for L=100s

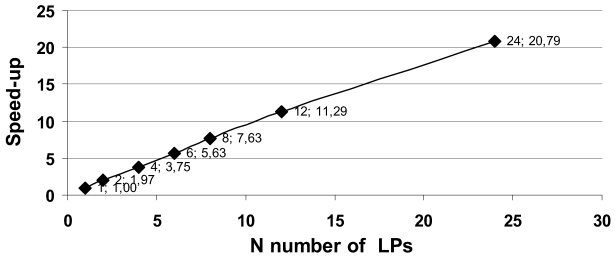


Figure 7. Speed-up in the Function of N for L=1000s

Results in Figures 3-7 have been summarized in Figure 8, and speed-ups have been translated to relative speed-up (speed-up divided by the number of LPs) for better comparison.

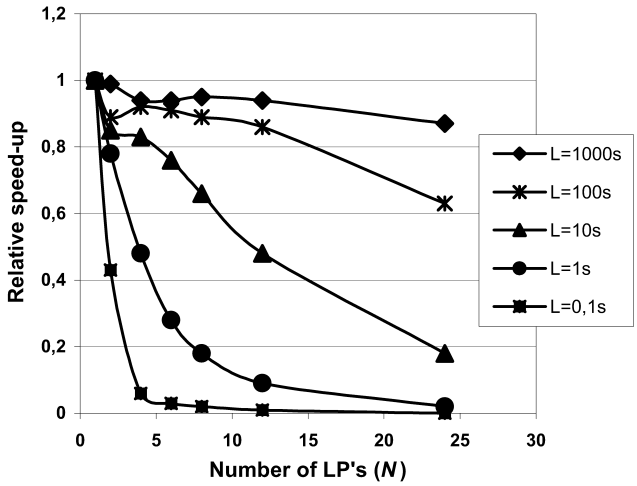


Figure 8. Relative Speed-up in the Function of N for L=0.1s, 1s, 10s, 100s, 1000s Lookahead

Figure 8 shows the relative speed-up in the function of N for these values of L. The nearly constant 1 value of the relative speed-up for L=1000s means a near-linear speed-up. This is in accordance with the value of λ_{24}=19246.76/24≈802 (taken from Table 3). As for the other extremity, there is no speed-up even for two processors for L=0.1s – this is also complies with our expectations as λ_2=2.43/2≈1.21. Halfway (on the logarithmic scale) between them, there is a certain speed-up for L=10s, see Figure 5. Here, the measured value of λ_4≈60.68 is practically equal to the previously used estimated one (62.5). For λ_8≈30.34, there is an acceptable speed-up of 5.27, and probably this is the most economical working point of the system, as the further increase of the number of LPs (and CPU cores) does not significantly improve the performance.

Figure 4 shows the speed-up for L = 1s lookahead. Here, the highest speed-up of the value 1.92 can be observed when there are 4 LPs executed by 4 cores. Here sequential λ≈24.14 results in λ_4≈6. Now, we are on the theoretical boundaries: λ_{24}≈1 and the simulation in 24 LPs requires nearly twice longer execution time than the sequential simulation.

Figure 9 shows the relative speed-up in the function a λ_N. The data series of a given color belong to a given value of L. Note that as λ_N is decreasing in the function of N, if we consider the values of N in ascending order: (2, 4, 6, 8, 12 and 24) then we need to look at the data series from the right to left.

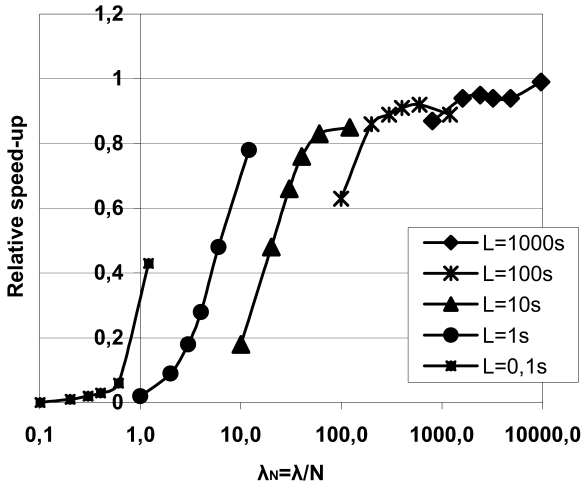


Figure 9. Relative Speed-up in the Function of λ_N for Different Values of L, and for N=24, 12, 8, 6, 4, 2

On the basis of our experimental results summarized in Figure 9, we can confirm the findings of (Varga et. al. 2003) that:

- If λ is in the order of a couple times 100 or higher, then we may expect good speed-up. It may be nearly linear (that is the relative seed-up is close to 1) even for higher number of LPs (we checked it until 24) if λ_N (that is λ divided by the number of LPs) is also at least in the order of a couple of hundreds.

- If λ is in the order of a couple of times 10, we may experience fair speed-up for a number of processors until λ_N (that is, λ divided by the number of LPs) also remains in this order (or at least λ_N is higher than a couple times 1).

- No speed-up is possible if λ or λ_N (that is, λ divided by the number of LPs) is less than 1.

CONCLUSIONS

In this paper we have experimentally verified that a coupling factor of $\lambda >> 1$ is a necessary precondition of getting a good speed-up with conservative parallel simulation. We have used a closed queueing network as simulation model, and run it on up to 24 CPU cores. The results confirm that with our model, a $\lambda_N = \lambda/N$ (N being the number of LPs) value near or below 1 practically prohibits good parallel performance. The 10..100 range of λ_N can provide an acceptable speed-up, and there is a high chance for a good speed-up if λ_N is above that range.

One has to keep in mind that λ only helps to determine whether a given lookahead is enough for a given simulation model and software/hardware environment, so it cannot guarantee good speed-up. Speed-up can still be adversely affected by other factors such as excessive message traffic between LPs as a result of partitioning.

However, the criterion for λ provides a quick and convenient way to determine whether it makes sense to experiment with parallelizing a particular simulation model or not, before actually investing work in the parallelization.

We conclude that the recommended methods are worth further studying.

ACKNOWLEDGEMENT

The authors of this paper would like to express their thanks to Éva Szabó, who conducted the measurements used in this paper.

REFERENCES

Fujimoto, R. M. 1990. Parallel Discrete Event Simulation. *Communications of the ACM* 33 no. 10, 31-53

Lencse, G. 1998. "Efficient Parallel Simulation with the Statistical Synchronization Method" *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation (CNDS'98)* (San Diego, CA. Jan. 11-14). SCS International, 3-8.

Lencse, G. 1999. "Applicability Criteria of the Statistical Synchronization Method" *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation (CNDS'99)* (San Francisco, CA. Jan. 17-20). SCS International, 159-164.

Lencse, G. 2002. "Parallel Simulation with OMNeT++ using the Statistical Synchronization Method" *Proceedings of the 2nd International OMNeT++ Workshop* (Jan. 8-9, 2002, Technical University Berlin, Berlin, Germany) 24-32.

Pongor, Gy. 1992. "Statistical Synchronization: a Different Approach of Parallel Discrete Event Simulation". *Proceedings of the 1992 European Simulation Symposium (ESS'92)* (Nov. 5-8, 1992, The Blockhaus, Dresden, Germany.) SCS Europe, 125-129.

Pongor, Gy. 1992. "Statistical Synchronization: a Different Approach of Parallel Discrete Event Simulation". *Proceedings of the 1992 European Simulation Symposium (ESS'92)* (Nov. 5-8, 1992, The Blockhaus, Dresden, Germany.) SCS Europe, 125-129.

Varga, A., Y. A. Sekercioglu and G. K. Egan. 2003. "A practical efficiency criterion for the null message algorithm". *Proceedings of the European Simulation Symposium (ESS 2003)*, (Oct. 26-29, 2003, Delft, The Netherlands.) SCS International, 81-92.

Sekercioglu, Y. A., Varga, A. and Egan, G. K. Egan, 2003. "Parallel Simulation Made Easy with OMNeT++". *Proceedings of the European Simulation Symposium (ESS 2003)*, Oct. 26-29, 2003, Delft, The Netherlands.

Varga, A. and Hornig, R., 2008. "An overview of the OMNeT++ simulation environment", *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. March 7, 2008, Marseille, France.

AUTHOR BIOGRAPHY

GÁBOR LENCSE received his M.Sc. in electrical engineering and computer systems at the Technical University of Budapest in 1994, and his Ph.D. in 2001 from the same university. His area of research is simulation methodology and parallel discrete event simulation. His academic interests include the acceleration of the simulation of information and communication technology systems. Since 1997, he has been affiliated with the Széchenyi István University in Győr, where he lectures on computer networks, networking protocols and Linux as Associate Professor. He is a founding member of the Multidisciplinary Doctoral School of Engineering, Modelling and Development of Infrastructural Systems at the Széchenyi István University. He performs R&D in the field of the simulation of communication systems for the Elassys Consulting Ltd. since 1998. Dr Lencse has been working part time at the Budapest University of Technology and Economics (the former Technical University of Budapest) since 2005, where he teaches computer architectures.

ANDRÁS VARGA received his M.Sc. in electrical engineering and computer systems from the Technical University of Budapest in 1994. Between 1994 and 1998 he pursued PhD studies at the same university, but went to the industry before completing the PhD. Between 2003 and 2005, he spent altogether more than a year at CTIE, Monash University, Melbourne as a visiting research fellow, where he was performing research on parallel simulation of large telecommunication networks, and working towards the PhD degree. His field of interest is the simulation of telecommunication networks, and he is the architect and main developer of the widely used OMNeT++ network simulation software. Currently he works at OpenSim Ltd. that he founded for commercializing the OMNeT++ software.

FLUID CHEMICAL SIMULATION

FAULT DETECTION IN FLUID SYSTEMS: AN INTERACTIVE REAL-TIME APPROACH

C. Angeli
Department of Mathematics and Computer Science
Technological Education Institute of Piraeus
Konstantinoupoleos 38, N. Smirni
GR-171 21 Athens, Greece
E-mail: c_angeli@otenet.gr or
c.angeli@computer.org

A. Chatzinikolaou
S. Patsi 62
GR-118 55 Athens, Greece
E-mail: achatzi@otenet.gr

KEYWORDS

Modelling, Simulation, Real-time, Hydraulic systems, Fault detection

ABSTRACT

Model-based fault detection is typically realised by evaluating the possible deviation between expected and real behaviour of a system. In the cases that this comparison is performed under dynamic conditions during the normal operation of an industrial system a special kind of models is needed for this interaction. In this paper an implementation model is presented where simulation results and real-time data acquired from an actual system are effectively compared on-line in a suitable environment and the residuals are evaluated for the fault detection process. The experimental results show that the proposed method is reliable and can be applied to real world systems.

INTRODUCTION

Fault detection and diagnosis is an important and challenging problem in engineering applications and continues to be an active area of research over the last years. For the complex highly automated systems encountered in process industries it is fundamental to be able to monitor the health of the installation in order to detect faults and to locate defective components.

Engineering industrial systems can be expected to develop faults during their operation in a length of time. These possible faults need to be detected and identified by an automated system that would be able to recognise incipient faults in specific components in particular those that are central to the optimum and safe operation of the mechanical system.

Reliability is essential for electro-hydraulic systems due to the high energy densities associated with their operation. A failure can have catastrophic results if for example a containment failure of hydraulic fluid at high pressure occurs. Model-based fault diagnosis offers a well established approach that improves the reliability of a fault detection system. Model-based techniques for non linear engineering systems have been proposed by many researchers such as (Patton et al. 2000, Witczak 2007, Isermann 2005, Narasimhan and Biswas 2007,Zhao et al. 2005, Bokor 2009, Medjaher and Zerhouni 2009). The on-line implementation of these techniques presents some problems due to the uncertainties arising in dynamic industrial environments. Most industrial diagnostic systems rely on the comparison of measured signals to specified thresholds but they don't exploit the correlation existing between the different signals that allow the detection and evaluation of important deviation for incipient changes.

In this paper an approach for effective interaction of real-time data and modelling information for the detection and diagnosis of incipient faults in online performance of electro-hydraulic systems is presented. The dynamics of hydraulic motor-mass-pipes system was modelled. Real-time data are acquired from the actual system and compared with the relevant variable values of the simulation process. The detection of incipient faults is realised taking into consideration specific parallel deviation of pressure signals with or without relevant deviations of angular velocity signals. Previous research work in development of diagnostic functions for these systems is provided by (Angeli 2008, Murrenhoff et al. 2004, Kwan et al. 2003, Jelali and Kroll 2003, Gao et al. 2003, Angeli and Chatzinikolaou 2005, Ming-Chang and Tsai 1997).

MODELLING OF THE SYSTEM

The hydraulic system used in this work is a typical system for a production machine, Figure 1. A hydraulic motor is driven through a hydraulic power unit. The system works with a proportional 4-way valve. The proportional 4-way valve controls the flow rate to the motor according to an input current I and controls the rotation of the hydraulic motor with the attached mass J_m.

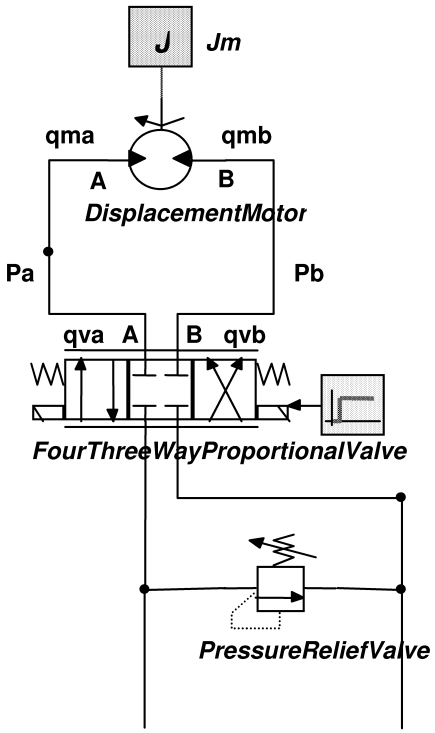


Figure 1. The actual system

The hydraulic motor is a fixed displacement motor with an attached rotary encoder. The function diagram represents the operating curves of the motor. The flow curves, Figure 2, can be used for the estimation or validation of the volumetric efficiency of the hydraulic motor.

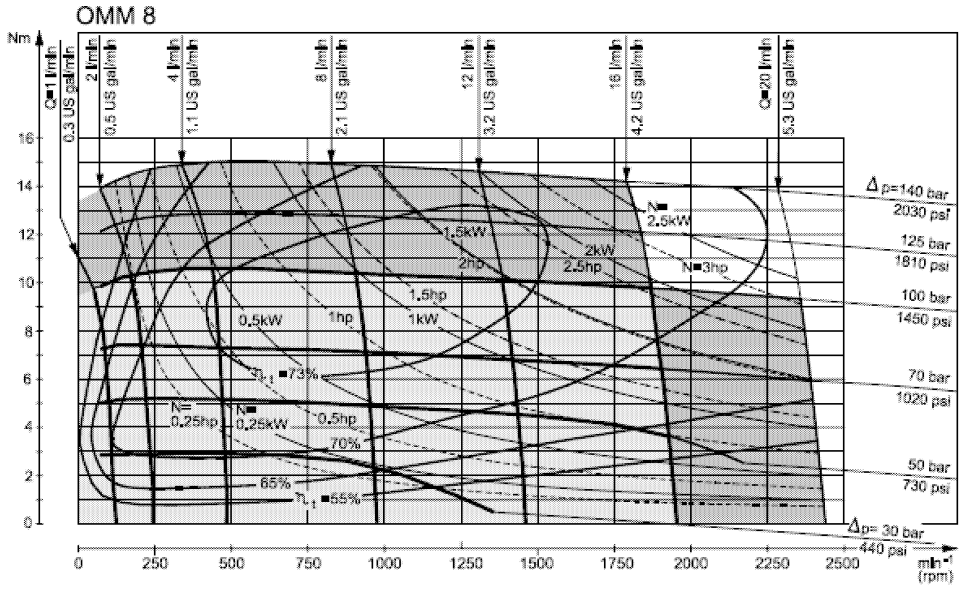


Figure 2. Motor performance as a function of operating speed

The hydraulic system of Figure 1 drives the hydraulic motor through a cyclical routine which requires a high speed initially and then returns to a low speed according to a periodically changed voltage U_2. Assuming that the pressure of the power unit is constant, the variables of the system are following: The pressures p_a p_b and at the ports A and B respectively of the hydraulic motor, the angular velocity ω, the flows q_{va} and q_{vb} through the A and B ports of the proportional 4-way valve, the flows q_{ma} and q_{mb} through the ports A and B of the hydraulic motor, and the voltage to the amplifier of the proportional valve U_2.

It can be assumed for components with large quantities of oil as pipes and hydraulic actuators that the change in pressure in the dynamic state is proportional to the net inflow of oil, that is:

$dp / dt = (E / V_o) . \Sigma Q$ where

E is the elasticity module of the oil plus the included air,

V_o is the Volume of the pipe plus a part of the volume of the attached actuator,

$\Sigma Q = Q_{in} - Q_{out}$,

Q_{in} is the incoming flow to the volume V_o of a connecting pipe and

Q_{out} is the outgoing flow from V_o.

In consequence the pressure increase in a pipe element at a junction is proportional to the algebraic sum of the incoming and outgoing flows and inverse proportional to the included oil volume.

The modelling of the hydraulic elements with the coupled moving masses leads to a non-linear system of equations.

For the simulation process the 20-Sim software was used. The model was validated in comparison to the real process data to determine experimentally the uncertain parameter values and define the acceptable limits of deviation between the measurements and the simulation results in order to establish the fault detection system.

ACQUIRING OF REAL TIME DATA

In order to acquire real-time data a typical input voltage was applied to the system. Figure 4 shows the pulse waveform of the signal U to the amplifier of the proportional 4-way valve and the form of the corresponding waveform of the speed ω of the hydraulic motor.

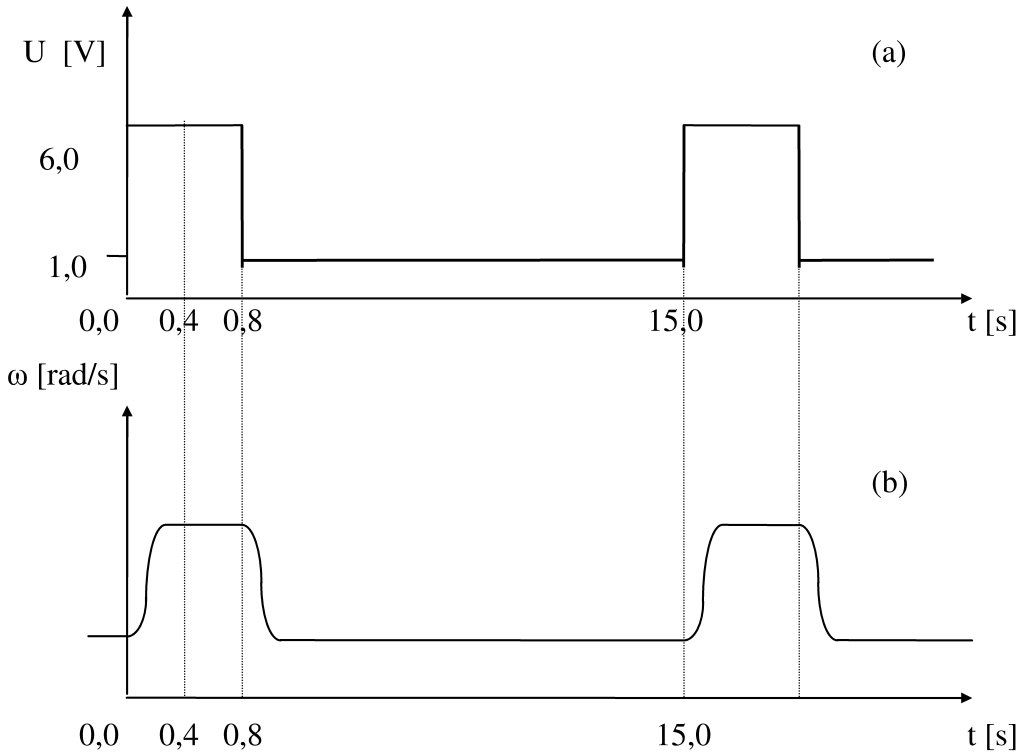


Figure 4. Operation cycle of the system: (a) U - pulse; (b) Response of the hydraulic motor

On the normal operation the hydraulic motor takes approximately 0,4 s to change the speed from a low value corresponding to U=1 V to a high value corresponding to U=6 V. Any fault that occurs in the system can affect both the dynamic and the steady state of the system. Thus if data are taken over a period 0 to 0,4 s after having applied the change of voltage U both the dynamic and the steady condition should be determinable.

FAULT DIAGNOSIS ALGORITHM

In model-based fault detection, faults are detected by setting a threshold on a residual generated from the difference between measurements and estimates of these measurements. The residual signals carry information on the time and isolation of the faults. When a fault occurs, the deviation from zero is such that the new condition can be distinguished from the fault free working mode. The role of the decision system is to determine whether the residuals differ significantly from zero and to decide which is the faulty component. Figure 5 presents a diagram of diagnosis process using system's models.

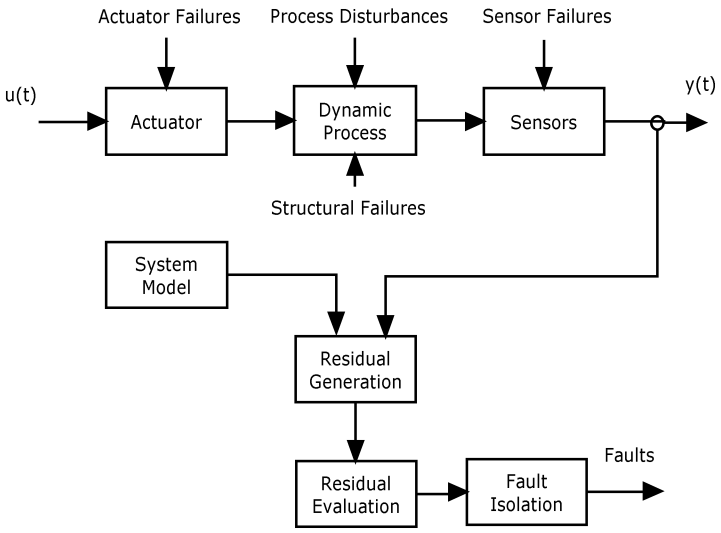


Figure 5. Fault diagnosis using system models

In following the on-line comparison process between measured and simulated values of the pressures p_a, p_b and the angular velocity ω is presented as well as the transformation of the comparison results to linguistic variables.

For this process the DASYLab software was used. The comparison of measured and calculated values of pressure at different point and the angular velocity are presented in Figure 6.

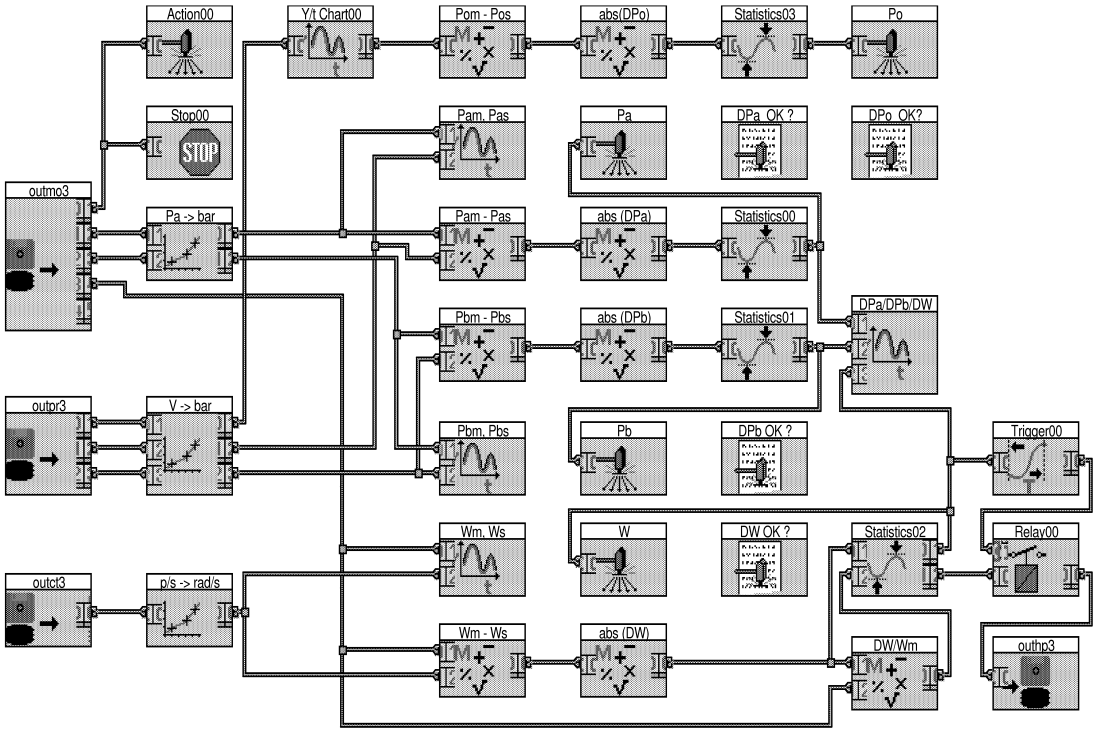


Figure 6. Comparison of measured and calculated data

In Figure 6 the module “outmo3:” and “outpr3:” contain the numerical data from the simulation process and the respective measured values of the pressures p_a, p_b. The processing of the pressure p_a is performed by the modules “Y/t Chart00:chart” " Pam-Pas:" and "abs(DPa):". The module " Pa:" sends the message "OK" in the case that the difference between Pam and Pas is 0 to3 bar, "SMALL" in the case that this difference is 3 to 6 bar and "MED" in the case of difference 6-10 bar to the message module “DPo OK?” and writes these values to the file "fpa.txt". Similar processing has been performed for the pressure at other points of the hydraulic system. This information is used by the expert system as input together with the text file information coming from the data acquisition process and experiential knowledge to determine more precisely a faulty element. Incipient fault are detected in the case of small

deviation in the positive response of the pressure curves in relation to the relevant deviation of the angular velocity signal. In following Figure 7 measured and calculated data for the pressure values p_a are simultaneously presented for the fault free case are presented.

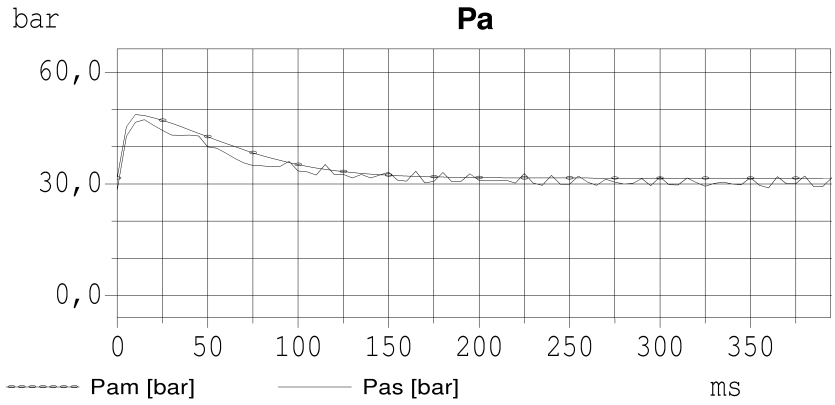


Figure 7. Comparison between measured and calculated pressure values p_a in a leakage free case

A variation of parameter values as the friction torque M_r, the moment of inertia J_m and the oil elasticity E of ± 5 %, and ± 10 % on the simulation is not affect the fault detection process.

CONCLUSION

This paper facilitates an interaction approach of modelling information with real-time data for fault detection and diagnosis in electro-hydraulic driven systems. The research identifies new opportunities to further development of automated tasks for these systems and as consequence a higher degree of reliability of the operation of these systems in an industrial environment by preventing low performance, small faults or unexpected shutdowns. The experimentation results show the reliability of the developed diagnostic system.

REFERENCES

Angeli C. 2008. "On-line Expert Systems for Fault Diagnosis in Technical Processes". *Expert Systems*, Vol. 25, No2, pp. 115-132.

Angeli C. and A. Chatzinikolaou. 2005. "Troubleshooting in Hydraulic Systems using knowledge-based Methods" *International Journal of Engineering Simulation*, Vol. 6, Nr.1, pp. 24-29.

Bokor J. 2009. "Fault Detection and Isolation in Nonlinear Systems", *In Proceedings 7th IFAC Safeprocess 09*, Barcelona, Spain.

Gao Y., O. Zhang and X. Kong. 2003., Wavelet-based Pressure Analysis for Hydraulic Pump Health Diagnosis, Transaction of ASAE, Power and Machine Division, Vol. 46(4), pp. 969-976.

Jelali M., and A. Kroll,. 2003. "Hydraulic servo systems – Modelling, Indentification & Control – (advances in industrial control)" , Springer Verlag, London.

Isermann R. 2005. "Model-based fault detection and diagnosis status and applications" *Annual Review of Control*, Vol. 29, No. 1, pp. 71-85.

Kwan C., R Xu. and X. Zhang. 2003. "Fault detection and Identification in Aircraft Hydraulic Pumps Using MCA", In Proceedings Safeprocess 03, Washington, U.S.A. pp. 1137-1143.

Medjaher K., and N. Zerhouni. 2009. "Residual-Based Failure Prognosis in Dynamic Systems", *In Proceedings 7th IFAC Safeprocess 09*, Barcelona, Spain.

Murrenhoff H. Th. Meindorf and C. Stammen 2004. "Condition Monitoring in Fluid Technology". *Proceedings of 4th International Fluid Power Conference*, Vol. 2, Dresden, pp. 219-244.

Ming-Chang Shin and Chung-Din Tsai. 1997. "Servo Hydraulic Cylinder Position Control using a NeuroFuzzy Controller", *Mechatronics*, 7/4, 1997, pp. 355-368.

Narasimhan S., and G. Biswas. 2007. "Model-based diagnosis of hybrid systems", *IEEE Transactions on Systems, Man and Cybernetics, A Syst. Humans*, Vol. 37, No. 3, pp. 348-361.

Patton R., Frank P and Clark R. 2000. "Issues in Fault Diagnosis For Dynamic Systems", Springer-Verlag, New York.

Witczak M. 2007. "Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems: From Analytical to Soft Computing Approaches". Springer-Verlag, Berlin-Heidelberg.

Zhao F., X. Koutsoukos, H. Haussecker, J. Reich and P. Cheung. 2005. "Monitoring and Fault diagnosis of hybrid systems", *IEEE Transactions on Systems Man and Cybernetics*, Vol. 35, No. 6, pp. 1225-1240.

TOPOLOGICAL FEATURES AND PARAMETERS OF BIOCHEMICAL NETWORK STRUCTURE

Tatjana Rubina, Egils Stalidzans
Latvia University of Agriculture, Faculty of Information Technology,
Department of Computer Systems, Latvia, Jelgava, Liela 2
Tatjana.Rubina@llu.lv, Egils.Stalidzans@llu.lv

KEYWORDS

Network, topological model, parameter identification, bond graphs, computer software.

ABSTRACT

In the cell, tissue, organ and organism under different conditions operate metabolic, protein interaction, gene or genetic regulatory and signaling networks that describe biochemical reactions, biochemical or biophysical process. The biochemical networks can present the relationship between genes and gene products, proteins, metabolites and etc. The exploration of these networks is the key task in systems biology in the last few years and helps to understand some cellular process, functions or properties of biological system better. We provide researchers and coterie information on network structure analysis, existing and used topological features with the following goals: 1) to accumulate the existing knowledge about the network structure analysis; 2) to provide a list of topological features and characterize the most of them, mentioning its biological sense; 3) to provide the information of the existing software tools for the structure analysis.

1 INTRODUCTION

The goal of systems biology is to understand living organisms at the systems level, combining quantitative information on individual components in order to understand the emergent behaviors that result [14]. According to the systems theory, a system is defined as the set of objects with relations between objects and these properties. The living organisms can be considered as biological systems. In most cases, the biological systems are very complex. System biologists build a system-level understanding of how the biological world works and solve problems by understanding systems and then applying that knowledge to control them [20].

During the last several years, the biological research produces increasing volumes of data describing genome sequence of biological organisms, cellular components, their interactions, and states of biochemical networks for model organisms [12]. This available information about biological entities and their interactions enable us to consider different organisms as biological systems which control the genetic information [22].

To understand biology at the system level, we must examine the structure and dynamics of cellular and organismal function (Kitano) [1]. According to the systems theory, a structure is defined as the set of elements and relations (links) between them that define functions of system and distribution of the purposes [7]. Analyzing the structure, usually the researchers are interested in the properties of the system, which remain invariable for a long period of time or functioning on the system. Living organisms are subjected to the process of genetic mutations dependant on environmental conditions (external perturbations), it can be necessary to detect invariable and varyable system properties. It can be done with the purpose to compare several systems of one type, to detect influence of the certain and separate factors on the system.

In this research paper, we focused on the structure of biochemical networks and their exploration.

2 EXPLORATION OF BIOCHEMICAL NETWORKS

The goals of systems biology are to understand the mechanisms of how biochemical networks generate particular cellular functions in response to environmental stresses or genetic changes [18]. Structural or topological analysis of cellular interaction networks contributes to a deeper understanding of network-wide interdependencies, causal relationship, and basic functional capabilities. Structural analysis, towards a functional analysis of the structure is not based on quantitative and dynamic properties and can thus only provide qualitative answers [13].

It is important and necessary to explore, for example, protein interaction or gene regulatory networks, while they play a key role in disease understanding and accurate drug targets [11] finding. Structural or topological (parameter-free) and dynamic models are used to explore the networks. It is necessary to notice that the basis of each model is the

structure. In this paper, we focused on a structural model. In structural models networks are represented in directed or undirected graph form [13, 28, 29, 31] and mixed graphs [28], that consist of nodes and edges. Nodes represent genes, gene products, proteins, chemical compounds or small molecules. Graph links (edges in undirected graph or arcs in directed graph) represent functional relationships [8] i.e., various types of interactions or associations between the pair of nodes, e.g. metabolic reactions or events, gene modification, protein/protein-nucleotide interactions, protein modification, regulatory relationships such as transcriptional and translational regulation, signaling pathways etc. [25, 29]. Connections can be directed or undirected dependent on graph type; they can have physical meaning, denote general associations; they can represent shared characteristics [28] between components. Nodes depend on the number of connections and the type of graph can be hubs (highly connected nodes) [30] and metanodes [28].

The choice of the network representation is often dictated by a research issue. Directed networks are suitable when the interactions between two components have a well-defined direction, for example, the direction of metabolic flow from substrates to products, or the information flow from transcription factors to the genes that they regulate. For example, Zhao et al in their work [27] represent Homo sapiens metabolic network by a directed graph, which nodes correspond to metabolites and the arcs – to reactions between these metabolites, in which irreversible reactions are represented as directed arcs while reversible ones as bi-directed arcs (see Fig.3.). Transcription factor binding networks [28] can be represented too as directed networks.

Undirected networks, such as protein interaction networks, represent a mutual relationship: if protein A binds to protein B, then protein B binds to protein A. This type of representation also often applies to predictions made by high-throughput proteomic or genomic analysis, or indirect links based on shared genes or protein components between pathways and complexes [28].

3 FEATURES AND PROPERTIES OF NETWORK STRUCTURE

Networks have "emergent" properties that are distinct from those of their individual components. Emergent properties are non-linear, aggregated and combinatory effects generated by the interaction of the components of the network. For example, properties such as topology, information flow and the stable state of a network can only be detected at the network level, not by examining the individual components such as genes or proteins. The structural and dynamic features of genetic networks ultimately contribute to biological functions, robustness and evolvability of the networks [9].

Examining scientific literature, publications and analyzing software tools, we found many network structure properties and features, which can be categorized in five groups: network metrics, network motifs, topological parameters, topological features, and structure qualitative parameters (see Fig.1.). These properties and features are not the only one, but mostly used in studies and researches.

3.1 Topological parameters

In this paragraph we will describe such topological parameters as degree, degree distribution, paths, path length, path distribution, network diameter. The following definitions will be used graph node, vertex – a network component, graph edge or arc – link or connection.

Degree of a network component. The degree (or connectivity [3, 21]) of an undirected network element or node, k_i, is the number of edges (links) that it has with other elements that is incident with i:

$$k_i = \sum_j^n k_{ij} \qquad (1)$$

A degree is also a feature that distinguishes hubs (highly connected nodes) from leaves or orphans (weakly or non-connected nodes) in the network [31]. In protein interaction and genetic interaction networks, for example, the degree of a hub (highly connected component) is often its importance and essentiality for cell function [28], process or whole system.

For example, Han et al. [9] and Partil and Nakamura [19] defined hubs as nodes with degrees of more than 6 ($k \geq 6$).

Hase et al [11] in their study of Structure of Protein Interaction networks (PIN) and their implications on Drug design, has analyzed budding yeast and Human PINs to identify these topological features. Depending on degree value, there are 3 types of node degrees for each PIN: low-degree, middle-degree and high-degree nodes. **In the budding yeast PIN:**

- ✓ **Low-degree nodes** are nodes with degree of less than 6 ($k<6$),
- ✓ **Middle-degree nodes** are nodes with degrees from 6 to 38 ($6 \leq k \leq 38$),
- ✓ **High-degree nodes** are nodes with degrees of more than 38 ($k>38$).

In the Human PIN they define:

- ✓ **Low-degree nodes** are nodes with degree of less than 6 ($k<6$).
- ✓ **Middle-degree nodes** are nodes with degrees from 6 to 30 ($6 \leq k \leq 30$).
- ✓ **High-degree nodes** are nodes with degrees of more than 30 ($k>30$).

The degree of node can be used to examine the network structure. By examining the network structure, researcher

should examine these elements, i.e., nodes by using degree of node. If the network has isolated [35] elements (which degree is 0), it means that in construction, description or creation of network structure have been committed errors.

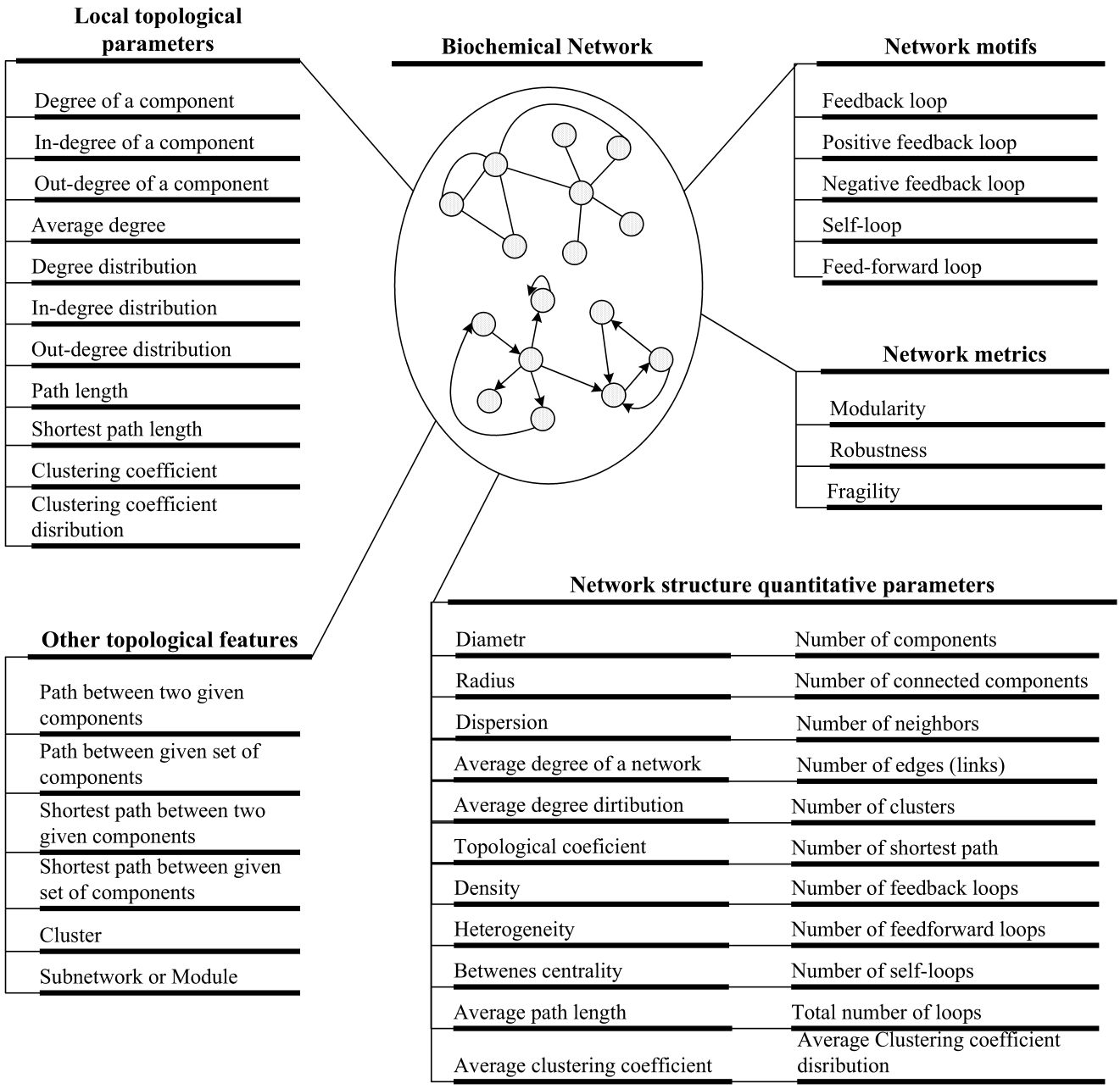


Fig. 1. Biochemical network measures and properties

In-degree and Out-degree distribution. For directed networks, such as transcription factor binding networks, the degree is separated into 'in'-degree and 'out'-degree, depending on the directions of interaction between two given elements [28].

Incoming degree (In-degree). The incoming degree is a number of links that point to the network component *I* [21]:

$$k_{+i} = \sum_{j}^{n} k_{ij} \qquad (2)$$

Outgoing degree (Out-degree). The outgoing degree is a number of links that start with network component [21].

$$k_{i+} = \sum_{j}^{n} k_{ij} \qquad (3)$$

Average degree. For an undirected network, the average degree in an undirected network [3, 5]:

$$K = < k > = \frac{2L}{N} \qquad (4)$$

, where N – the total number of components

L – the total number of links

Average degree of a network. For an undirected network, the average degree of the whole network is the average of the degrees of all individual network components [5]:

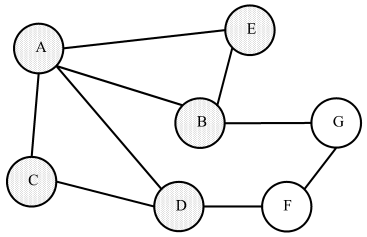
$$K = \frac{\sum_{i \in V} k_i}{N} \qquad (5)$$

, where k_i - the degree of node i

N - the total number of nodes

Degree distribution. Degree distribution d_k is the number of nodes with degree k ($k=1,2,...n$) [5, 21]. In other words, degree distribution is the number of neighbors of network component i.

For directed networks the degree distribution is separated into **in-degree** and **out-degree distribution**.



$k_A=4$

$k_B=3$

Fig. 2. Undirected network

Let K be the degree of a network element. Then a statistical model for the degree distribution is represented by:

$$P(K = k) = f(k) = \frac{N_k}{N} \qquad (6)$$

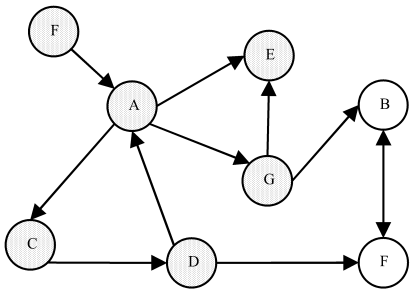
,where $f(k)$ is a probability distribution

N_k – a number of nodes with degree $k=1,2...n$

N – the total number of nodes

The distribution of degrees $f(k)$ in undirected network, gives the probability that a selected component has degree k [3]. In the case of directed networks one needs to consider two distributions, $P(k_{in})$ and $P(k_{out})$ [4].

The distribution of degrees among network components is useful for characterizing the topology and scale of a network, and often has meaningful biological interpretation [28]. In general, degree distribution allows to separate different network types or classes.



$k_{Ain}=2$

$k_{Aout}=3$

Fig. 3. Directed network

The degree distribution of many types of real-life networks, such as metabolic, scientific collaboration networks [21, 26]:

$$P(K = k) \sim k^{-\gamma} \qquad (7)$$

- this function is called a power law

,where γ - a constant or the degree exponent

The smaller the value of γ, the more important the role of the hubs is in the network. Whereas for $\gamma>3$ the hubs are not relevant, for $2>\gamma>3$ there is a hierarchy of hubs, with the most connected hub being in contact with a small fraction of all nodes and for $\gamma= 2$ a hub-and-spoke network emerges, with the largest hub being in contact with a large fraction of all nodes [4, 3].

A **power-law degree distribution** indicates that a few hubs hold together numerous small components or nodes [3]. A network with this degree distribution is called scale-free, indicating that there is a high diversity of node degrees and no typical nodes in the network that could be used to characterize the remaining nodes [28]. In a scale-free networks most nodes have only one or two functional links, whereas a small number of nodes, the hubs, have many links [8]. Nearly all biological networks, including regulatory, interactome and metabolic networks are scale-free [3, 4].

3.2 Topological parameters and features

In studying the function of pathways, the property of interest is often how a given gene or protein is related to (or responds to) an up- or downstream signal. Given a large data set of interactions, it may be useful in some contexts to find the most direct path between two genes, proteins, complexes or pathways; for example, the overall lengths of such pathways may be related to the immediacy or breadth of signal response [28].

The Path [35] form n_0 to n_k, according to the graph theory, is the sequence of nodes.

There can be different types of paths: chain (have all different edges), simple chain (have all different nodes), closed chain or cycle (starts and ends with the same node).

The path length l_{ij} is the number of edges in path from node i to j.

The path between two given nodes (see Fig.4.). In case of signaling networks, the computation of all paths between pair of species helps to recognize all the different ways in which a signal can propagate between two nodes, e.g. all the different ways by which a certain transcription factor (or any other species from the output) can be activated or inhibited by signals riving the input layer.

In metabolic networks, researchers are particularly interested in the reactions (edges), because they correspond to enzymes that are subject to regulatory processes and can be knocked-out in experiments. In metabolic pathway analysis, a statistical or combinatorial analysis of the participation and co-occurrences of reactions in elementary modes proved to be useful for obtaining system-wide properties, such as the detection of essential reactions/enzymes or correlated reaction sets (enzyme subsets) [13].

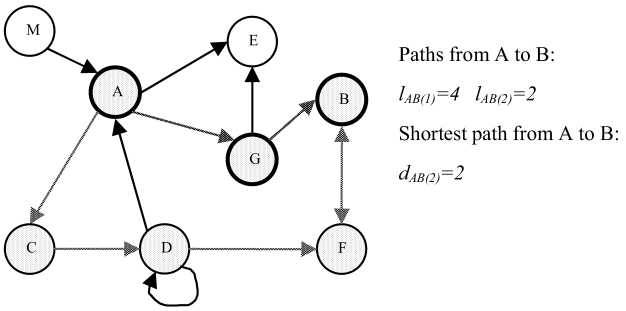


Fig. 4. Paths between two given nodes

The shortest path between two nodes is a path with the minimum number of edges that is necessary to traverse from one node to the other.

In weighted directed graph the shortest path [32] is the path between two nodes (source and target nodes, start and end nodes) with minimal sum of weights of the edges making the path.

The suboptimal path [32**]** is constructed by removing all edges in all shortest paths one by one and one at a time and finding the shortest path.

Average Path length of a network is the average of the shortest path lengths between every node pair [5].

$$L = \frac{2\sum_{\substack{i,j\in V \\ i<j}} d_{ij}}{N(N-1)} \qquad (8)$$

, where d_{ij} – the shortest path length between nodes i and j. Average path length reflects how closely nodes are connected within the network and offers a measure of a network's overall navigability [5]. The average shortest path also indicates the well-known 'small-world' property of many real-life networks [28, 5].

Shortest path distribution. *SP(l)* is defined as the proportion of shortest paths with a specified length l in a network and reflects the diversity of the graph distances between two nodes in the network. In most real-life networks, there is a relatively short path between any two nodes, and the length is in the order of logarithm of the network size. This property is known as "small world" [31].

Diameter of the network is the (longest) distance between two most remote nodes [5]:

$$D = \max_{i,j\in V} d_{ij} \qquad (9)$$

, where d_{ij} – the shortest path length between nodes i and j.

In a network, different nodes have different levels of connectivity. It is necessary to evaluate which node is the most important. Network centrality is a local quantitative measure for assessing the position of a node relative to the other nodes, and can be used to estimate its importance or role in a global network organization. Different information sources reveal several centrality measures such as degree centrality, closeness centrality, and betweenness centrality [5]. Betweenness centrality better predicts the essentiality of a node then degree centrality [8]

The **closeness** of a node is defined as the inverse of the average distance from all other nodes. The **betweenness** is one of the standard measures of node centrality, originally introduced to quantify the importance of an individual in a network [4]. The betweenness *bi* of a node *i*, is defined as:

$$bi = \sum_{\substack{j,k\in N \\ j\neq k}} \frac{n_{jk}(i)}{n_{jk}} \qquad (10)$$

, where n_{jk} - the number of shortest paths connecting nodes j and k,

$n_{jk}(i)$ - the number of shortest paths connecting nodes j and k and passing through i.

The concept of betweenness can be extended also to the edges. The **edge betweenness** is defined as the number of shortest paths between pairs of nodes that run through that edge [4].

3.3 Network motifs

Identifying topological features in networks is an important part of understanding the relationship between structure and function of network motifs, e.g. feedback loops, feedforward loops (see Fig.5.).

Structural model characterize and provide information of the connectivity (topology) of the interactions involved in a biological process or system.

However, some insights into the dynamic properties can nevertheless often be obtained, because fundamental properties of the dynamic behavior are often governed by the network structure [13]. Feedback and biological regulation are two sides of the same coin, reflecting the need of the living cell to deal with changing environtments, to generate cell to cell heterogeneity and to optimize cellular metabolism to a given external condition [15]. By using feedback loops a

network component promotes its own accumulation or activation. For example, a protein can activate its own transcription factor or inhibit its own proteolysis. It may activate its own activator, or somehow remove its own inhibitor. [1]

According to the graph theory, the **feedback loop** is a directed simple cycle or circuit that is defined as a sequence of arcs that starts and ends at the same node (vertex) *i* and visit no node twice.

Feedback loops may be positive or negative, depending upon the parity of the number of negative interactions in the loop, and they play key role in controlling the dynamics [24] of a wide range of biological systems. Negative feedback loops tend to act within biological systems to maintain homeostasis [10] (are essential for homeostatic mechanisms (i.e. for adjusting and maintaining levels of system variables) [13] and are necessary for the existence of periodic behavior [24]. Systems involving negative feedback loops tend to settle to a steady state, which may be stable or unstable [10]. The behavior of a negative feedback loop depends partly upon its length. A one element loop generates a single, stable, steady state; a two element loop produces a single steady state which is approached to and departed from in a periodic way; while a loop with three or more elements can generate damped or stable oscillations depending upon parameter values [10]. Networks with larger number of independent negative feedback loops tend to have longer limit cycles and thus may exhibit more random or chaotic behavior [24].

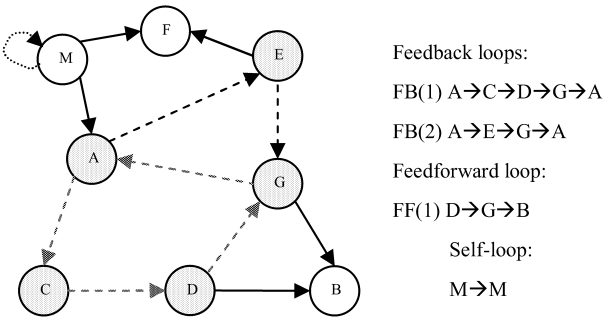


Fig. 5. Feedback, feedforward end self- loops

Positive feedback loops are responsible and even required for multiple steady state behaivior in dynamical systems. In biological systems, multistationarity plays a central role in differentiation processes and for epigenetic and switch-like behavior [13]. Multistationarity is the existence of a number of different stable state [10].

Self-Loop. Self-loop is an arc connecting a species with itself.

3.4 Subnetworks and clusters

It is possible to construct two networks with identical topological measures including degree distribution and clustering coefficient but different hierarchical structure. Therefore, local topological features such as modularity, motif, and network clustering are likely to be key concepts in understanding cellular mechanisms and biological functions in biomolecular networks [5].

The functions of biomolecular networks are closely related to their topologies and facilitated by characteristic topological patterns. Components of cellular networks including genes, proteins, and other molecules usually act in collaboration to carry out specific biological processes and biochemical activities, by forming relatively isolated functional units called modules. From the topological perspective, a module can be understood as a subnetwork that is densely connected within itself but sparsely connected with the rest of the network. For example, in cellular networks, a module refers to a group of physically or functionally connected biomolecules that work together to achieve some desired cellular function [5]. But, in the network of gene expression data, interaction subnetworks is a connected sets of interactions, whose genes show particularly high levels of differential expression. The interactions contained in each subnetwork provide hypotheses for the regulatory and signaling interactions in control of the observed expression changes [5, 23].

Cluster. Depending on the type of network, clusters may mean different things. For instance, clusters in a protein-protein interaction network have been shown to be protein complexes and parts of pathways. Clusters in a protein similarity network represent protein families [5, 23]. In gene regulatory networks, the cluster is a group of genes that are active at the same time are likely to be involved in the same regulatory process [16].

Cluster analysis is applied, for example, to microarrays data that have been collected over a variety of conditions or a series of time points. Its goal is to determine the original groupings of the genes. This technique assumes that genes that are active at the same time are likely to be involved in the same regulatory process. It assumes that genes are grouped and within the group the genes would produce the same expression profile [16].

Clustering coefficient. Clustering coefficients is the ratio of the number of existing connection or links between a node's neighbors to the maximum number of possible links between them [5, 28, 3, 33]:

$$C_i = \frac{2n_i}{k_i(k_i - 1)} \qquad (11)$$

, where n_i is the number of links between k_i neighbors.

While node *A* have 5 neighbors (*C,D, B, E, H*), the number of links between node *A* neighbors $n_A=2$ node A degree is $k_A=5$, the node *A* clustering coefficient is:

$$C_A = \frac{2 \cdot 2}{5 \cdot (5-1)} = \frac{1}{5}$$

Clustering coefficient measures how well the neighbors of a node i are locally interconnected [33] and the tendency of a network to have highly connected clusters. In large-scale mass spectrometric networks in yeast, this property can be

used to identify groups of proteins involved in the assembly of the ribosome [28].

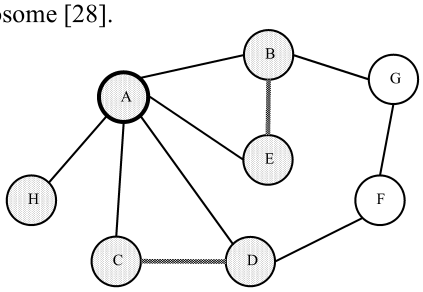


Fig. 6.Clustering coefficient of node A

Average clustering coefficient is the average of all individual clustering coefficients of the nodes in the network. It is the clustering coefficient of a network that characterizes the overall tendency of nodes [3, 4] to form clusters or groups:

$$C = \frac{\sum_{i \in V} C_i}{N} = \frac{\sum_{i \in V} \frac{2n_i}{k_i(k_i - 1)}}{N} \qquad (12)$$

, where n_i – the number of edges existing between the k_i nodes that are connected to node *i,*

k_i – the degree of node *i.*

Watts and Strogatz [34] have shown in their work that the clustering coefficient of many real systems networks is orders of magnitude larger than the one expected for a random network and, therefore, they are far from being random [33].

In particular, clustering coefficient, degree, the minimum path distance between pair of nodes, have attracted the attention of the physics community. Watts and Strogats have shown that real networks are characterized by a small average minimum path distance and a large clustering coefficient that together are named as *small world effect*. The topology of real networks is characterized by degree correlation and clustering hierarchy [33].

Clustering-Coefficient distribution. Cluster coefficient distribution *C(k)* is defined by averaging the clustering coefficients of nodes that have the same degree *k* [3]. It characterizes the diversity of cohesiveness of local neighborhoods and the overall tendency of nodes to form clusters or groups [5]. For many real networks, the exponential degree of the log-linear fit

$$C(k) = k^{-\gamma} \qquad (13)$$

holds, which implies a network's hierarchical character or, in other words, is an indication of a network's hierarchical character. This function *C(k)* is a measure of the network's structure [3] and it captures a network's generic features and therefore can be used to classify various networks.

The **modularity metric** is defined as the gap between the fraction of arcs within clusters and the expected fraction of arcs if the arcs are wired with no structural bias [27]:

$$M = \sum_{i=1}^{r} \left[e_{ii} - \left(\sum_{j} e_{ij} \right)^2 \right] \qquad (14)$$

, where r – the number of clusters,

e_{ij} – the fraction of arcs that leads between nodes (vertices) of cluster i and j.

The maximum modularity metric corresponds to the partition that comprise as many as within-module links and as few as possible inter-module links.

3.5 Software tools for structure analysis

Software tools NetworkAnalyzer and Visant analyze network structure.

Cytoscape is a complex network visualization and analysis tool supporting the standards (SBML, BIOPAX) and customizable network display styles [25]. Cytoscape specializes in the representation of interaction networks. Automatic layout algorithms help to organize massive amounts of interaction data relating to a set of molecules [5].

NetworkAnalyzer is the versatile Cytoscape plug-in that computes a comprehensive list of simple and complex topology parameters (single values and distributions) for directed and undirected networks using efficient graph algorithms. Simple parameters are the number of nodes, edges, self-loops, and connected components, the average number of neighbors, the network diameter, radius, density, centralization, heterogeneity, degree of component and clustering coefficient, the number of shortest paths, and the characteristic path length. Complex parameters are distributions of node degrees, neighborhood connectivity's, neighbors connectivity distribution, average clustering coefficients, average degree distribution, topological coefficients, shortest path lengths, shortest path distribution, average shortest path length and shared neighbors of two nodes. Network-Analyzer displays the distributions as histograms or scatters plots and allow export them as chart images in the formats JPG/PNG/SVG or as tables in plain text files [2].

VisANT is a web-based software framework (Java application) for visualizing and analyzing many types of networks of biological interactions and associations, as well as an especially useful tool for integrating information from a wide variety of sources. VisANT explicitly allows creation of mixed networks involving different types [28], i.e. networks containing both directed and undirected links [25]. Networks can also be analyzed for topological characteristics to identify larger global properties, such as degree (in-degree, out-degree) of a component, degree (in-degree, out-degree) distribution, average degree distribution, path length, shortest path length, clustering coefficient, clustering coefficient distribution calculations [25,29], clustering coefficient distribution and average clustering coefficient

distribution, connections finding between a given set of genes/proteins, i.e., shortest path between two given components or given set of components, feedback loops (cycles), feedforward loops and self-loops.

VisANT provides:

(1) visualization, mining, analysis and modeling of the biological networks, which extend the application of GO [30].

(2) supporting exploratory pathway analysis using metagraphs, which includes multi-scale visualization of multiple pathways, editing and annotating pathways using a KEGG compatible visual notation and visualization of expression data in the context of pathways [29].

(3) the statistical and analytical tools needed for extracting topological properties of the user-defined networks.

4 CONCLUSIONS

The structural or topological analysis of biochemical networks contributes to a deeper understanding of network-wide interdependencies, causal relationships, and basic functional capabilities. The structural analysis is not based on quantitative and dynamic properties and can thus only provide qualitative answers.

Networks have "emergent" properties that are distinct from those of their individual components. Emergent properties are non-linear, aggregated and combinatory effects generated by the interaction of the components of the network.

The network structure properties and features can be categorized in the following groups: network metrics, network motifs, topological parameters(a Cytoscape plug-in) and VisANT are the most powerful tools for biochemical network structure analysis.

5 ACKNOWLEDGEMENTS

Academic study and publication financed by the project "Support for doctoral studies in LUA" /2009/0180/1DP/1.1.2.1.2/09/IPIA/VIAA/017" agreement Nr. 04.4-08/EF2.D1.D6".

6 REFERENCES

[1] Alberghina L., Westerhoff H. (Eds.) (2005) Systems Biology. Definitions and Perspectives.Springer Verlag Berlin Weidelberg, 2005

[2] Assenov Y., Ramirez F., Schelhorn S.-H., Lengauer T., Albrecht M. (2008) Computing topological parameters of biological networks. Bioinformatics Applications Note, Vol.24 no. 2 2008, pp.: 282-284

[3] Barabasi A.L., Oltvai Z.N. (2004) Network biology: understanding the cell's functional organization. Nat Rev Genet 5: 101–113

[4] Boccaletti, S., Latora, V., Moreno, Y.,Chavez, M., Hwang, D.-U.(2006) Complex networks: Structure and dynamics. Physics Reports 424: 175-308.

[5] –Chen L., Wang R.-S., Zhang X.-S. (2009) Biomolecular networks: Methods and Applications in System Biology. John Wikey & Sons, Inc., Hoboken, New Jersey, 2009 ISBN 978-0-470-24373-2

[6] Cline M.S., Smoot M., Cerami E., Kuchinsky A., Landys N., Workman C., Christmas R., Avila-Campilo I., Creech M., Gross B., Hanspers K., Isserlin R., Kelley R., Killcoyne S., Lotia S., Maere S., Morris J., Ono K., Pavlovic V., Pico A.R., Vailaya A., Wang PL., Adler A., Conklin B.R., Hood L, Kuiper M., Sander C., Schmulevich I., Schwikowski B., J Warner G., Ideker T., D Bader G. (2007) Integration of biological networks and gene expression data using Cytoscape. Nature Protocols 2, 2007, Page(s): 2366 – 2382, doi:10.1038/nprot.2007.324

[7] Grundspeņķis J. (2006) Sistēmu teorija un vadība. ESF projekta ietvaros izveidots metodiskais materiāls.

[8] Han J.-D. J. (2008) Understanding biological functions through molecular networks. Cell Research (2008) 18:224-237

[9] Han J.D., Bertin N., Hao T., Goldberg D.S., Berriz G.F., Zhang L.V., Dupuy D., Walhout A. J., Cusick M.E., Roth F.P., and Vidal M. 2004. Evidence for dynamically organized modularity in the yeast protein–protein interaction network. Nature 430**:** 88–93.

[10] Hallinan J.S., Jackway P.T. Network Motifs, Feedback Loops and The Dynamic of Genetic Regulatory Networks.

[11] Hase T., Tanaka H., Suzuki Y., Nakagawa S., Kitano H. (2009) Structure of Protein Interaction Networks and Their Implications on Drug Design. PLoS Computational Biology, Volume 5, Issue 10, October 2009

[12] Herrgard M., Lee B.S., Portnoy V., Palsson B.O. (2006) Integrated analyses of regulatory and metabolic networks reveals

[13] Klamt S., Saez-Rodriguez J., A Lindquist J., Simeoni L., D Gilles E. (2006) A methodology for the structural and functional analysis of signaling and regulatory networks. BMC Bioinformatics2006, 7:56 doi:10.1186/1471-2105-7-56

[14] Kholodenko B., Bruggeman F., Sauro H. (2005) Mechanistic and modular approaches to modeling and inference of cellular regulatory networks, Systems Biology: Definitions and Perspectives. Springer, 2005

[15] Krishna S., Semsey S., Sneppen K. (2007) Combinatorics of feedback in cellular uptake and metabolism of small molecules. PNAS, Vol. 104, No.52, December 26, 2007, 20815-20819

[16] - Myers C. J. (2010) Engineering Genetics Circuits. Chapman & HALL/CRC Mathematical and computational Biology Series. 2010. Taylor and Francis Group, LLC. Page(s): 278.

[17] Newman, M.E.J. (2003) The structure and function of complex networks. SIAM Review 45:167-256. Press. August 25, 2007. [Available at: http://bioinformatics.oxfordjournals.org/cgi/reprint/btm401v1.pdf]

[18] Nishio Y., Usuda Y., Matsui K., Kurata H. (2008) Computer-aided rational design of the phosphotransferase system for enhanced glucose uptake in Escherichia coli. Molecular Systems Biology 4: Art.Nr. 160 [19] Patil A., Nakamura H. (2006) Disordered domains and high surface charge confer hubs with the ability to interact with multiple proteins in interaction networks. FEBS Lett 580: 2041–2045

[20] Paxson R., Zannella K.. System biology: Studying the world's most complex dynamic systems// Journal TheMathWorksNews&Notes - June (2007), pp.4-7

[21] Robins G, Pattison P., Koskinen J. (2008) Technical report. Network degree distribution. University of Melbourne, 2008

[22] Rubina T., Brusbardis V. (2009) Applications of biochemical networks discovering control mechanisms in systems biology. Proceedings of the Annuals Students International Scientific Conference "YOUTH IN SCIENCE AND PROFESSIONAL PRACTICE", 2009

novel regulatory mechanisms in Saccharomyces cerevisiae. Genome research, 2006

[23] - Shannon P., Markiel A., Ozier O., Baliga NS., Wang JT., Ramage D., Amin N., Schwikowski B., Ideker T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Research 2003 Nov; 13(11), Page(s): 2498-504

[24] Sontag E., Vilz-Cuba A., Laubenbacher R., Jarrah A.S. (2008) The Effect of Negative Feedbach Loops on the Dynamics of Boolean Networks. Biophysical Journal, Volume 95, July 2008, 528-526

[25] Suderman M., Hallett M. (2007) Tools for visually exploring biological networks. 2006. Published by Oxford University

[26] Zhang J., Shakhnovich E.-I. (2008) Sensitivity-dependent model of protein-protein interaction networks. Phys. Biol. 5 036011 (6pp.)

[27] Zhao J., Ding G.-H., Tai L., Yu H., Yu Z.-H., Luo J.-H., Cao Z.-W., Li Y.-X. (2007) Modular co-evolution of metabolic networks. BMC Bioinformatics 2007, 8:311

[28] - Zhenjun Hu, Joe Mellor, Jie Wu, Takuji Yamada, Dustin Holloway, Charles DeLisi. (2005) VisANT: data-integrating visual framework for biological networks and modules. Nucleic Acids Research 2005, Volume 33(Web Server Issue):W352-W357

[29] - Zhenjun Hu, David M. Ng., Takuji Yamada, Chunnuan Chen, Shuichi Kawashima, Joe Mellor, Bolan Linghu, Minoru Kanehisa, Joshua M. Stuart,

Charles DeLisi. (2007) VisANT 3.0: new modules for pathway visualization, editing, prediction and construction. Nucleic Acids Research, 2007.
[30] - Zhenjun Hu, Jui-Hung Hung, Yan Wang, Yi-Chien Chang, Chia-Ling Huang, Matt Huyck, Charles DeLisi. (2009) VisANT 3.5multi-scale network visualization, analysis and inference based on the gene onthology. Nucleic Acids Research, 2009, Vol. 37, Web Server issue W115-W121
[31] –Zinovyev A., Viara E., Calzone L., Emmanuel Barillot. (2008) BiNoM: a Cytoscape plugin for manipulating and analyzing biological networks. Bioninformatics applications note. Vol. 24 no. 6 2008, Page(s): 876-877
[32] Zinovyev A., Calzone L. Binom Manual Version 1.0. InstitutCurie.
[33] Vazquez A. (2003) Growing network with local rules: preferential attachment, clustering hierarchy, and degree correlations. Phys Rev E Stat Nonlin Soft Matter Phys 67: 056104
[34] Watts D.J., Strogatz S.H. (1998) Collective dynamics of ‘small-world’ networks. Nature, Vol. 393, No. 6684. (04 June 1998), pp. 440-442.
[35] Wilson R.J. (1972) Introduction to Graph Theory. Oliver and Boyd. Edinburgh, 1972

SIMULATION IN ENERGY

SPATIAL POWER DISTRIBUTION SIMULATION FOR NUCLEAR REACTOR

Janos Sebestyen JANOSY
Head of Simulator Development Department
MTA KFKI Atomic Energy Research Institute (AEKI)
Budapest, Hungary
jjs@aeki.kfki.hu

Andras Kereszturi, kere@aeki.kfki.hu
Head of Reactor Analysis Department
Gabor Hazi, gah@aeki.kfki.hu
Jozsef Pales, palesj@aeki.kfki.hu
Simulator Development Dept., MTA KFKI AEKI
Endre Vegh, vegh@aeki.kfki.hu
Atomic Energy Engineering Ltd

ABSTRACT
Fuel assemblies are very expensive parts of the nuclear reactor. Initially they were used in Hungary for 3 years, now for 4 years and soon they will stay in the core for 5 years. Each year only 1/3rd, 1/4th later 1/5th of them is replaced, therefore the change of the fuel type is a lengthy process, with mixed cores used. The authorities require that the staff should be trained to each particular core before they operate it. For this reason the simulator should be upgraded to simulate the exact behavior of each core foreseen for the next 5 years. The RETINA code (Reactor Thermo-hydraulics Interactive) is a 3D offline code, developed in our department.[5] KIKO3D - Neutron Kinetics 3D - has been developed in our Institute, too, in the Reactor Analysis Department [4]. Both of them should be integrated into our full-scope replica simulator, coupled, and stressed to operate parallel in real-time, using four hi-power processors [6]. The simulation-specific details are discussed in the paper.

Keywords: NPP simulation, parallel processing, real-time simulation, coupled thermo-hydraulics and neutron kinetics simulation

INTRODUCTION
Fuel elements, integrated into fuel assemblies produce heat in the nuclear reactors in rather difficult, harsh conditions. The pressure and temperature is high - up to 160 bar and 320°C - and the power density in some reactors reaches 90 kW/liter. They are made from expensive metals using expensive technologies. They should not leak - the cladding represents the first barrier between the radio-active materials and the environment (usually there are at least three barriers). If there is a remarkable leak, the reactor should be stopped and the leaking fuel assembly replaced - a procedure causing significant economic loss.

Nevertheless, some fuel assemblies are well made and they practically never leak. During the 20-year-history of the four-unit Paks NPP there was detectable leak only once or twice. The fuel elements originally spent three years in the core, nowadays they stay for four years - with slightly higher uranium content, of course. If they should stay for five years, the increasing of the enrichment is not enough - the control system of the reactor is not designed to cover the excessive reactivity of the core, produced by the higher enrichment of the fresh fuel.

The solution is the Gadolinium (Gd) which is a burnable neutron poison. In the first year - or so - it helps to cover the excessive reactivity by absorption of neutrons, then it burns out and do not causes any problem in the upcoming years.

Now we replace every year 1/4th of the fuel elements with fresh ones. If we start to replace them with the new types, supposed to stay for five years, it means that we are going to use mixed cores at least for four years. These cores need special treatment and the operators should be trained to it. The core surveillance system must be fitted to these mixed cores, too.

Up to now the reactor model of the full-scope simulator was rather simple - because the related tasks of the operators was simple, too. To train the operators to their more sophisticated duties we had to replace the reactor model and the model of the primary circuit with more elaborated 3D spatial models.

NEW MODEL REQUIREMENTS
We have 349 fuel assemblies in the core, each of them can be of different age and different composition. The core configuration is carefully optimized each year to ensure that the power distribution and burn-out corresponds to the maximal safety and to the best fuel economy.

The water flowing through the core of PWRs acts not only as coolant but as moderator, too - that means if fulfills the task of slowing the neutrons down in order to optimize the neutron balance and making the chain reaction stable and possible. Careful design of the reactors results in negative temperature and volumetric coefficients, that means that the reactor is capable to self-regulate its power - because making the coolant hotter and thinner means worse neutron balance and therefore it decreases nuclear power.

These effects make the neutron kinetic model of the reactor and the thermo-hydraulic model of the primary cooling circuit tightly coupled, therefore they must be solved simultaneously. Describing very different physical phenomena they contain very different equations - that causes the problems of the simultaneous numerical solution. The structure of the primary circuit - reactor, loops with pumps and valves, steam generators - is shown of Fig. 1.

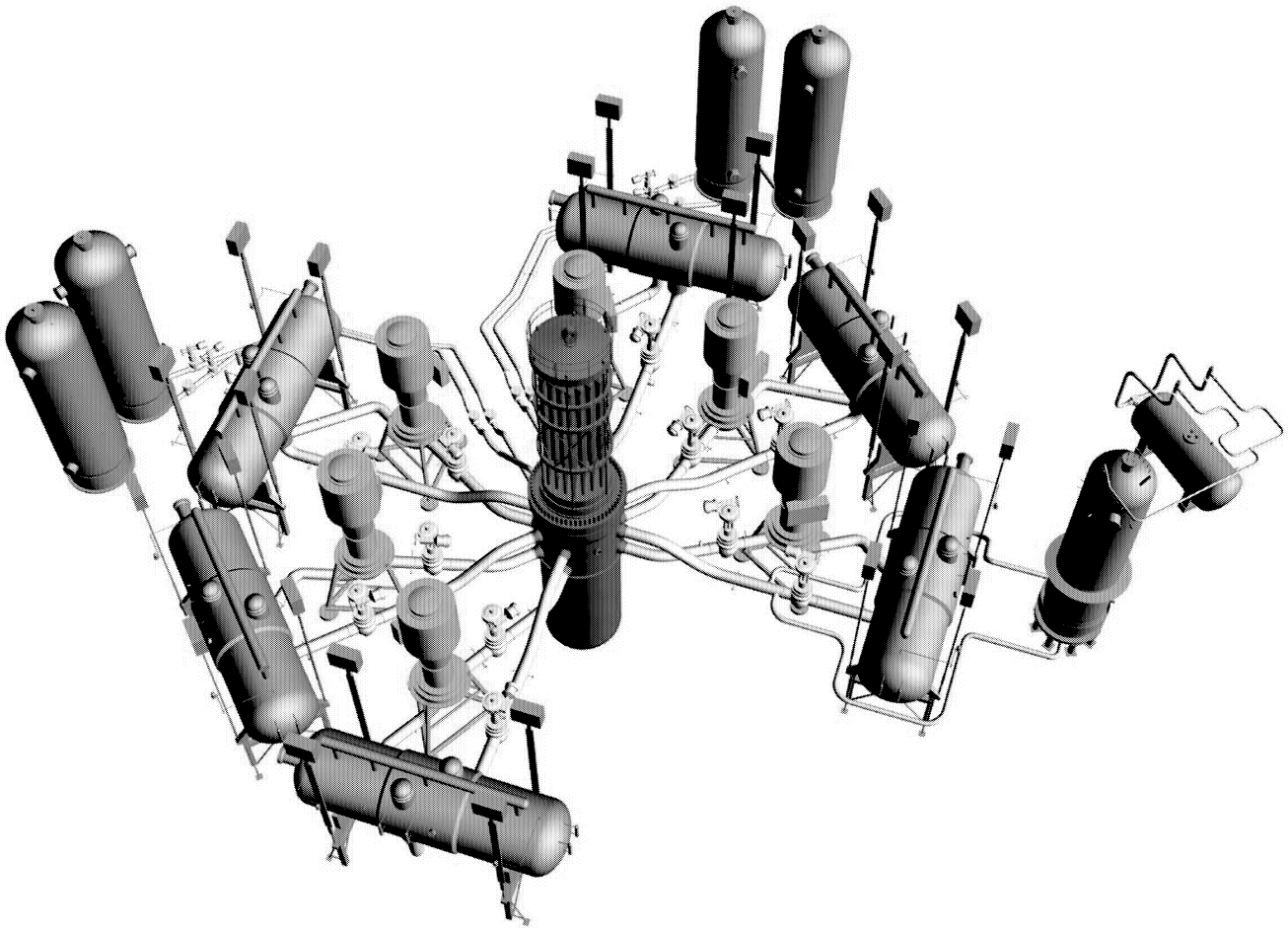


Fig. 1. Primary circuit of a 6-loop Pressurized Water Reactor

The crucial point is: how to nodalize the nuclear reactor and the primary circuit in order to achieve high fidelity of simulation with reasonable computer loads - in other words achieving accurate simulation still remaining in real-time. It looks easy to divide the equipment to very small parts, and solve the problem using them as coupled nodes. Decreasing the size of the individual nodes not only increases their number according to the third power, but in the same time it significantly decreases the necessary time step of the numerical integration.

NODALIZATION PROBLEMS: NEUTRONICS

As it is shown on Fig. 2, we have in the core 349 hexagonal fuel assemblies (the numbers outside the core refer to the six cooling loops). The 37 numbered fuel assemblies are used to control the chain reaction. They are twice as long as a normal fuel assembly. The upper part is made from special steel designed do absorb the proper amount of neutrons in order to be able to control the chain reaction. The lower part is an usual fuel assembly containing usual amount of fuel. ***Pulling out*** this control assembly means that the lower part enters the core, ***lowering it*** causes this part to leave and to be replaced by the neutron absorber assembly.

The 37 control assemblies are organized into 6 groups, containing 6 assemblies except the 6th one, which contains 7 (this 7th is the central one). The first five groups with 30 assemblies are used as the "safety rods", fully pulled out during normal operation and fully lowered during reactor shut-down. The 6th group is normally used as "control rods", during normal operation they are always in different intermediate positions according to the prescribed power of the reactor. In some very rare situations the 5th group is helping to the 6th one, sometimes staying in intermediate position, too.

That evidently means that the first four groups do not influence the spatial distribution of the neutrons, they absorbents are pulled out and their fuel assemblies are inserted. Lowering them the reactor is shut down, and the spatial distribution is not interesting any more. In the same time, the last two groups - the 5th and the 6th - can seriously influence the 3D distribution of the neutrons, being in different intermediate positions according to the different operating conditions of the reactor and the primary circuit

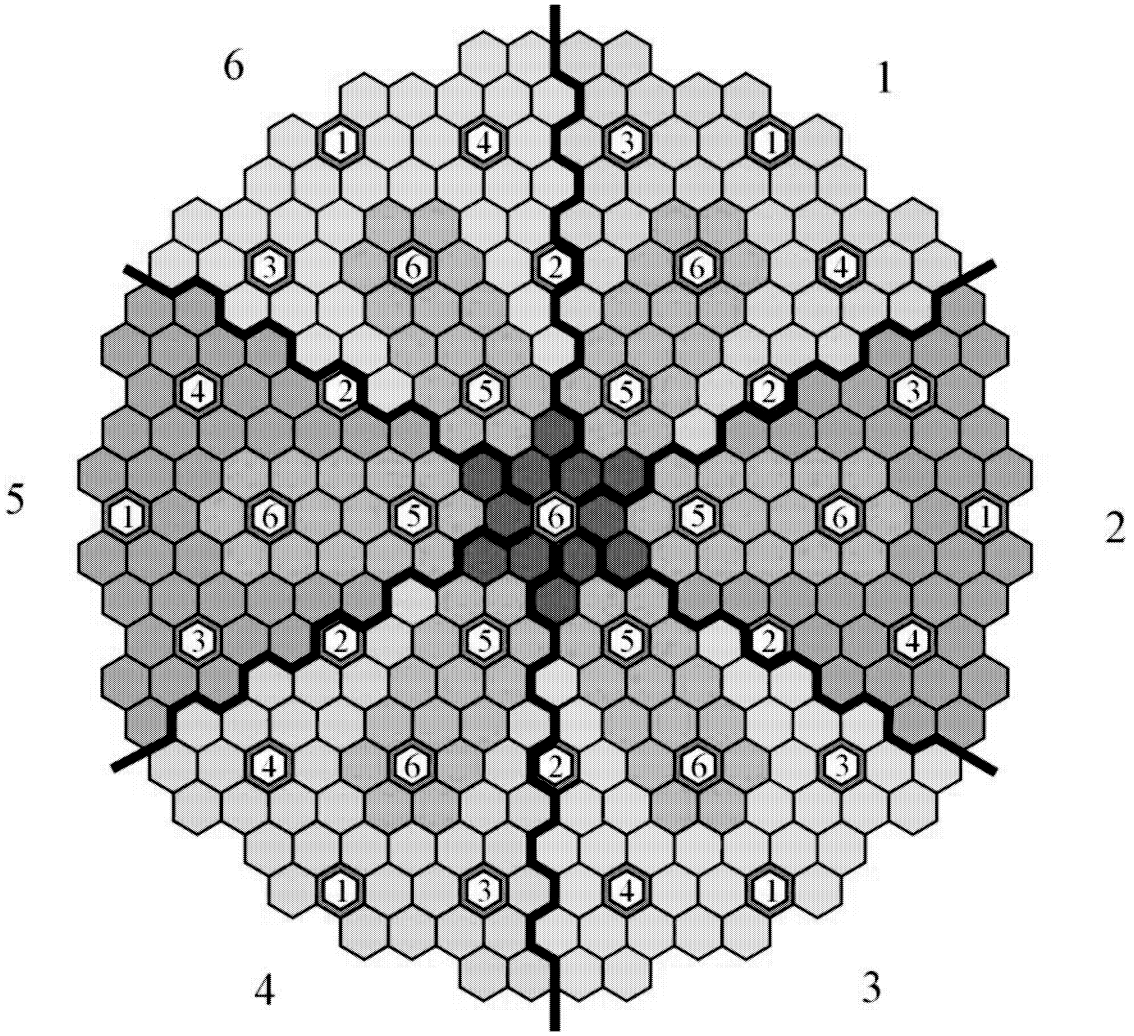


Fig 2. The map of the core with the 349 fuel assemblies, including the 37 control ones.

The nodalization of the core from the neutron kinetics point of view does not leave us too much freedom: each "neighbor" to each assembly can be of different "age" in the reactor (zero to four, later zero to five years), with or without Gadolinium content accordingly. Different "age" means different burn up, thus different stage of enrichment and different isotope content. That means that in horizontal plane each assembly should be a separate node.

As to the vertical nodalization, we must have not less than 8 or 10 planes to get enough resolution (8 to 10 points) to describe the axial neutron (and heat) distribution. We have chosen 10 planes vertically - that means, we have finally 349 x 10 nodes for the KIKO3D model.

Real-time spatial (3D) simulation of 3490 nodes in several groups of neutrons according to their actual energy requires huge computer power. The only way to do it using finite number of processors means to separate the time and space problem. The result can be written as a product **of two functions:** the amplitude ***function of tim***e, and the distribution ***function of space***.

Moreover, the amplitude function can change quickly, much faster, than the spatial form, the distribution of the neutrons (and the power) in the core. That means that if we calculate the actual average power as a function of time in every 0.2 seconds, we can calculate the new shape function, the spatial distribution of neutrons even every 1-5 seconds - it will give still satisfactory results for the real-time simulation.

NODALIZATION: THERMOHYDRAULICS

Thermo-hydraulic nodes should be much larger in space than the neutron-kinetic nodes. It is connected with the 0.2 sec. time step of the full scope replica simulator of the power plant. If we want to avoid large number of iterations, the amount of the steam/water leaving/entering the node each time step must be probably less than the full amount of the steam/water inside the node. It means, that if we multiply the maximal feasible volumetric flow-rates with the 0.2 sec. integration time step, we get the minimal volumes for the nodes in question. Fig 3. shows a vertical plane of the core nodalization.

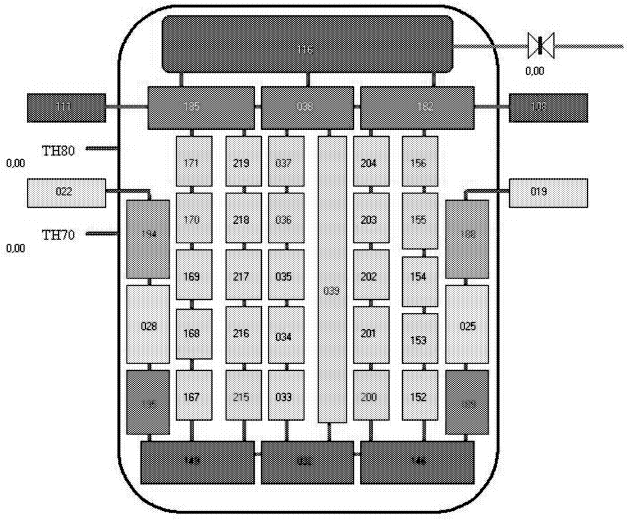


Fig 3. Debugging tool - core nodalization

Creating relatively large nodes we have to group fuel assemblies very carefully, to get as detailed simulation results as possible. The color coding of Fig. 2. shows our results.

The central (red) node on Fig. 2. contains 13 fuel assemblies, including the central "control rod" (from group No. 6). Six inner (green) nodes contain 16 fuel assemblies including 2 control assemblies (one form Group 5 one form Group No. 6) The peripheral six large nodes, shown in different colors, contain 40 fuel assemblies each. (We have the downcomer, the upper and lower plenums and the dome, and a bypass channel (reflector to the core) of the reactor vessel, too, not shown on the Fig. 2. because these nodes are not coupled to the neutron-kinetic model of the reactor.

This kind of thermo-hydraulic nodalization provides the following benefits:

- Only control rods of the 5th and 6th control rod group may have intermediate positions, influencing the spatial distribution of the neutrons. The inner 6 nodes and the central node are responsible for the calculation of these effects.
- One or more cooling loops may fail, usually because of the tripped main circulating pumps (MCPs). The six outer large nodes can respond spatially to these effects.

The original instrumentation of the nuclear power plant can not show the power, the pressure, the temperature and the steam content of the water in each simulation node of our simulator. It is not necessary to measure these parameters in such detail for the operation of the plant. It means, that the full/scope replica simulator has no tools to follow the actual values in these nodes. For development and debugging we had to develop special debugging tools (programs) to display these data on different windows on the screen in order to follow the performance of our model programs closely.

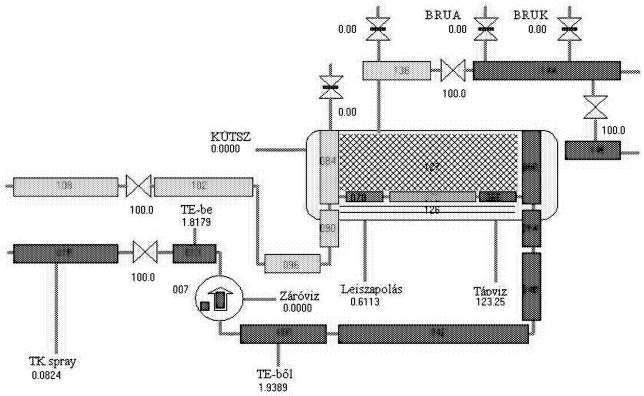


Fig 4. Debugging tool - Loop nodalization

One of these windows is shown on Fig 3. The cold and hot legs of Loop No. 1 and No 2 can be seen on left and right. Inside the vessel close to the wall both side we can see the downcomer, delivering the coolant to the lower plenum (nodes 149, 32 and 146). Passing the core, the coolant reaches the upper plenum (nodes 185, 38 and 182) and the dome (116). The bypass channel is 39, the central node of the core starts with 33, the inner nodes (for loops 1 and 4) with 215 and 200, the outer nodes with 167 and 152. As we can see, vertically we have not 10 nodes, as for the neutron kinetic model but only 5. The color coding of this picture is chosen to show the pressure now (lower window shows "PR").

In the same way, special pictures show the state of each individual cooling loop (see Fig. 4.) Separate picture is dedicated to the state of the pressurizer (not shown).

Thanks to the nodalization scheme described above, different spatial effects in the core can be studied. As an example, the "rod drop" malfunction is presented.

If a control rod erroneously drops into the core, the negative reactivity caused by it can be compensated by the power controller, pulling all the other rods a little out from the core. However, the power locally will be less around the fallen neutron absorber.

All well-designed reactors are self-regulating, that means overheating causes negative reactivity thus decreases the heat power, and overcooling does the opposite - it leads to positive reactivity and the power increases a little. This effect compensates the locally introduced (by the fallen rod) negative reactivity, and that's why the distortion of the power field - and the resulting temperature field - is not so strong than it could have been expected, not taking into account the temperature effects and the resulting self-regulating features.

However, the resulting asymmetry of the temperature field is remarkable, as it can be seen on Fig. 5.

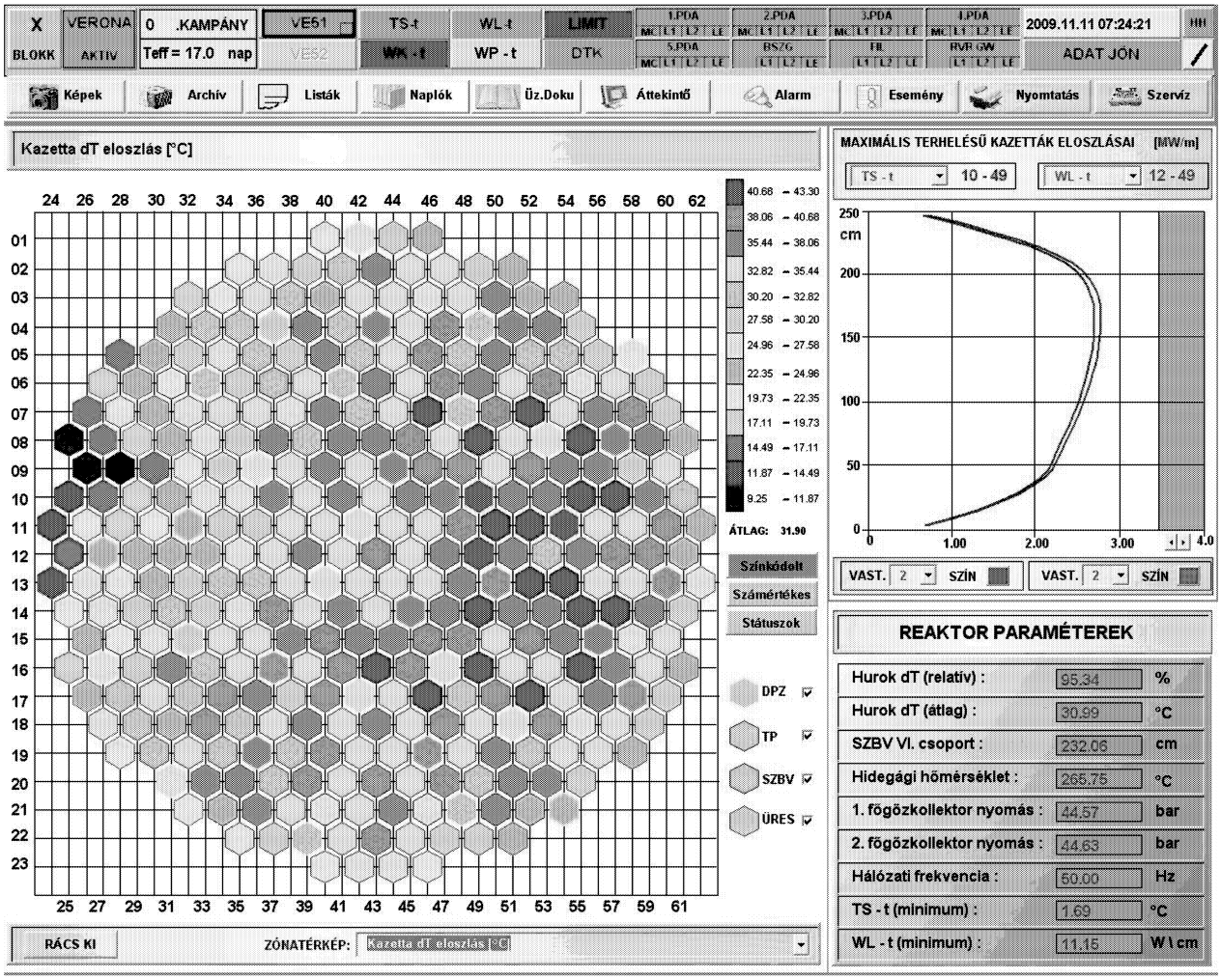


Fig. 5. Picture on the in-core surveillance system VERONA - driven by "rod drop" state data from the simulator

REACTOR THERMOHYDRAULICS

Normally the primary circuit of a pressurized water reactor is filled up with water and there is no boiling. Nevertheless, we have the pressurizer with the steam cushion, the secondary parts of the steam generators with boiling, the steam headers, therefore we have to construct a two-phase-flow (steam, water) model anyway.

Moreover, in case of LOCA (Loss of Coolant Accident) if the water flows out through the break of some pipelines, the emergency core cooling circuits step in, pump water into the primary circuit and the reactor vessel in order to keep the core covered with water and cooled. Air and other non-condensable gases may enter the primary circuit. During the startup of the plant the initial pressure is reached by nitrogen cushion in the pressurizer. Because these states the simulation model should handle not only ***water and steam***, but ***non-condensable gases*** (third component.)

The best tool to simulate these states is the so called "6-equation" model - energy, mass and momentum balance equations for steam and water separately, but with enabled state changes (boiling, condensation). (The non-condensable gases usually are added to the steam phase but state change is disabled for them).

Solving 6-equation models in real time is an exceptionally demanding task, requiring very powerful computers. Things getting much simpler using 5-equation models (common equation for the momentum of steam and water) but in this case the water and steam velocity should be the same - these models are accurate only in case of low steam content (emulsion flow). Higher steam content or stagnant flow causes separation and different speeds. The commonly used solution is the so called "5½-equation" model - one momentum equation but so called "drift flux model" which allows different speed for water and steam but handles it with algebraic approximations.

The most demanding task during the tuning and V&V of the simulator is to tailor these drift-flux correlations to behave correctly in very different circumstances - stagnant, layered coolant in the pressurizer vessel, isolated loop, etc

We have very different situations - normal operating and transients, stopping, cooling down, heating up, reaching criticality of the reactor, loading up, separating loops with main gate valves (a rare feature possible only with Soviet/Russian reactors). There real art during the V&V of the models lies in handling and tuning of the drift flux correlations (nodes above each other, separating, nodes horizontally following each others, pipes and stagnant coolants in vessels, etc.)

As non-condensable gases are carried with steam, boron acid solvent (used to absorb neutrons) and radionuclides carried with water (in case of leakage) are simulated, too, but this task is much simpler than the correct simulation of the two-phase-flow. (However, "aging", decaying radionuclides continuously are changing the concentration of different isotopes and this has to be taken into account, too.

The simulation of the normal operating modes of the power plant are the less demanding for the stability of the numerical integration of the models, and in the same time we have ample data and recordings to fit the parameters of our models. On the other hand, the training of the operators to anticipated (but rarely happening) transients and accidents is the most important and valuable, but these are difficult physical states with no data (so far, so good) to verify and validate the simulation.

The scope of simulation - by definition of the full-scope, replica simulators - covers anything that can happen to the filled-up primary circuit with the closed reactor vessel. That means

- normal operation with normal transients (load changes, frequency regulation)
- bringing to power (heating up, reaching criticality with the reactor, producing steam, speeding up the turbines, reaching the nominal power
- stopping the reactor, cooling down, changing to natural circulation without the pumps, etc.
- malfunctions of any kind, pump trips, valve jams, valve leaks, short circuits, failures of electrical supplies, island mode of operation (separation from the electrical grid)
- accidents up to the design basis accident, which means a large-break LOCA - breaking of the pipe of the pipeline of the main cooling loop.

According to the integration of the models described above, all these transients can be studied "4D" in the reactor - that means in time and space, too.

COMPUTER POWER REQUIRED

The programs of the Paks NPP full-scope replica simulator now are running on a HP Integrity Itanium server with four processors. The original simulator - before the upgrade - required only 15% of CPU time of one processor.

The operation system used is Open VMS with FORTRAN 77 and FORTRAN 90 compilers (solution of equations), C and C++ with X11 (communication and data presentation purposes).

After the careful tuning and verification the RETINA code (thermohidraulics) requires 85% CPU time of one processor, the KIKO3D (neutron kinetics) 75% of the second, the rest of all real-time simulation tasks app. 20% of the third. (The KIKO3D is divided into two sequentially executed programs: KIKOSIM calculates the amplitude, KIKOSHP the spatial shape of the neutron flux.) The fourth processor in theory is completely idle. (of course, the system distributes the tasks between the processors more evenly.

CONCLUSIONS

After accomplishing a two-and-a-half year project, we are able to simulate all important processes of our pressurized-water power reactor practically in all necessary states and in all necessary detail in order to train our operators to the upcoming new fuel and the corresponding new procedures. It is very important that the extremely fast-growing computer power can be used not only to enhance the quality of the GUI but the more detailed and correct simulation of sophisticated systems.

ACKNOWLEDGMENT

This long work could not be successfully accomplished without the brilliant knowledge and grateful assistance of the instructors and other personnel of the Paks full-scope replica simulator. The help and cooperation of György Nagy, Laszló Dercze, Sándor Borbély, Sándor Czekmeister, József Göttli and others was essential to accomplish this work.

REFERENCES

[1] J.S. Janosy: Modeling and Simulation of Nuclear Energy in Eastern Europe. Business and Industry Simulation Symposium, 2003 Advanced Simulation Technologies Conference, Orlando, Florida, March 30 - April 03, 2003

[2] Janos Sebestyen Janosy: Simulation Aided Instrumentation and Control System Refurbishment at Paks Nuclear Power Plant. First Asian International Conference on Modeling and Simulation, AMS 2007, 27-30 March 2007, Phuket, Thailand, ISBN 0-7695-2845-7

[3] Janos Sebestyen Janosy: Simulators are the key for large-scale Instrumentation and Control System Refurbishment Projects. Keynote speech, Second Asian International Conference on Modeling and Simulation, AMS 2008, May 12-15, 2008, Kuala Lumpur, Malaysia.

[4] A. Keresztúri, Gy. Hegyi, Cs. Maráczy, I. Panka, M. Telbisz, I. Trosztel and Cs. Hegedűs, Development and validation of the three-dimensional dynamic code - KIKO3D, Annals of Nuclear Energy 30 (2003) pp. 93-120.

[5] Gábor Házi, Gusztáv Mayer, István Farkas, Péter Makovi and A. A. El-Kafas: "Simulation of a small loss of coolant accident by using RETINA V1.0D code", Volume 28, Issue 16, November 2001, Pages 1583-1594

[6] István Farkas, Gábor Házi, Gusztáv Mayer, András Keresztúri, György Hegyi and István Panka, "First experience with a six-loop nodalization of a VVER-440 using a new coupled neutronic-thermohydraulics system KIKO3D-RETINA V1.1D" Annals of Nuclear Energy, Volume 29, Issue 18, December 2002, Pages 2235-2242

MODEL BASED REAL TIME AUTOMATED TEMPERATURE CONTROL SYSTEM FOR RISK MINIMIZATION IN HONEY BEE WINTERING BUILDING

Aleksejs Zacepins
Jurijs Meitalovs
Egils Stalidzans
Department of Computer Systems
Faculty of Information Technologies
Latvia University of Agriculture
Liela iela 2, Jelgava, Latvia
E-mail: a.zacepins@inbox.lv

Keywords

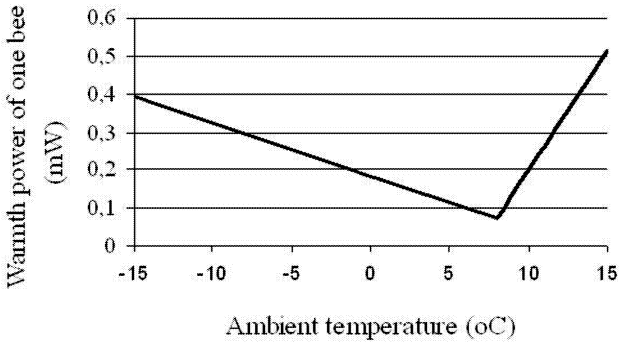
Programmable logic controller (PLC), temperature control system, risk minimization, honey bee wintering building.

Abstract

During the winter time honey bees can be placed in a special wintering building to save physiological resources of bees and honey compared with outdoor wintered bees. This building should have stable and convenient microclimate conditions for bees. The most important parameter of microclimate to be controlled is temperature. For our bee wintering building we have developed a special temperature control system, using a programmable logic controller (PLC) that is permanently connected via Internet to the model based risk management system. The developed system helps to minimize honey bee wintering risks. The possible risks are the start of intensive brood rearing before early spring and the risk of control system failure. To minimize technical system errors *SMS* and e-mail notifications are used.

Introduction

Our task is to maintain stable climate conditions in constructed experimental honey bee wintering home. At the moment the single measured and controlled parameter is temperature, but in the future the count could be extended with other needed parameters, for example humidity, pressure etc. Temperature is the primary climate parameter and during the winter time it is needed to maintain the constant temperature in the building. The ambient temperature of +8°C corresponds to the lowest warmth power produced by bees (see Figures 1) and therefore it is the desired temperature in the building.



Figures 1: Dependence of bee warmth power from ambient temperature (Jeskov 1983)

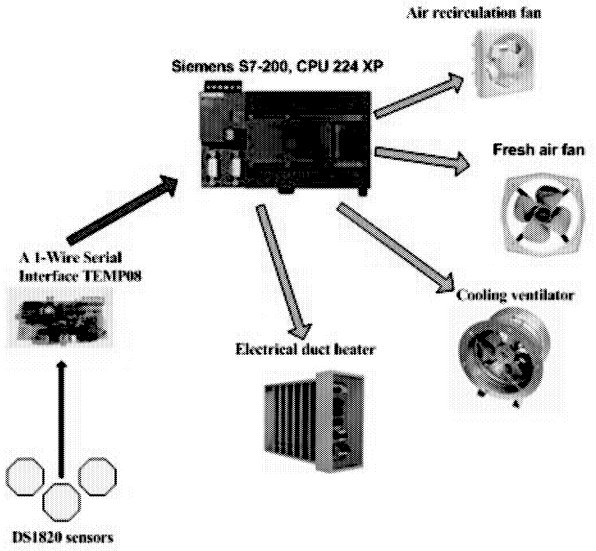
If temperature is on this level, honey bees are consuming the least amount of feed. Sharp fluctuations in temperature have negative effect on bees. As a result of such fluctuations the density of the bees cluster could change, crystallization of honey in the comb accelerates and dampness could be induced in the hive. Significant temperature increase has the most negative effect on bees. This can lead to a situation, when bees would come to the active state of intensive bee breeding too early and bee queen would start to lay eggs. Too early start of bee breeding in the winter can cause high energy consumption of bee colony that, in turn, can lead to a possible death of that colony. That is the main risk factor during the wintering in the special building. In case of increased activity of colonies the target temperature should be decreased according to a model.

Wintering building and temperature control system

The honey bee wintering building is a building with size 6,25m x 4,00m x 2,4m made from sandwich panels with wall thickness of 80 mm and the roof 100 mm (Osadcuks and Stalidzans 2008). The developed control system is based and controlled by the PLC. In our case we are using Siemens S7-200 controller with CPU 224XP (Berger 2000). PLC is a digital computer used for automation of electromechanical processes and it is designed for multiple input and output arrangements. PLC controls other equipment in the building (see Figures 2). The controlled equipment is:

- Air recirculation fan for homogenising the air and its temperature inside the building;
- Cooling ventilator system;
- Fresh air fan;
- Electrical duct heater.

The main task of the system is to control the temperature changes. Temperature is measured, using the DS18S20 sensors. Duct heater is functioning using the PLC integral PID algorithm. The air is heated till the target temperature is reached. PLC user program is written using the Ladder Diagram programming language. (Berger 2001)



Figures 1: Temperature control system elements

Risks minimization of bee wintering

Bee wintering buildings generally do not need human attendance during the winter and therefore automatic control system usually functions unattended for a long period of time - weeks or even months. PLC controls all other control system components, such as sensors, heaters, fans and other. Several risks have to be considered: 1) start of intensive brood rearing before early spring and 2) failure of the control system itself.

If intensive brood rearing of colonies starts during the winter (instead of early spring) the whole bee colony could die. Another risk is the failure of the control system elements. The death of bee colonies can occur due to overheating, overcooling or lack of oxygen.

The detection of the risk of early start of brood rearing can be done placing temperature sensors between the bee nest and pillow. The rapid increase of temperature is a signal of a potential start of intensive brood rearing. Never the less, additional calculations using the model of bee colony dynamics should be used to determine the significance of early brood rearing risk. The communication of PLC with the central computer in the office allows using the model located in the office computer and make intensive calculations. These calculations cannot be done on PLC which is installed in the bee wintering building. In case of risk, the target temperature of the wintering building can be decreased automatically or after a confirmation by beekeeper.

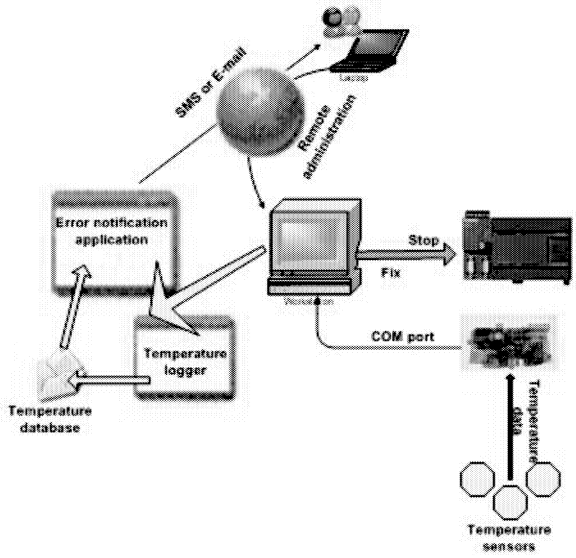
The solution for minimizing equipment failure risk could be realized by applying error notification method. For example, when the temperature reaches the critical limit, it is important to inform the responsible person or persons (beekeepers) about this. The critical temperature limit is defined in the system by a system user. Notification methods that can be realised are *SMS* or e-mail sending. But *SMS* sending is a more efficient way of notification, because beekeepers will receive the error message without any significant time delay, but e-mail could be read after some time, when PC will be used. But in our case the reaction from beekeepers is needed immediately, if they want to save their honey bees.

We have realised the error notification method in such a way. In the honey bee wintering building we have placed four DS18S20 temperature sensors which are directly connected to TEMP08 Serial One Wire Interface. (http://www.midondesign.com/TEMP08/TEMP08.html.)

This device is connected to a PC using the standard serial interface RS232. We have also developed two related PC applications, using the C# programming language. The idea of the first program is to monitor the temperature sensors by receiving data using the COM port of the PC. Program user can change the time intervals to define how often it is needed to receive the temperature data from the sensors. The interval can be, for example, one minute or one hour. Then all received data is stored in the MS Access database. In the program we have also added some visualisation of data, so that the user can see the chart of temperature values. Other program functionality can be added as needed. So this program could be named as temperature logger. (Meitalovs et al. 2009) The second program is named as error notification application. It uses data from the mentioned database. Every time when temperature data is taken from it, it is saved in a simple txt file. Every day a new txt file is created and once a day it is sent to beekeeper's e-mail. This is needed to give him a possibility to check whether the system is working properly. But the main task of the program is to check whether the temperature is within critical limits and if temperature is out of these limits *SMS* is automatically sent to responsible persons for an immediate reaction.

One more feature that was developed is a remote connection to the PC, which is connected to the PLC device. This feature allows the beekeeper to remotely access the logical part of temperature control system. This option is not needed if beekeeper can easily access bee wintering building and fix or change the control system program on site. But if this is not possible or the responsible person is far away from bees, then remote connection is the right way to solve small program problems or fix errors. In our case we are using standard MS Windows remote connection application for remote access to the PC. And beekeeper can access his control system from any other PC. One more reason of using remote connection rather than fixing errors on site is that it is not recommended to use light in the honey bees wintering building, while solving system problems in the dark is not very comfortable and easy.

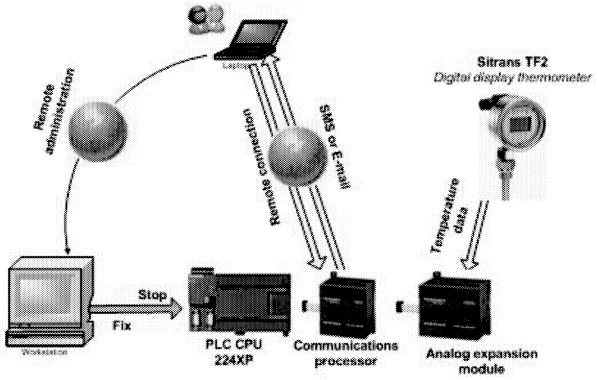
So in that way we are minimizing the risk of system errors (see Figures 3). But of course, there are also other risk factors left. For example Internet connection to the bee wintering building could be broken and then we will not receive any error notification messages.



Figures 2: Realization model for risk minimization of PLC based temperature control system

The future development of the temperature control system

The temperature control system developed by us can be modified and upgraded. The first possible upgrade is to modify error notification system. It would be good to develop it using the PLC communication device, rather than using the separate C# application. So the main idea is to connect PLC device to the Internet using the additional IT communication module (see Figures 4). This will give a possibility for the beekeeper to remotely control the PLC device using its WEB interface. And *SMS* sending process also could be done by this module. The second upgrade could be the measuring of additional parameters in the wintering building. One of the most important additional parameters to be considered is humidity, but the ideal relative humidity for a wintering chamber has not been determined. Some tests show that bee colonies are very adaptable to a wide range of relative humidity situations. Satisfactory level of relative humidity in the wintering building is in the range of 50-75%. (Furgala and McCutcheon 1992) When humidity is too low, bees are suffering from thirst and to quench it, bees have to consume honey strenuously, but this could lead to diarrhoea disease. But when humidity is on high level, crystallization of honey accelerates and bees become very loud, crawl out of the hive and die. Also mice cause great concern for bees, and that is way it is needed to detect them on time. To complete this task some floor movement sensors could be used.



Figures 3: The future development of the temperature control system

Conclusions

The developed temperature control system is used for maintaining stable microclimate in honey bee wintering building. The main controlled parameter is temperature, because it has the major impact on bees during the wintering time. The control system is based on PLC, which is used to control equipment needed for heating the air in the building. The controlled equipment consists of air recirculation fan, cooling ventilator system, fresh air fan and electrical duct heater. Temperature data is received using the DS18S20 sensors. The developed temperature control system helps to minimize such honey bee wintering risks as 1) start of intensive brood rearing before early spring and 2) failure of control system. Error notification system is developed to minimize PLC control system risks. Advised notification method is *SMS* sending to one or more mobile phones. E-mail notifications could be used as well.

Academic study and publication financed by the project „Support for doctoral studies in LUA” / 2009/0180/1DP/1.1.2.1.2/09/IPIA/VIAA/017/ agreement Nr. 04.4-08/EF2. D1.09.

References

Berger H. *Automating with SIMATIC*. 2000, 214 pages.

Berger H. *Automating with STEP7 in LAD and FBD*. 2nd revised edition, 2001, 356 pages.

Furgala B., McCutcheon D.M. *Wintering Productive Colonies*. Chapter 20 in the book “*The Hive and the Honey Bee*”. Revised edition, 1992.

Jeskov J. K. *Mikroklimat pcelinogo zilischa*. (Moscow, Rosselhozizdat, 1983, pp.191.

Meitalovs J., Histjajevs A., Stalidzans E. *Automatic Microclimate Controlled Beehive Observation System*, 8th International Scientific Confernce ‘Enginieering for Rural Development’ 28.-29. May 2009 Jelgava, Latvia, pp. 265-271.

Osadcuks V., Stalidzans E. *Metabolic Activity Assessment Using Specialised Microclimate Chamber with Control System*, In Proceedings of Applied Information and Communication Technologies 2008, 10.-11. April 2008, Jelgava, Latvia, pp. 78-85.

http://www.midondesign.com/TEMP08/TEMP08.html.

ADJUSTMENT OF HYDROSTATIC TRANSMISSIONS THROUGH VIRTUAL INSTRUMENTATION TECHNIQUE

Teodor Costinel Popescu
INOE 2000-IHP,
Bucharest, Romania,
E-mail: popescu.ihp@fluidas.ro

Dragos Daniel Ion Guta
INOE 2000-IHP,
Bucharest, Romania,
E-mail: dragos.ihp@fluidas.ro

Alexandru Marin
"POLITEHNICA" University
of Bucharest, Romania,
E-mail: alexandru.marin@upb.ro

KEYWORDS: LabVIEW, AMESim, hydrostatic transmissions

ABSTRACT

Electro hydraulic adjustment systems are complex systems which involve both phenomena regarding fluid flow, in the field of volumetric hydraulic transmissions, and phenomena specific to automatic adjustment processes. As a result of complexity of such phenomena, optimal solutions for designing and developing automatic electro hydraulic adjustment systems are designated iteratively, generally by means of dedicated languages. With support from process identification methods and power of numerical computing equipment, researchers and designers can shorten the development period of applications in various fields by generating more realistic solutions even from design stage.
The paper presents a hybrid solution of modeling / simulation of a hydrostatic transmission, with mixed adjustment, in open circuit. Starting from a known mathematical model of hydrostatic transmission, by means of two distinct simulation environments, AMESim and LabVIEW, an adjustment model of the system was developed, optimized and experimentally validated.

INTRODUCTION

Latest trends in modeling and simulation of dynamic systems regard new concepts, such as co-simulation and real-time simulation.
The concept of real-time simulation of dynamic systems allows simultaneous existence of both a part of a mathematical model, and a part of a physical model, thus obtaining a decrease of the degree of uncertainty, caused by certain phenomena neglected in the mathematical model.
Co-simulation involves simultaneous use of resources from multiple modeling /simulation environments, with the opportunity to enjoy maximum performance of each environment.
The concepts of co-simulation and real-time numerical simulation are not newly developed concepts. Due to their potential and rapid development time, numerical computing systems have begun to join in the industrial area of control and radjustment relatively recently after their appearance. However, because of the specific language used by these machines, their command and programming were a long time the prerogative of specialists in informatics. Obvious need of their use by a much larger range of users, especially those in scientific areas involving mathematical calculations, led to the evolution of specialized interpreters, that is easy control / programming languages of computer equipment.

The article presents an example of achieving a numerical co-simulation network based on two programs: AMESim and LabVIEW. With AMESim program a hydrostatic transmission model was developed (Popescu et al. 2009), while analysis of data obtained from simulations in AMESim has been performed with virtual instrumentation, by means of LabVIEW software.
Simultaneous use of two simulation/ programming environments, AMESim and LabVIEW, can lead to the development of an advanced modeling / simulation network of electro hydraulic systems.
Modern methods of experimentation in the field of hydraulic and pneumatic systems imply the existence of at least one numerical computing equipment. Need to use electro-hydraulic converters for control and adjustment of various physical parameters such as force, displacement, together with the exponential growth in digital electronics have confirmed this. Digital equipment is found as a component of sensors and transducers, numerical displays, electronic servo amplifiers (compensators) or process computers. Of any modern laboratory equipment for electrohydraulic drives are not missing sensors and transducers with electronic communication interface, systems equipment (proportional electrohydraulic directional control valves, servo pumps / hydraulic or pneumatic actuators, etc.) with analog/ digital control ports and electronic adjustment blocks.
By "intelligent" combining of these devices can be made various installations, equipment, stands with specific functionalities. Coupling of industrial computers to these systems allows flexible "manipulation" of functionality of mentioned systems. The ability to "load" numerical computing systems with "virtual models" of systems, developed using advanced modeling language, enlarges their flexibility even more.
For hydrostatic transmissions three types of terms are known (Drumea et al. 2010): **primary adjustment**, in which the volumetric machine of the primary sector of transmission (pump) has adjustable capacity, while the one in the secondary sector (motor), powered at constant flow, has fixed capacity; **secondary adjustment**, in which the volumetric machine of the primary sector of transmission has fixed capacity, while the one in the secondary sector, powered at constant flow, has adjustable capacity; and **mixed adjustment**, in which volumetric machines in the primary and secondary sectors of transmission are adjustable.
In terms of functional performance and energy savings, hydrostatic transmissions based on secondary adjustment offer advantages in comparison to those based on primary adjustment. But they raise some difficulties, especially if long lines, with more consumers, related to the influence of load variation at some consumers on load variation at other consumers.

PROBLEM FORMULATION

The analyzed hydrostatic transmission, with mixed adjustment, with single consumer of type rotary volumetric motor, adjustable, according to figure 1, contains:
in the primary sector: a MOOG servo pump, place 1, with radial pistons and integrated electronics, with three loops of adjustment, that is in flow, in pressure, in flow and pressure; an electric motor for servo pump actuation, with constant speed, place 2; a pressure limiting valve, place 3; a flow transducer, place 4; a pressure transducer, place 5; a hydro pneumatic accumulator and a flow transducer, place 6.
in the secondary sector: a BOSCH servo motor, place 8, with axial pistons, tilted block and integrated electronics, a torque transducer place 9; a speed transducer, place 10; an axial piston pump, with tilted block and fixed capacity place 11, to simulate the load of hydraulic servo motor, two pressure transducers, place 12 and place 14; four way valves, place 13, fitted on suction / load repression side of pump; a pressure adjustment valve, with electric control, place 15 for adjusting load of hydraulic servo motor.
a PXI-NATIONAL INSTRUMENTS **block**, place 7, which provides a virtual interface of the adjustment process of capacity of the adjustable volumetric machines (LabVIEW / PXI).

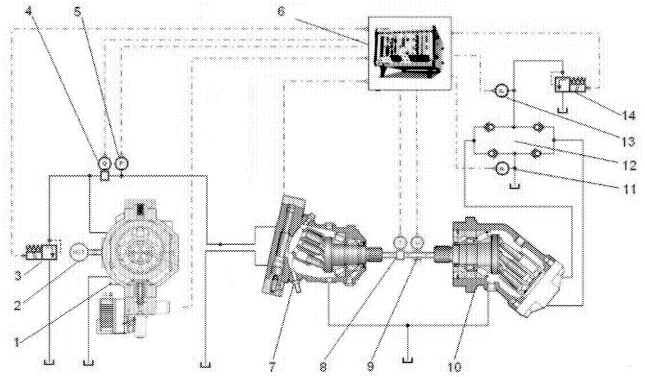


Figure 1: Simulation network of a hydrostatic transmission – basic model

The adjustment model of hydrostatic transmission ensures automatic, succesive change of capacity of both adjustable volumetric machines, depending on the variation of rotational speed at the shaft of hydraulic motor. There is a threshold, programmable, of hydraulic motor's speed value up to which the growth of its rotational speed requires from the control system a drive in order to increase pump capacity, while maintaining maximum capacity of the motor, and going beyond this threshold requires from the control system a drive in order to decrease motor capacity while maintaining pump capacity at the value corresponding to the moment when threshold of rotational speed is hit.
To optimize adjustment model of the analyzed hydrostatic transmission, we have gone iteratively through the following steps:
a) a basic model of transmission (Figure 1) was developed, which was physically materialized in the laboratory of INOE 2000-IHP Bucharest, as shown in figures 2, 3 and 4;
b) the mathematical model of transmission was determined;
c) the simulation model in AMESim of transmission was developed, as shown in figure 6;
d) virtual interface of the adjustment process was developed, in LAbVIEW, as shown in figure 7, which was coupled to the simulation network, developed in AMESim;
e) was simulated the dynamic behavior of the hydrostatic transmission to stage type rotational speed signals prescribed for the hydraulic motor, which involve adjustment control of capacity of hydraulic pump / motor.
f) was experimentally validated the adjustment model of hydrostatic transmission, by decoupling the LabVIEW/ PXI virtual interface from simulation model in AMESim, followed by coupling of this interface to the physical model, inside the laboratory, raising experimental dynamic characteristics and their comparison with the results of simulations.

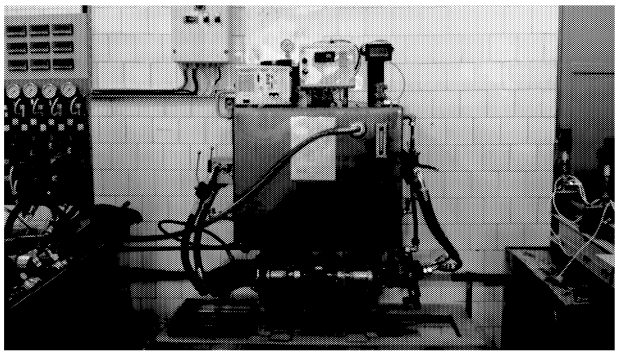


Figure 2: Secondary sector of hydrostatic transmission – laboratory model

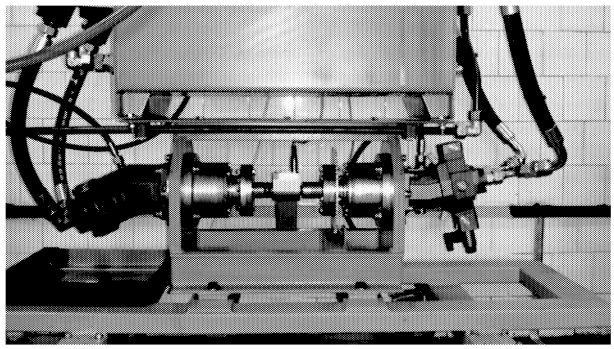


Figure 3: Fixed pump and Bosch adjustable motor in the secondary sector of hydrostatic transmission

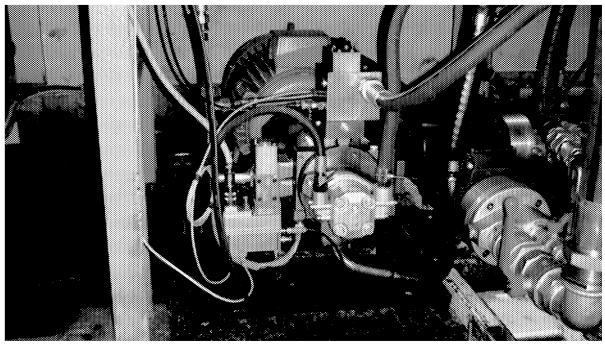


Figure 4: MOOG servo pump in the primary sector of hydrostatic transmission

MATHEMATICAL MODEL OF HYDROSTATIC TRANSMISSION

Mathematical model of hydraulic system shown in figure 5 is characterized by the following equations:

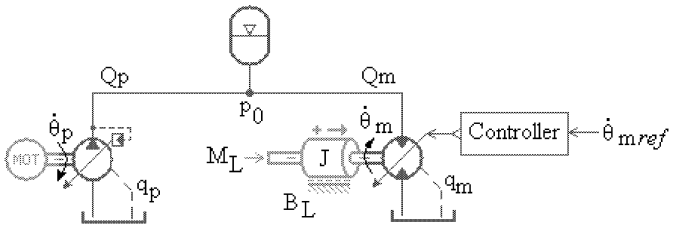


Figure 5: Basic schematic diagram of open circuit hydrostatic transmission

Equation of flow conservation at pump level:

$$Q_p = V_p\theta_p' - C_{pi}(P_0 - P_{tank}) - C_{pe}P_0 \quad (1)$$

$$V_p = \frac{\alpha_p V_{pmax}}{2\pi} \quad (2)$$

$$q_p = C_{pe}(P_0 - P_{tank}) \quad (3)$$

Equation of flow conservation at pressure line level:

$$Q_p - Q_m = \frac{dP_0}{dt}\left(k_e + \frac{V_c}{E}\right) \quad (4)$$

Equation of flow conservation at hydraulic motor level:

$$Q_m = V_m\theta_m' + C_{mi}(P_0 - P_{tank}) + C_{me}P_0 \quad (5)$$

$$V_m = \frac{\alpha_m V_{mmax}}{2\pi} \quad (6)$$

$$q_m = C_{me}(P_0 - P_{tank}) \quad (7)$$

Equation of equilibrium of torques at hydraulic motor level:

$$J\frac{d^2\theta_m}{dt^2} + B_L\frac{d\theta_m}{dt} = V_m(P_0 - P_{tank}) - B_m\frac{d\theta_m}{dt} - M_L \quad (8)$$

Significance of the terms in equations (1)...(8) is the following:

α_p – adjustment angle of pump;

α_m – adjustment angle of motor;

V_p – adjustable geometric volume of pump;

V_m – adjustable geometric volume of motor;

θ_p' – angular driving speed of pump;

θ_m' – angular speed of motor;

q_p – external flow loss from pump;

q_m – external flow loss from motor;

Q_p – flow discharged by pump;

Q_m – flow intaken by motor;

P_0 – pressure on the working circuit;

M_L – load torque (disturbance);

B_L – coefficient of load absorbtion;

B_m – coefficient of friction due to viscosity;

J – inertial torque of mass actuated by the motor;

$\theta_{m\,ref}'$ – angular speed imposed to the motor;

C_{pi} – coefficient of internal flow losses of pump;

C_{mi} – coefficient of internal flow losses of motor;

C_{pe} – coefficient of external flow losses of pump;

C_{me} – coefficient of external flow losses of motor;

k_e – elasticity coefficient of pipes;

V_c –capacity of pipe pomp-motor;

E – elasticity module of working fluid.

Parameters of simulation network are the following:

$N_{electrical\ motor}$=1500 rpm; $V_{p\,max}$ = 32 cm^3/rot;

$V_{m\,max}$ = 28 cm^3/rot; $V_{p\,load}$ = 125 cm^3/rot;

E = 14000 bar; J = 0.01 Kgm^2; B_m = 0.08 Nm/(rev/min).

DETAILED MODEL OF SIMULATION NETWORK AND VIRTUAL INTERFACE OF ADJUSTMENT MODEL

Detailed model of simulation network, as shown in Figure 6, uses components of the AMESim libraries and it contains: adjustable pump from the primary sector of transmission, which ensures supply at constant pressure of hydraulic motor within the secondary sector of transmission; hydraulic servomechanism for adjustment of capacity of this pump, with its pressure source; electric motor for actuation of the pump within the primary sector, adjustable hydraulic rotary motor, within the secondary sector of transmission; hydraulic servomechanism for adjustment of capacity of this motor, with its source of pressure; fixed bidirectional pump, for simulation of load of hydraulic motor; four way valves, which provide suction / discharge of fixed pump; a hydro pneumatic accumulator; a pressure control valve with electric drive, which adjusts the value of load of hydraulic motor; two position transducers; three pressure transducers; a flow transducer; a torque transducer; a rotational speed transducer; a tank of oil; a PXI module.

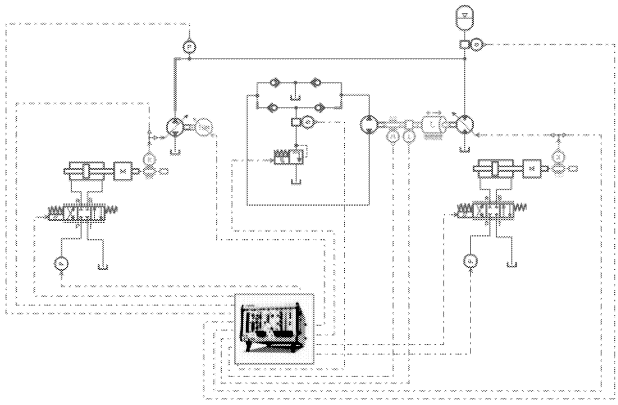


Figure 6: Simulation network of hydrostatic transmission – detailed model

Virtual interface of adjustment process, as shown in Figure 7, provides:

- prescription of supply pressure of the two hydraulic servomechanisms for adjustment of position, which adjust capacity of the two adjustable volumetric machines;
-prescription of value of load of hydraulic rotary motor within the secondary sector of transmission;
- prescription of value of pressure on the pump's discharge circuit within the primary sector of transmission and automatic maintenance of this value, by changing the pump's

capacity depending on the way and size of variation of adjusted pressure;
- prescription of rotational speed of hydraulic rotational motor within the secondary sector of transmission and automatic maintenance of this value, by changing the capacity of hydraulic motor within the secondary sector when its rotary speed tends to increase or decrease, depending on the variation of load in its shaft.
Process computer used runs simulation network (real-time model) and automatic routines needed for prescription of references of numerical model and data acquisition within the process. Acquired data are sent regularly to process visualization system. Using specific interface of the process that can be made on standard PC computer, can be ordered, by an operator, modification of input data of mathematical model.
The connection between the industrial computer and experimental stand is made with dedicated interface boards of PXI system. Link is bidirectional; computer system generates orders and acquires necessary information for closing internal calculation loop (feedback or feedforward). If you want to run the mathematical model (simulator) simultaneously with the adjustment software model of experimental stand, then it should be considered the deterministic character of development of mathematical model.

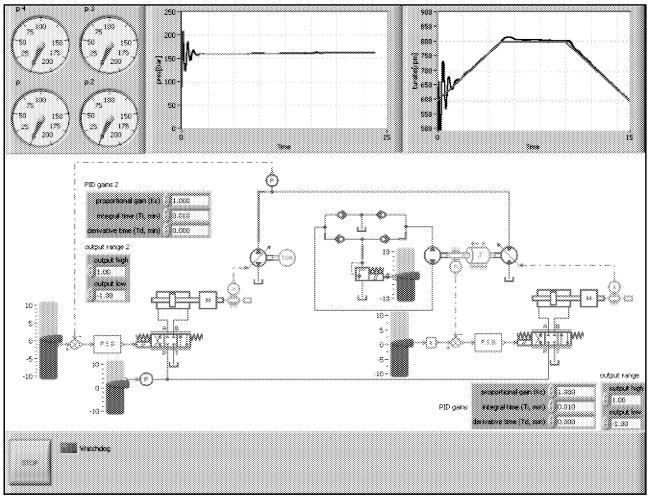


Figure 7: Virtual interface of adjustment model (LabVIEW / PXI)

DYNAMIC CHARACTERISTICS OF HYDROSTATIC TRANSMISSIONS OBTAINED BY NUMERICAL SIMULATION

By running the simulation model of the hydrostatic transmission, developed in AMESim (Popescu et al. 2009), is highlighted the influence of rotational speed step signals required for the adjustable hydraulic motor within secondary sector, upon the adjustment drive of capabilities of the two volumetric machines, with and without error compensation. A reference rotational speed of 320 rpm was established, which when exceeded adjustment system of transmission switches from pump capacity adjustment on hydraulic motor capacity adjustment.

Increasing hydraulic motor rotational speed up to 320 rpm, involves a drive of the adjustment system in order to increase pump capacity. Variation of rotational speed of hydraulic motor, developed as a result of this drive, with and without error compensation, is presented in Figure 8.
Increasing hydraulic motor rotational speed up to 420 rpm, involves a drive of the adjustment system in order to decrease hydraulic motor capacity. Variation of rotational speed of hydraulic motor, developed as a result of this drive, with and without error compensation, is presented in Figure 9.

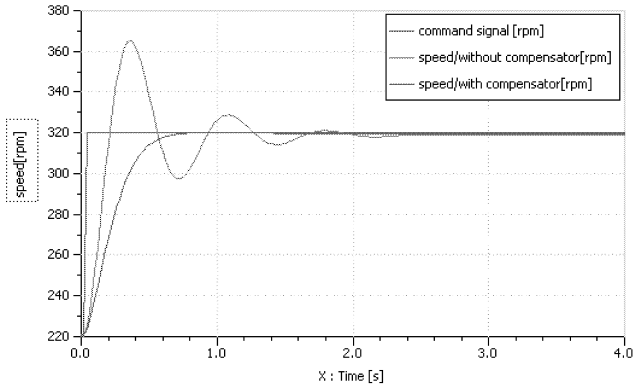


Figure 8: Response of the adjustment system of hydraulic motor rotational speed to step-type excitation signal –pump capacity drive

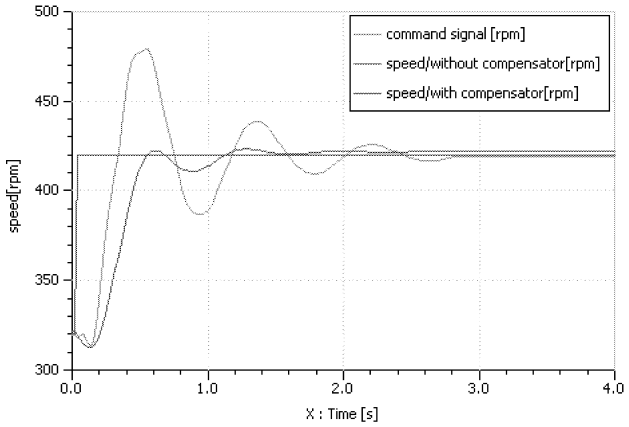


Figure 9: Response of the adjustment system of hydraulic motor rotational speed to step-type excitation signal –motor capacity drive

EXPERIMENTAL VALIDATION OF ADJUSTMENT MODEL OF HYDROSTATIC TRANSMISSION

For experimental validation of the adjustment model of hydrostatic transmission (Popescu et al. 2008), virtual LabVIEW / PXI adjustment interface was coupled to the physical model of transmission. Hydraulic motor within the secondary sector of transmission was excited with step-type and ramp type rotational speed signals.
Response of the adjustment system of transmission was traced; it acts on the servo adjustment mechanism of capacity of the pump within primary sector or on the servo adjustment

mechanism of capacity of the motor within the secondary sector, according to a predefined rotational speed threshold. Were raised dynamic characteristics of the system which highlight:
- influence of a rotational speed step-type signal of 312 rpm, required for hydraulic adjustable motor within the secondary sector, on the adjustment drive of the adjustable pump capacity, with and without error compensation, according to figure 10;

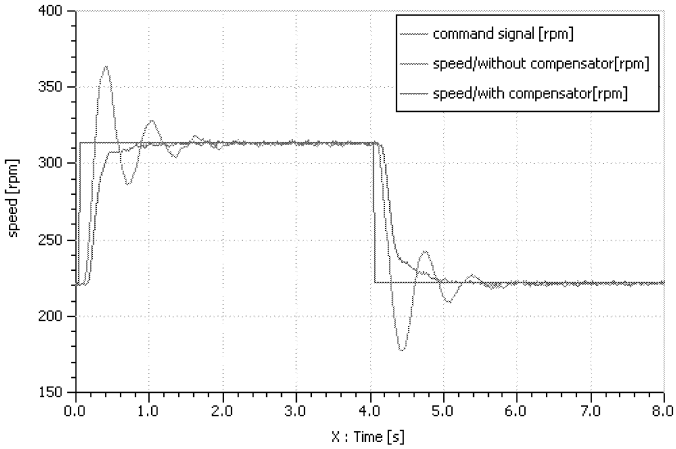


Figure 10: Response of the rotational speed adjustment system of hydraulic motor to step-type excitation signal – pump capacity drive

- influence of a rotational speed step-type signal speed of 410 rpm, required for hydraulic adjustable motor within the secondary sector, on the adjustment drive of the adjustable motor capacity, with and without error compensation, according to figure 11;

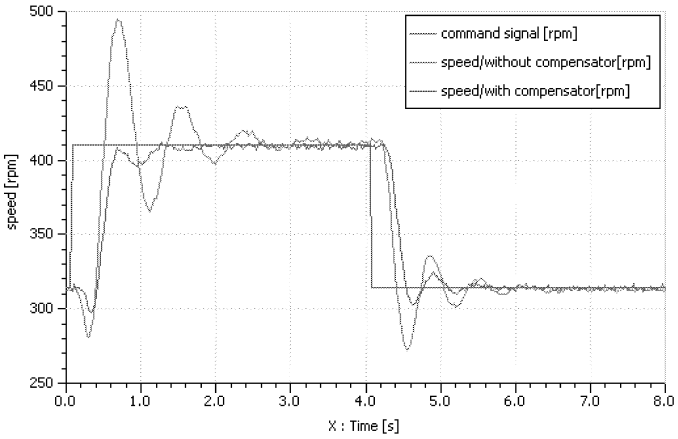


Figure 11: Response of the rotational speed adjustment system of hydraulic motor to step-type excitation signal – motor capacity drive

-influence of combined rotational speed step-type signals, of 308 rpm, respectively of 408 rpm, required for hydraulic adjustable motor within the secondary sector, on the adjustment drive of the adjustable pump capacity, respectively of adjustable motor, with and without error compensation, according to figure 12;
-influence of combined rotational speed ramp-type signals, with maximum value of 315 rpm, respectively of 400 rpm, required for hydraulic adjustable motor within the secondary sector, on the adjustment drive of the adjustable pump capacity, respectively of adjustable motor, with and without error compensation, according to figure 13.

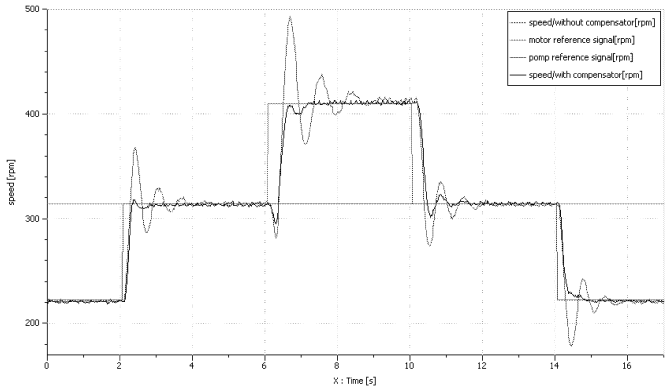


Figure 12: Response of the rotational speed adjustment system of hydraulic motor to combined step-type excitation signal – pump and motor capacity drive

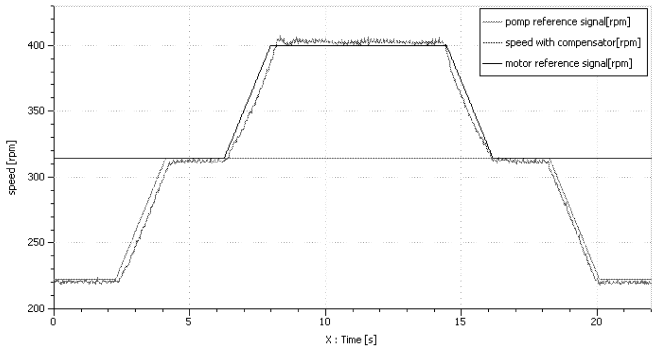


Figure 13: Response of the rotational speed adjustment system of hydraulic motor to combined ramp-type excitation signal – pump and motor capacity drive

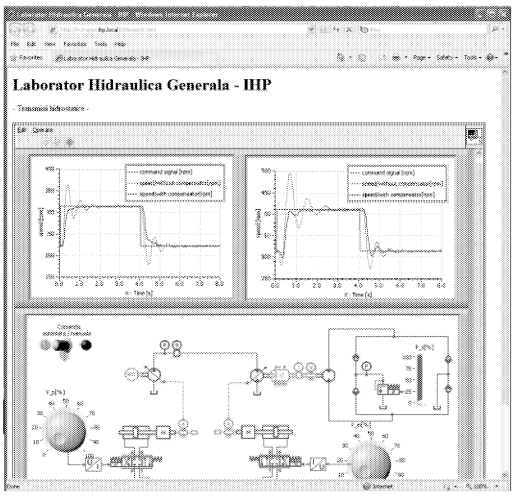


Figure 14: Web interface of the adjustment model (LabVIEW / PXI)

The software application contains a module for conveying experimental results collected on the stand via the Internet using a web page, as shown in Figure 14. The online application also allows control of process parameters, capacity of volumetric pump, motor capacity and pressure within the load circuit.
Web interface of the adjustment model (LabVIEW / PXI) allows remote control of hydrostatic transmission, i.e. laboratory testing of transmission also by persons outside the testing laboratory.

CONCLUSIONS

The adjustment model of hydraulic transmission was tuned by successive iterations of simulations, performed in AMESim and data processing, performed in LabVIEW.

Validation of the adjustment model is confirmed by the shape of characteristic curves experimentally raised, by coupling of LabVIEW/ PXI virtual interface to a physical model, developed in the laboratory, equivalent to AMESim simulation model. One can notice comparable dynamics, theoretically and experimentally identified, of the adjustment system of hydraulic transmission.

Development of adjustment models for other configurations of hydraulic transmissions, of primary, secondary or mixed adjustment, can be regarded as a customization of this developed model.

The ability to use both mathematical models and real equipment allows the achievement in research laboratories of certain tests that would otherwise require *on-site* presence.

Co-simulation allows simultaneous use of several modeling languages within the same simulation network.

Use of this analysis method is extremely useful during development and design stages of complex automated systems, providing a quick solution to identify characteristics of the examined system.

Current state of research in the field of hydrostatic transmissions emphasizes need for use of advanced simulation models for their systemic analysis.

By using advanced modeling and experimental identification procedures high dynamic performance can be achieved for hydrostatic transmissions.

REFERENCES

Teodor Costinel POPESCU, Ph.D., graduated Institute of Civil Engineering – Faculty of Fluid Systems in 1978, and Polytechnic Institute of Bucharest – Faculty of Mechanichal Engineerig in 1989, Fluid Machines Faculty. Since 1983 he is working as senior scientific reserchер in fluid power systems field at Fluid Power Systems Research Institute – INOE 2000 – IHP Bucharest, Romania.

Dragos Daniel ION GUTA, Ph.D. graduated Mecanical Engineering Faculty from University Politehnica of Bucharest in 2002. He works as scientific researcher in fluid power systems at field at Fluid Power Systems Research Institute – INOE 2000 – IHP Bucharest, Romania; he is working also in the field of modeling and real time simulation of the fluid control systems.

Alexandru MARIN, Professor Ph.D. at University Politehnica of Bucharest, specialist in applied hydraulics, with contributions in experimental and theoretical research on mechatronic systems, automatic electro-hydraulic systems, non-Newtonian automotive hydraulic dampers, lubrication systems, flexible hydraulic tools at 700 bar, solid waste processing machines.

BIOGRAPHY

Popescu, T.C.; Dutu, I.; Vasiliu, C. and Mitroi, M. 2009. "Adjustement of conformity parameters of PID-type regulators using simulation by AMESim." *7th International Industrial Simulation Conference 2009, ISC 2009,* June 1-3, 2009, Loughborough, United Kingdom.

Drumea, P.; Popescu, T.C.; Blejan, M. and Rotaru, D. 2010. "Research Activities Regarding Secondary and Primary Adjustment in Fluid Power Systems." *7th International Fluid Power Conference Aachen „Efficiency through Fluid Power",* 22-24 March 2010, Aachen, Germany.

Popescu, T.C.; Drumea, A. and Dutu, I. 2008. "Numerical simulation and experimental identification of the laser controlled modular system purposefully created for equipping the terrace leveling installations", *International Spring Seminar on Electronics Technology (ISSE),* Budapest, Hungary; 7-11 May, 2008, in: " Proceedings - Reliability and Life-time Prediction", ISBN: 978-963-06-4915-5; pp.336-341.

LATE PAPER

A FUZZY Q-LEARNING APPROACH TO SIMULATE INTELLIGENT TRAFFIC CONTROL

Juan C. Pacheco[1,2], Rosaldo J. F. Rossetti[1], and Cesar H. Rodriguez[2]

[1] Department of Informatics Engineering (DEI), Artificial Intelligence and Computer Science Lab (LIACC)
Faculty of Engineering, University of Porto
Rua Dr Roberto Frias, S/N – 4200-465 Porto, Portugal

[2] Department of Electronic Automation and Design
La Salle University
Carrera 5 No.59 A-44, Sede Chapinero, Bogotá, Colombia

E-mail: jpacheco07@unisalle.edu.co, rossetti@fe.up.pt, cerodriguez@unisalle.edu.co

KEYWORDS
Agent-based modelling and simulation, microscopic traffic simulation, fuzzy control, Markov-Decision Process

ABSTRACT

This paper is particularly concerned with the representation of an agent behaviour suitable to model autonomous and efficient traffic control at urban network junctions. Network control efficiency is expected thus to emerge from these simpler autonomous entities cohabitating in a common environment. A fuzzy Q-learning approach is used to implement the agent-based reasoning and decision-making mechanisms and some preliminary simulation experiments are set up so as to assess the feasibility of our model to cope with urban traffic coordination.

INTRODUCTION

Transportation engineering encompasses methodologies and scientific principles in order to address classical transport issues. This scientific area deals with processes aimed at urban transport plan, system functional design, as well as operation and management of traffic networks (Roess et al. 2004). Moreover, traffic problems involve several variables that ought to be studied within a virtual controlled environment, which turns simulation methodologies into an important tool to describe transportation aspects such as geometry and topology design, traffic operations, control and management. The importance of simulation has grown far beyond, and became an imperative analysis technique to assess more intelligent, greener, and sustainable technologies developed under the umbrella of the Intelligent Transportation Systems (ITS) (Ferreira 2008). On the other hand, urban congestion has been a major problem for a long time already, challenging most traditional traffic control approaches to improve mobility in major large cities. This is quite a scenario where the implementation of autonomous, adaptable and intelligent controllers have been tested and applied. Besides mobility per se, it is important to bear in mind the fact that transportation systems are a major component of the world economy and have a big impact on the shape of every society, and traffic control becomes even more crucial in order to improve transportation systems now accounting for the economic perspective. Therefore, featuring smarter controllers with the adequate sensibility to all these issues is a key factor towards efficiently tackling mobility problems, in all levels.

Simulation projects are often intrinsically an interdisciplinary effort, which definitely leads to potential issues and calls for specific methodological considerations. In this context, traffic simulation evolves as a new generation platform encompassing different experts and perspectives, corroborating to one single common goal, namely to enhance control and monitoring techniques to improve mobility. If on the one hand stochastic simulation has played an important role in describing some complex aspects of traffic issues as they have the natural ability to describe probabilistic and statistical information about an environment, on the other hand microscopic simulations can discreetly or continuously predict the state of individual entities belonging to the virtual domain. Combining both is essential for us to describe vehicle behaviour responding to the dynamic changes of the environment, such as traffic control devices, pedestrian activities, vehicle dynamics and interactions, and so forth.

Among other methods, fuzzy logics and machine learning techniques, such as Q-learning, have been widely applied as an attempt at improving traffic control systems. Combining both consists of one type of integrated intelligent techniques so as to improve discrete and continuous processes, compensating disadvantages from each other. Q-learning alone consists of an on-line learning procedure based on Markov Decision Process (MDP), which can acquire knowledge about an environment from experiences even without the mathematical behaviour model. On its turn, fuzzy systems based on several of the so-called if-then rules, are capable of providing a linguistic interpretation of variables of an environment. Given such characteristics, the natural complimentarily of both has already been reported in several studies. One important concern in such a combination, however, is which model modifies the other as each method could have a cooperation module feeding back the other. For instance, previous works describe a Q-learning agent that compensate fuzzy controllers disadvantages in order to have a modified fuzzy system that can learn through experience by an exploration process that optimises the fuzzy rules related to the environment (Berenji 1994; Glorennec 1994; Kim et al. 2000; Deng and 2003). On other works, such as in (Li et

al. 2008), the fuzzy inference is introduce into the Q-learning procedure in order to facilitate generalisation in the state space. The proposed controller, as argued by the authors, "can learn to take the best action to regulate with the features of high throughput and low packet loss ratio." As we have mentioned, Q-learning describes a Markov Decision Process interpreting a finite environment of state-action pairs, so it has a discreet behaviour that might not be the better option to test continuous processes. In such situations, fuzzy systems, which have a continuous natural behaviour underlying their principal characteristic, can be integrated into the intelligent agent in order to tune Q-learning actions. This proposed model embedded into a feedback control diagram describes a novel approach of "fuzzy Q-learning" profiting from advantages of each algorithm to optimise continuous models.

In this paper, we describe a probabilistic model of real traffic scenario in order to devise a stochastic simulation based on a Markov chain process. We present a novel approach of the classical reinforcement Q-learning algorithm adopted on traffic simulations, which consists of a typical intelligent Q-learning agent that is embedded into a feedback control diagram with a fuzzy controller. The fuzzy inference acts as a linguistic interpreter of the network error in order to tune the Q-learning agent actions, improving an optimal policy applied to the traffic lights. Additionally, we test with this algorithm within a microscopic traffic simulation environment so as to analyse and assess both continuous and discreet performance measures. Simulation scenarios were set up in a way that we could observe the proposed approach applied to models representing specific parts of real traffic environments.

STOCHASTIC SIMULATION OF TRAFFIC CONTROL

In major urban scenarios, infrastructure works near to saturation. This issue implies several consequences on economic, social and environmental sectors. Simulating traffic control techniques is imperative then so as to improve transportation systems, since it helps establishing a reasonable model to improve control techniques and management policies. Stochastic simulation based on real data of traffic behaviour defines the probabilistic functions in order to describe an average model. Moreover, intelligent traffic controllers featuring simulation-based learning may potentially improve its performance.

The traffic domain might be considered as a system whose evolution is described by stochastic processes, so the Markov chain metaphor seems to fit such a finite environment. Stochastic models consist of algorithms that evolve from discreet values based on random variables (Haykin 1998). Markov chains have a state space of all possible values that random variables can assume. This structure of stochastic processes, namely $\{X_n, n=1,2, ...\}$ is associated with a conditional probability distribution that depends only on the value X_n in order to establish the next state X_{n+1}, independently of all previous states. The "Markovian" nature of such systems is then described as "a sequence of random variables $X_1, X_2, ..., X_n, X_{n+1}$ that form a Markov chain if the probability for the system to be in state X_{n+1} at time $n+1$ depends exclusively on the probability for the system to be in state X_n at the time n" as stated in (Haykin 1998). In order to describe a traffic behaviour based on the Markov chain metaphor, we establish a simple traffic intersection as a common way to describe an evaluative traffic model based on real statistical data. Such information was borrowed from a work where authors took information from the local mobility secretariat in Bogota, Colombia (Bohórquez and Melo 2005). In this way, in order to design an intelligent controller, we interpret information that consists of a Passenger Car Unity (PCU) value per each 15 minutes on all roads that compose an intersection. As shown in Figure 1, the structure of the stochastic simulation was applied to a high flow car intersection with a simple composition.

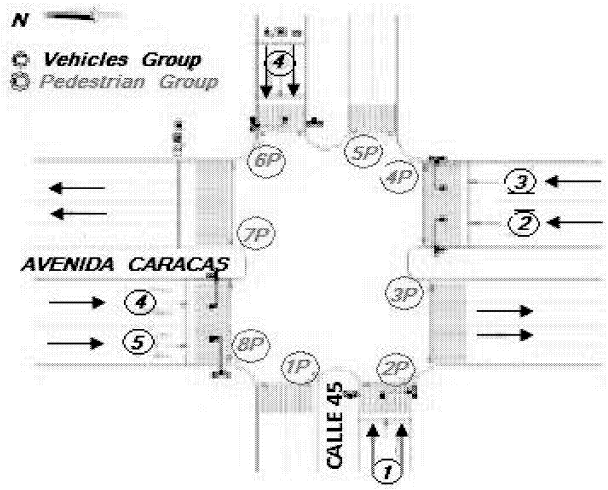


Figure 1: Intersection Layout

We described the whole behaviour of the traffic network based on a set of finite states (Equation 1) based on the occupation percentage per each road (given by Equation 2). As shown in Equation 3, we applied the basic concept of probability, namely the frequency of some event in a random environment to be verified (Devore 1999). Then, we normalised such a concept in order to get transition probabilities between traffic states and interpret the problem as a Markov chain.

$$S = \begin{cases} if\ OP \geq 0\ and\ OP < 25 \Rightarrow S = 1 \\ if\ OP \geq 25\ and\ OP < 50 \Rightarrow S = 2 \\ if\ OP \geq 50\ and\ OP < 75 \Rightarrow S = 3 \\ else \Rightarrow S = 4 \end{cases} \quad (1)$$

$$OP\% = \frac{PCU_{(t)}}{MaxPCU} \quad (2)$$

$$P = \frac{Event_{(S)}}{NumberExperiments} \quad (3)$$

In Equation 1 above, OP is the occupation percentage and S is a state. The probabilistic description based on statistics, interprets traffic behaviour through a day that has several intervals involving different traffic performance depending on typical transportation variables such as the time, collisions

and road maintained. As shown in Figure 2, the Markov chain description generated a state value of the traffic behaviour per each period, which has a random module in parallel in order to adapt real data into a stochastic simulation, meaning that the results give an occupation percentage per road depending on the probabilistic assignment.

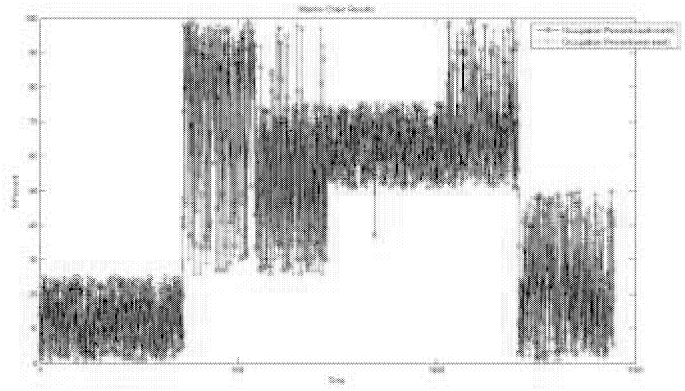


Figure 2: Markov Chain Process Results

AN APPROACH TO TEST FUZZY Q-LEARNING TRAFFIC CONTROL SYSTEMS

An Intelligent Learning Agent

In order to test the proposed approach, we have implemented a Q-learning based agent as the underlying structure of our traffic controllers. Q-learning is an incremental dynamic programming procedure that determines the optimal policy solving Markovian decision problems without a behaviour model and explicit knowledge about the system. It consists of an on-line learning procedure getting experience about an environment on the basis of samples of the form as shown in Equation 4.

$$S_n = (i_n, a_n, j_n, g_n) \quad (4)$$

In the expression above, n denotes discrete time and each sample S_n represents a four-tuple variable, which describes a trial action on a state i_n that results on a transition to another state $j_n=i_{n+1}$ at a cost $g_n=g(i_n, a_n, j_n)$ (Watkins 1989). The learning algorithm implemented by the agent can be summarized as follows:

1. Set parameter gamma ($0<\gamma<1$) and environment reward matrix R.

We applied this reinforcement learning into the traffic systems in order to improve its behaviour, and we defined a state space based on occupation percentage of the road that composes an intersection. Thus, in order to describe a reward matrix we basically take state 1 as an optimal goal that the system could interpret in order to describe the policy (3).

2. Initialize Q as a null matrix, which consists in the brain of the agent interpreting a state-action (s, a) environment.

In traffic systems, this Q matrix describes the state space involving actions based on different plan times, and applies it into the traffic lights.

3. Thereafter, stochastic simulation plays an important role in order to improve the learning processes. The last parameters were established to describe an exploration procedure based on the stochastic simulation, encompassing the following:
3.1 Validate a traffic state based on the occupation percentage.
3.2 Select one among all possible actions for the current state.
3.3 Get maximum Q value of this current state based on all possible actions.
3.4 Compute Equation 5, below:

$$Q(s,a) = R(s,a) + \gamma \times \max[Q(next_S, all_A)] \quad (5)$$

3.5 Set the next state as the current state.

After the learning process, the agent improves its behaviour based on the information resulting from the stochastic simulation. During this process, the Q-values update the information in order to adapt to environment based on the Q-learning algorithm. In the end, the values are expected to converge very likely resulting in optimal control policies, whereas the learning process per each state could be different.

An Adapted Fuzzy Controller for Actions Calibration

Many physical systems or processes involve an uncertain or complex description of a domain and it could be difficult to acquire an analytical model to find an optimal solution for it. Therefore, an expert knowledge of the environment based on a human cognitive process can describe the problem even without any mathematical model. Fuzzy set theory is “a body of concepts techniques that give a form of mathematical precision to human that are in many ways imprecise and ambiguous by standard classical mathematics” (Shin and Xu 2009). Also, in a fuzzy system knowledge is represented by if-then rules that involve the concept of fuzzy inference called approximate reasoning that interprets linguistic representation of knowledge. In Equation (6) we show the basic form of the rule.

if S1 ***is*** E1 ***and*** S2; S2... ***then*** y1 ***is*** B1... (6)

Our fuzzy controller then was adapted into the traffic environment and included into the feedback control loop. The inference module interprets an error and a derivate error of the system. The fuzzy knowledge base was established in these two linguistic variables, in order to generate a delta time to tune actions expressed on plan time. Table 1 establishes the fuzzy rules that have the linguistic representation of the traffic feedback control model, where each abbreviation represents a MF (Member Function) per universe in the fuzzy inference. We described a common way for all the assignments to be into the fuzzy controller, namely PG (Big Positive), PP (Small Positive), M (Medium), NG (Big Negative), NP (Small Negative). Then, the inputs and

outputs of the system have the same universe, but with their own behaviour description.

Table 1: Fuzzy Knowledge Base

	NG	NP	M	PP	PG
NG	NG	NP	NP	M	M
NP	NG	NP	NP	NP	NG
M	NG	NP	M	PP	PG
PP	PG	PP	PP	M	M
PG	PG	PG	PP	PP	PG

Once the fuzzy inference was defined, the next step was modelling a surface control, which describes the output value of the fuzzy controller through the deffuzzificator module. This module obtained a value based on the fuzzy inference by the general deffuzification method known as Centre of Area (COA), as given in Equation (7).

$$\Delta_t = \frac{\sum_{i=1}^{N} x_i \times \mu_A(x_i)}{\sum_{i=1}^{N} \mu_A(x_i)} \qquad (7)$$

In Equation 7 above, N is the number of the discrete elements in the universe of discourse, x_i is the value of discrete element, and $\mu_A(x_i)$ represents the corresponding MF (Member Function) value at point x_i. The surface control interprets the fuzzy inference in order to get an optimal response in time that can be used into the feedback control loop.

Coupling Q-Learning Agents and Fuzzy Controllers

In order to integrate the Q-learning agent and the Fuzzy controller, we have proposed a feedback control diagram, which has a model based on an ideal behaviour of the traffic network. As we show in Figure 3, the ideal behaviour was established by an average of the stochastic simulation which describes an optimal performance based on the probabilistic response.

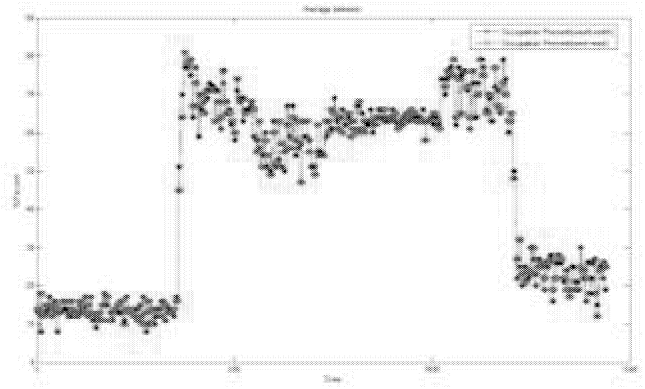


Figure 3: Average Traffic Behaviour Model

Such an ideal model is the main data comparator to improve actions into the traffic environment. On the other hand, the proposed fuzzy controller acts as the principal control module in order to tune actions that the intelligent agent takes into the finite environment. Moreover, fuzzy inference describes how far the system is from an optimal behaviour, interpreting an error and derivate error. The fuzzy rules were based on generating a delta of time in order to tune actions and adapt an optimal time plan. The resulting dynamical programming controller is shown in Figure 4 and, as we have mentioned, this one has two principal control modules interacting within a classical feedback control loop.

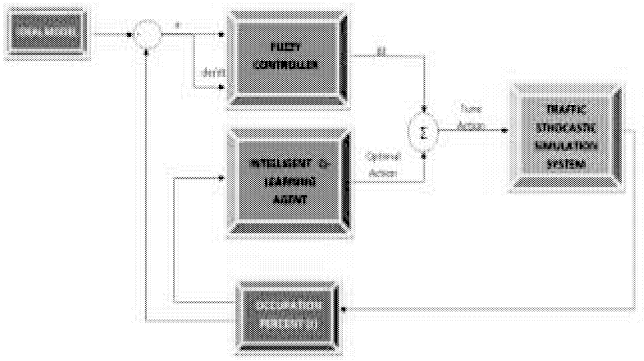


Figure 4: Control Feedback Loop

EXPERIMENTAL SET-UP

The simulation interaction plays an important role in the testing procedure of the intelligent controllers. The intelligent controllers interacting in the control loop are modelled and implemented into the microscopic simulation platform MAS-Ter Lab (Ferreira 2008). The learning procedure was based on the Markov chain stochastic simulation, whereas the intelligent agent acts together with different simulation scenarios in order to improve and test the adaptation of traffic controllers. As depicted in Figure 5 the simulation structure is composed by three basic modules, name the stochastic simulation (Markov chain), the fuzzy Q-learning agent model, and the microscopic simulator.

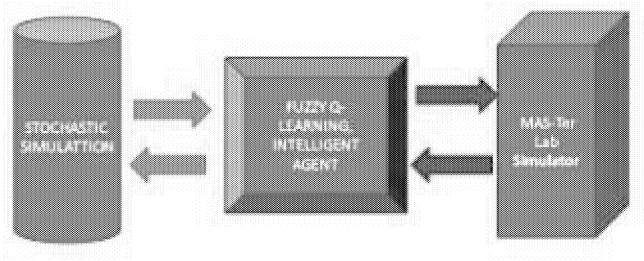


Figure 5: Simulation and Control Architecture

In order to establish critical traffic behaviour, we modelled a simple network into the microscopic platform, where the intelligent agent is acting in suboptimal manner. We upload some particular behaviour based on the simulator parameters and, to have different reactions of overall performance measures, we manipulated the traffic flow parameter, which are vehicles per hour value (vph) per each road at the intersection. Similar approaches have been proposed, such as in (Tan et al. 1996). We described two critical scenarios of high flow cars approaching the intersection. One can be interpreted as real traffic conditions as possible so as to represent the highest peak period (Figure 6). The other scenario consists of a combination of high and medium car flow approaching the intersection (Figure 7). Both scenarios

describe real behaviours in a microscopic simulation, as commonly found currently in major populated urban areas.

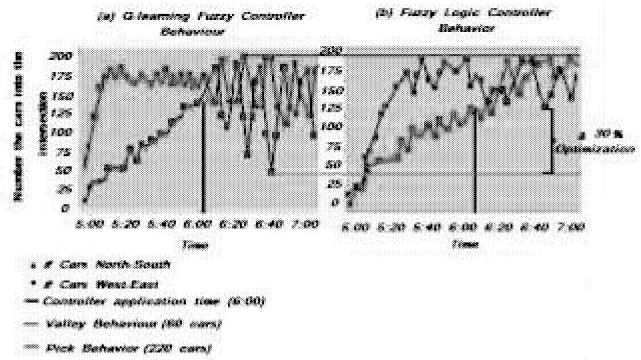


Figure 6: Highest Peak Period Scenario

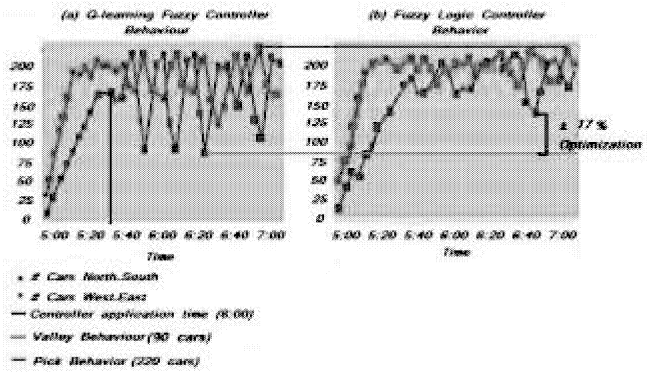


Figure 7: Combined High-Medium Behaviours Scenario

Both scenarios were tested and we can infer that traffic needs to have previous knowledge about the environment in order to improve overall system performance, while controllers can improve their ability to decrease cars delay throughout the network. Continuous output is an optimal way to test green extension time for the controller cycle, thus giving a wider range within which the systems can act. Both scenarios demonstrated to react positively to the approach proposed in this work. Nonetheless, further extensions are still need to achieve an efficient control system where optimum will also account for other performance measures rather than average delay. Area optimisation, based on cooperation among a number of agents is also a longer term goal to pursue.

CONCLUSIONS

Throughout this work, we adapted traffic controllers as an important way to integrate different types of simulation in order to improve new-generation control techniques. We adopted a stochastic simulation approach that acts into a finite environment in order to get an average behaviour model of traffic networks. Also, microscopic simulation was used as a monitoring interface of the intelligent controller in typical traffic conditions so as we could observe our approach's outcome. We have also relied on the combination of different Artificial Intelligence (AI) techniques as a means to describe different complex systems, in which way different approaches will have their own advantages and profit from each other. In this paper, we particularly study reinforcement learning as an agent exploring and getting experience from a simulation traffic environment in order to describe optimal actions. The fuzzy controller acts as a tuning module based on a fuzzy inference and is integrated into a feedback control diagram. The proposed approach suggests that control optimisation based on the combination intelligent techniques is a finest way to describe complex and stochastic systems such the traffic domain. Nonetheless, some issues are still open and need further discussion. For instance, integrating pedestrian-vehicle interactions with the control loop and analysing the effects of information systems on the decision processes are problems remaining to be tackled.

ACKNOWLEDGEMENT

The first author has been partially supported by the IAESTE Student Exchange Programme while spending a technical internship in the Department of Informatics Engineering, Faculty of Engineering, at University of Porto, Porugal.

REFERENCES

Berenji, H.R. 1994. "Fuzzy Q-Learning: A New Approach for Fuzzy Dynamic Programming." In *IEEE World Congress on Computational Intelligence Proceedings of the Third IEEE Conference on Fuzzy Systems*. 486-491.

Bohórquez, O. and A. Melo. 2005. *Semáforo Autónomo Inteligente*. Facultad de Ingeniería de Diseño y Automatización Electrónica, Universidad de La Salle, Bogotá, Colombia.

Deng, C. and M.J. Er. 2003. "Efficient implementation of dynamic fuzzy Q-learning." In *Proceedings of the Fourth International Conference on Information, Communications and Signal Processing.* Vol.3, 1854-1858.

Devore, J.L. 1999. *Probability and Statistics for Engineering and the Sciences*. Brooks/Cole Publishing Co., Pacific Grove, CA.

Ferreira, P.A.F. 2008. *Specification and Implementation of an Artificial Transport System*. MSc Dissertation. Porto: FEUP.

Glorennec, P.-Y. 1994. "Fuzzy Q-learning and dynamical fuzzy Q-learning." In Proceedings of the third IEEE International Conference on Fuzzy Systems. IEEE Computer Press, Piscataway, NJ. 474-479.

Haykin, S. 1998. *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice-Hall, Englewood Cliffs, N.J.

Kim, M.-S., S.-G., Hong, and J.-J. Lee. 2000. "Self-Learning Fuzzy Logic Controller using Q-Learning." *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.4, No.5, 349-354.

Li, X., X. Zhao, Y. Jing, and N. Zhang. 2008. "Fuzzy Q-learning flow control for high-speed networks." In *Control and Decision Conference* (Yantai, Shandong, July 2-4). 383-387.

Roess, R.P., W.R. Mcshane, and E.S. Prassas. 2004. *Traffic Engineering*. Prentice-Hall, Englewood Cliffs, N.J.

Shin, Y.C. and C. Xu. 2009. *Intelligent Systems: Modeling, Optimization, and Control*. CRC Press, Boca Raton, FL.

Tan, K.K., M. Khalid, and R. Yusof. 1996. Intelligent traffic lights control by fuzzy logic. *Malaysian Journal of Computer Science*, Vol.9, No.2 (Dec), 29-35.

Watkins, C.J.C.H. 1989. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England.

AUTHOR LISTING

AUTHOR LISTING

Abelha A. **67**
Andreescu C. **154**
Angeli C. **223**
Ardalan A. **75**

Bataus M. **154**
Benko B.K. **10**
Bioly S. **195**
Braekers K. **172**
Byrne P.J. **62**

Cabanellas J.M. **147**
Caramihai S. **161**
Caris A. **172**
Chapon D. **17**
Chatzinikolaou A. **223**
Conley W. **45/50**
Croitorescu V. **154**
Culita J. **27**

de Alegría Uriarte M. **117**
Delebecque F. **17**
Díaz de Arcaya A. **117**
Diaz R. **75**

Expósito-Sánchez M.A. . **131**

Fazekas P. **203**

Gamón-Giménez M.A. .. **131**
García I. **117**
García-Fernández I. **131/139**
Geraghty J. **62**
Gonçalves P. **67**
Govaert T. **102**
Guta D.D.I. **248**

Hazi G. **239**
Heavey C. **62**

Janosy J.S. **239**
Janssens G.K. **172/184**
Je T.-J. **122**
Ježek D. **57**
Jimena G.J. **147**
Juchova V. **57**

Kennedy B. **62**
Kereszturi A. **239**
Kim S.-J. **122**
Kirchhof P. **22**
Klumpp M. **195**
Komasilovs V. **39**
Kožusznik J. **80**
Kuchař Š. **80**

Lázaro G. **117**
Le D. **122**
Lee J.-M. **122**
Legeny M. **10**
Lencse G. **214**
Liu G. **167**
Lohrengel A. **5**
Lokshina I. **179/208**

Machado J. **67**
Maciac A. **154**
Marin A. **248**
Marques A. **67**
Martínez-Durá R.J. **131/139**
Mebrahtu H. **95**
Meitalovs J. **245**
Meseth N.G. **22**
Miranda M. **67**
Munteanu C. **161**

Neves J. **67**
Nikoukhah R. **17**

Oprean M. **154**

Pacheco J.C. **257**
Pales J. **239**
Peerlinck K. **102**
Pisuchpen R. **184**
Pla-Castells M. **131/139**
Popescu T.C. **248**

Radev D. **208**
Rodriguez C.H. **257**
Rossetti R.J.F. **257**
Roßmann J. **109**
Rubina T. **228**
Rupp K. **34**

Saur A. **195**
Schäfer G. **5**
Schiele E. **179**
Selberherr S. **34**
Shen L. **5**

AUTHOR LISTING

Stalidzans E. **39/228/245**
Stefanoiu D. **27**
Stern O. **109**
Stolfa S. **57**
Syberfeldt A. **89**

Tovar C. **147**

Udrea A. **161**
Usatorre L. **117**

Van Landeghem H. **102**
Van Volsem S. **190**
Varga A. **214**
Vegh E. **239**
Verdiesen J. **190**
Voinescu M. **161**
Vondrak I. **57**

Walker R. **95**
Weinbub J. **34**
Wischnewski R. **109**

Young P. **62**

Zacepins A. **245**
Zelewski S. **195**
Zheng P. **167**