10th INTERNATIONAL INDUSTRIAL SIMULATION CONFERENCE 2012

ISC'2012

EDITED BY

Vladimír Janoušek

and

Šárka Květoňová

JUNE 4-6, 2012

BRNO, CZECH REPUBLIC

A Publication of EUROSIS-ETI

10th International Industrial Simulation Conference 2012

BRNO, CZECH REPUBLIC

JUNE 4-6, 2012

Organised by

ETI- The European Technology Institute

Sponsored by

EUROSIS - The European Simulation Society

In Cooperation
With

Brno University of Technology

Co-Sponsored by

EK	BITE
Delft University of Technology	Ghent University
Higher Technological Institute	LMS
University of Skovde	UPV

Hosted by

Brno University of Technology
Faculty of Information Technology
Brno, Czech Republic

INTERNATIONAL PROGRAMME COMMITTEE

Michel Gourgand, Universite Blaise Pascal, Clermont-Ferrand, France
Imed Kacem, Université Paul Verlaine Metz, Metz, France
Habtom Mebrahtu, APU, Chelmsford, United Kingdom
Carlo Meloni, Politecnico di Bari - DEE, Bari, Italy
Pascal Meyer, Forschungszentrum Karlsruhe, Karlsruhe, Germany
Markus Vorderwinkler, PROFACTOR GmbH, Steyr, Austria
Roland Wischnewski, (RIF) e.V., Dortmund, Germany
Farouk Yalaoui, Universite de Technologie de Troyes, Troyes Cedex, France

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

Simulation in Automotive Systems
Track Chair: Steffen Strassburger, Technical University of Ilmenau, Ilmenau, Germany
Naoufel Cheikhrouhou, EPFL, Lausanne, Switzerland .
Julien Richert, Daimler AG, GR/PAA, Sindelfingen, Böblingen, Germany
Rob Walker, Anglia Ruskin University, Chelmsford, United Kingdom

Simulation in Robotics
Leopoldo Arnesto Angel, UPV, Valencia, Spain
A. Chatzinikolaou, Athens, Greece
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
Teresa Alvarez, University of Valladolid, Valladolid Spain
Christos Bouras, University of Patras, Patras, Greece
Tom Dhaene, Ghent University, INTEC, Ghent, Belgium .
Adrian Florea, University "Lucian Blaga" of Sibiu, Sibiu, Romania
Domenico Giunta, ESA-ESTEC, Noordwijk, The Netherlands
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
Ernst Kesseler, NLR, Amsterdam, The Netherlands
Wolfgang Kreutzer, University of Canterbury, Christchurch, New Zealand

Simulation in Logistics, Transport and Harbour Simulation
El-Houssaine Aghezzaf, Ghent University, Ghent, Belgium
Isabel Garcia Guttierez, Univ. Carlos III de Madrid, Madrid, Spain
Martin Grunow, TU Munich, Munich, Germany
Thomas Hanne, Univ. of Applied Sciences Northwestern Switzerland, Olten, Switzerland
Marie-Ange Manier, UTBM, Belfort, France
Herve Manier, UTBM, Belfort, France
Roberto Montemanni, IDSIA, Manno-Lugano, Switzerland
Jaap Ottjes, TU Delft, Delft, The Netherlands
Roberto Razzoli, University of Genova, Genova, Italy
Rosaldo Rossetti, University of Porto, Porto, Portugal
Rik van Landeghem, Ghent University, Ghent, Belgium

INTERNATIONAL PROGRAMME COMMITTEE

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
Jose Machado, University of Minho, Braga, Portugal
Peter Summons, University of Newcastle, Australia

Complex Systems Modelling
Track Chair: Igor N Litvine, Nelson Mandela Metropolitan University, Port Elizabeth, South Africa
Frantisek Capkovic, Slovak Academy of Sciences, Bratislava, Slovak Republic
Christophe Claramunt, Naval Academy Research Institute, Brest, France
Gabriella Dellino, University of Siena, Siena, Italy
Werner Dubitzky, University of Ulster, Coleraine, United Kingdom .
Krzysztof Fujarewicz, Silesian Technical University, Gliwice, Poland
Laszlo Gulyas, AITIA International Inc., Budapest, Hungary
Eugene Kindler, Ostrava University, Ostrava, Czech Republic
Alexandre Nketsa, LAAS, Toulouse, France
Michal Pechoucek, Czech Technical University, Prague, Czech Republic
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

Simulation in Industrial Design and Product Design
Chiara Catalano, IMATI-CNR, Genoa, Italy
Yan Luo, NIST, Gaithersburg, USA
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
Graham Saunders, Loughborough University, Loughborough, United Kingdom

Simulation in Energy and Power Systems
Janos-Sebestyen Janosy, KFKI Atomic Energy Research Institute, Budapest, Hungary

Simulation in Multibody Systems
Track Chair: Bernd Schaefer, DLR, Wessling, Germany
Ignacio Garcia-Fernandez, University of Valencia, Valencia, Spain
Jose Diaz Lopez, Anoto AB, Lund, Sweden

Simulation in Chemical, Petroleum and Mining Engineering
Diganta Das, 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

INTERNATIONAL PROGRAMME COMMITTEE

Carlos Palau, UPV, Valencia, Spain
Matthias Reuter, CUTEC Gmbh, TU-Clausthal, Clausthal, Germany

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
Marcos A.Rodrigues, Sheffield Hallam University, Sheffield, United Kingdom

Verification, Validation and Accreditation
Roberto Revetria, University of Genoa, Genoa, Italy

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

Workshops

Workshop on Intelligent Transport Systems
Track Chair: Anna Syberfeldt, University of Skovde, Skovde, Sweden
Paul Davidsson, Blekinge Institute of Technology, Ronneby, Sweden
Petr Hanacek, Brno University of Technology, Brno, Czech Republic
Jairo Montoya Torres, Universidad de la Sabana, Chia, Columbia

NANOSIM
Clemens Heitzinger, University of Vienna, Vienna, Austria
Yong K. Kim, University of Massachusetts Dartmouth, Dartmouth, USA
Javier Marin, ETSI, University of Malaga, Malaga, Spain .

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

INDUSTRIAL SIMULATION CONFERENCE 2012

All articles published in these Proceedings have been peer reviewed

EUROSIS-ETI Publications are ISI-Thomson and IET referenced
Legal Repository: Koninklijke Bibliotheek van België, Keizerslaan 4, 1000 Brussels, Belgium
CIP 12.620 D/2011/12.620/1

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, Ghent University Ostend Campus, 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: 978-90-77381-71-7
EAN: **978-90-77381-71-7**

PREFACE

Dear Participants,

It is our pleasure to welcome you to the 10th Annual Industrial Simulation Conference ISC'2012, organized by EUROSIS.

The conference is being held at Faculty of Information Technology (FIT), one of the eight faculties of Brno University of Technology (BUT). FIT was established on January 1st, 2002. It was created out of the amalgamation of the former Department of Computer Science and Engineering and the Faculty of Electrical Engineering and Computer Science, Brno University of Technology. The Department of Computer Science and Engineering was established in 1964 and has had a long tradition in teaching computer science and informatics in the framework of the former Faculty of Electrical Engineering, later called Faculty of Electrical Engineering and Computer Science.

Simulation has always been one of the main challenges which motivated the development of computers and research in computer science. Recall the Manhattan project in World War II, for example. Simulation is one of the key points where computer science meets other sciences and the connection is very fruitful and motivating.

Computer simulation in its many forms and applications is the key theme of this conference. Special emphasis is put on applications in industry, where the effect of simulation-based techniques has undoubted economic and economical impact. Nevertheless, the contributions are not strictly limited to the industrial applications. Topics include simulation-based analysis, optimization and decision support in many areas such as manufacturing, robotics, electronics, telecommunication, transportation, logistics, software processes, chemistry, etc. We believe that each of the contributions does its bit for advancing computer simulation.

We would like to thank the invited speakers and all the contributors. Finally, we would like to thank Philippe Geril, the main EUROSIS organizer and coordinator. Without his effort this event would be hardly possible.

We hope you will enjoy your stay in Brno and we believe that the conference will be the opportunity for interesting scientific discussions.

Vladimír Janoušek and Šárka Květoňová
ISC’2012 General Conference and Program Chairs

CONTENTS

PROJECT MANAGEMENT SIMULATION

SOFTWARE DESIGN SIMULATION

3D SIMULATION

CONTENTS

CONTENTS

LATE PAPERS

SCIENTIFIC PROGRAMME

PROJECT MANAGEMENT SIMULATION

MODELLING OF PROJECT PORTFOLIO IN RENEW BASED ON OBJECT ORIENTED PETRI NETS (OOPN)

Michal Němec
Šárka Květoňová
Faculty of Information Technology
Brno University of Technology
IT4Innovations Centre of Excellence
Bozetechova 2
612 66 Brno, Czech Republic
E-mail: xnemec19@stud.fit.vutbr.cz, kvetona@fit.vutbr.cz

KEYWORDS

Project Portfolio Management, Object Oriented Petri Nets (OOPN), Renew, Modelling, Simulation, Optimization.

ABSTRACT

The paper is focused on modelling of project portfolio in Renew tool based on Object Oriented Petri Nets (OOPN). It presents a possible way or concept for project portfolio control and optimization with respect to specific characteristic of individual projects. The main emphasis is put on the resources modelling, simulation, and optimization during the project life time. The 4 main parts of our model are described: portfolio, project, human resources and connection to other tools for project planning or scheduling. Finally, there is described a possible way

INTRODUCTION

The field of project portfolio management is broad, challenging and rewarding. An important part of project portfolio management is the resources management domain on which this paper is focused. This domain is fast the most crucial part of project management because it affects success and/or failure of the whole project (project portfolio). We keep in mind not only human resources, but material, financial etc., too. The paper proceeds from (Janoušek, Květoňová 2009) and it shows a possible way how to implement the described concept in Renew. The main motivation of our underlying research is to simplify the whole process of project portfolio management by means of appropriate models, techniques, and tools.

The need of efficient project portfolio control is required together with increasing complexity of project resource consumption. There are a lot of tools, techniques, methods etc. to model projects and their control but without respect to specific characteristics and attributes of individual projects and resources (especially human resources). One of the possibilities is a combined approach presented in (Janoušek, Květoňová 2009) and Renew tool usage both to project portfolio modelling, analysis, control, simulation and optimization. Renew provides a flexible modelling approach/environment and enables us to model and control the whole process of project portfolio management.

TECHNICAL BACKGROUND

First of all, basic terms and relations of project management and planning/scheduling domains are given. Nevertheless, it is expected that a reader is familiar with the concept. Deeper explanation of the terms can be found in (Gray and Larson 2005, Kerzner 2005, Morris and Pinto 2007, Kendall G. and Rollins S. 2003, Girault 2003, Meredith J. R. 2012 etc.).

A project is a temporary group activity designed to produce a unique product, service or result. A project is temporary in that it has a defined beginning and end in time, and therefore defined scope and resources. A project is unique in that it is not a routine operation, but a specific set of operations designed to accomplish a singular goal. See (P.M.I. 2012).

Project management is a procedure of managing and directing time, material, personnel and costs to complete a particular project in an orderly and economical manner; and to meet established objectives in time, costs, and technical results. Project management is the application of knowledge, skills, tools, and techniques to execute projects effectively and efficiently. It’s a strategic competency for organizations, enabling them to tie project results to business goals — and thus, better compete in their markets. See (Schwalbe 2005, P.M.I. 2012).

A process is a series of actions bringing about a result. It is a complex of mutually connected resources and activities, which changes inputs to outputs. At present, activities and resources under the project are managed almost entirely like processes. A resource processes the individual operations.

Program is a group of related projects managed together to obtain specific benefits and controls that would likely not occur if these projects were managed individually. While project management focuses on delivering the specific objectives of the project, program management is focused on achieving the strategic objectives and benefits of the integrated program (Robertson, 2012).

Project Portfolio is an open group of mutually coordinated and conditioned projects (new projects start and finish). Projects/programs in portfolio use shared resources with defined timetables. A simplified model of project portfolio is shown in Fig. 1.

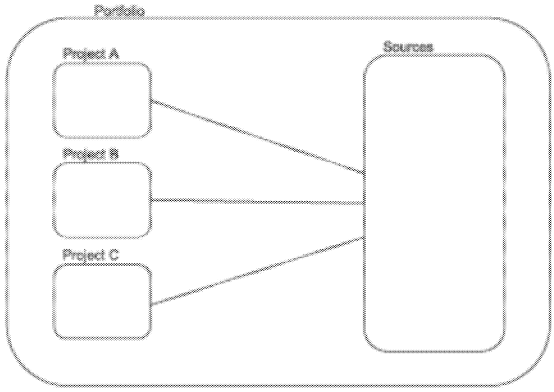


Fig. 1: Simplified model of project portfolio

Project Portfolio Management is the centralized management of one or more portfolios, which includes identifying, prioritizing, authorizing, managing, and controlling projects, programs, and other related work, to achieve specific strategic business objectives. It is about more than running multiple projects. Each portfolio of projects needs to be assessed in terms of its business value and adherence to strategy (P.M.I. 2012).

PROPOSAL OF THE MODEL

Every new project is based on some initial ideas or impulses. Then the ideas are discussed by relevant managers, who make decisions about project acceptance or rejection for future realization.

An example of the decision making process about new projects is shown in Fig. 2.

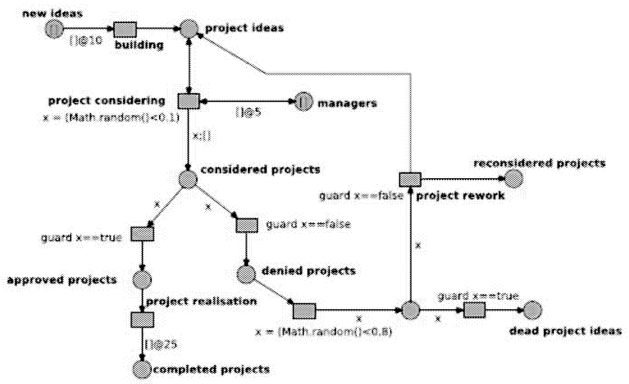


Fig. 2: Decision Making Process

Initial impulses are presented by the fact they arise every 10 time units (e.g. days). This process takes 5 days. Finally, only one project in ten is accepted. After that, concrete projects are realized (it takes on average 25 days).

In fact, there are many other factors and attributes that affect a project and we need to analyse them in more details. That enables us to create a correct model of the whole project portfolio.

Modelling of project portfolio

Project portfolio means a set of mutually related projects (in an organization) which use shared resources in order to comply with project goals.

First of all, an accurate and object oriented proposal of project portfolio is created. In Renew, all modelled entities by an implicit constructor or destructor are described, see below:

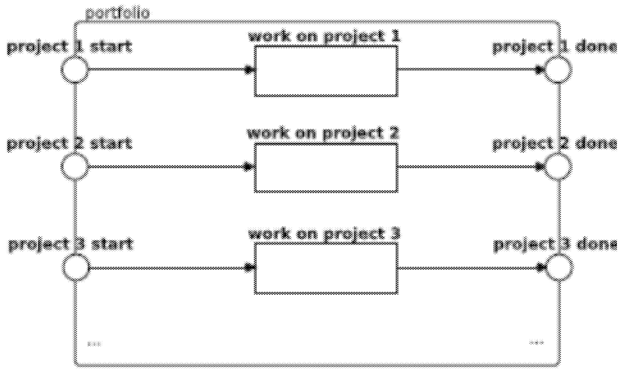


Fig. 3: Model of project portfolio

Fig. 3 describes an abstract view on project portfolio. Simply, every project has one initialization and one final state (in context of each project portfolio) and individual tasks are performed among them. These tasks require resources across the organization which assure their realization.

Portfolio includes many different projects. They consist of many individual activities sharing the same resource pool. Thus, There is a necessity to choose the most suitable (the best) resources for these activities. An abstract view on project portfolio is shown in Fig. 4 (Class Diagram).

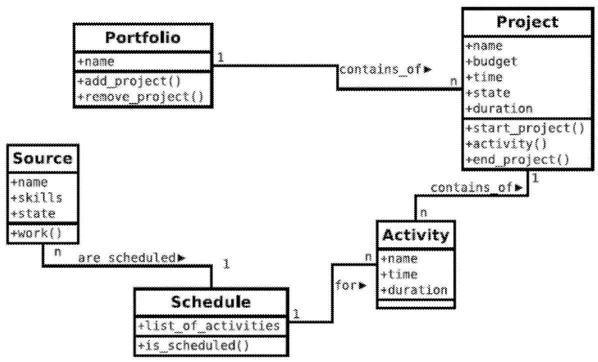


Fig. 4: Class Diagram

Main classes are: portfolio, project, activity and resource (including its schedule). Project portfolio is specified by its unique name and it uses two basic methods: add or remove project into / from project portfolio.

Project class is described by the following attributes: *name, budget, duration* and *status* (started, finished etc.). Methods included in this class enable to start a project, to add an activity and finish this project. An activity is described by two attributes: time and duration.

Main attributes of a resource are: name, abilities and state (available, working etc.). If a resource is required we use method *(work())* with parameters: start of an activity and its duration. After that we need to verify if an activity of a resource does not overlap another one (in class schedule).

Fig. 5 shows an implemented proposal of a project portfolio reference net in Renew.

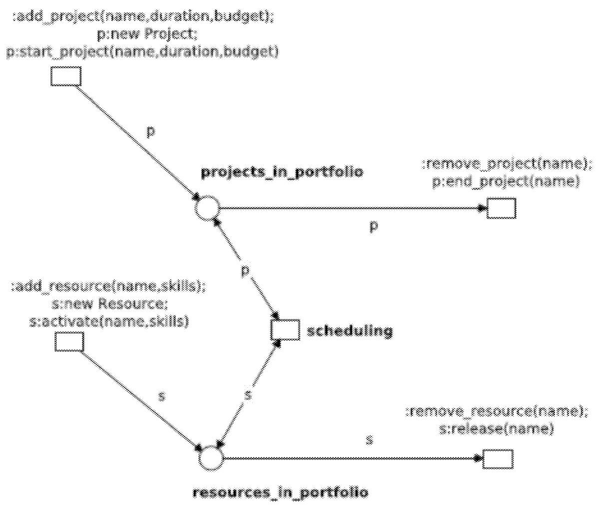


Fig. 5: Reference net of project portfolio (portfolio.rnw)

The net enable add/remove a project. After a new project *(:add project(name,duration,budget))* is added, a new object from project reference net is created. An activity is started with predefined parameters (name of project, duration, budget). A class includes one state where all active projects are saved (a list of relevant attributes).

In case of removing a project from a project portfolio we use method *((:remove project(name))*. There is chosen a project which is identified by the same name. Second part of the model is focused on resources which can be added into a project portfolio by method :*add resource(name,skills)* and removed by method *:remove resource(name)*. Transition *scheduling* executes resources scheduling on individual activities.

Modelling of projects

Projects consist of individual activities which can run parallel or sequential (an activity waits for another one). Petri net model of a project is presented in Fig. 6 (project is represented as a dynamically called class).

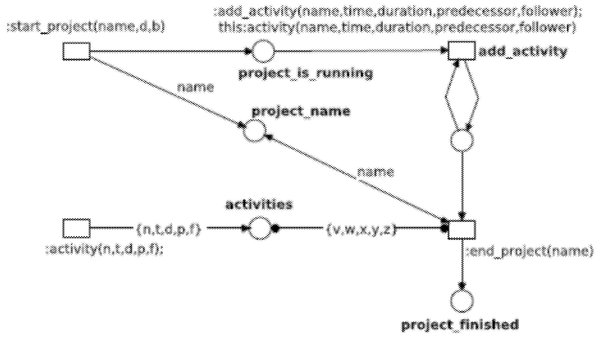


Fig. 6: Reference net of dynamic project - project.rnw

We use uplink *:start project(name,duration,budget)* for project activation. After a project is activated, we can add individual activities by method call *:add activity(name,time,duration,predecessor,follower)*. *Predecessor* represents a previous activity and *follower* represents a following activity. After that, individual activities are saved in place „activities“ (represented as a list of individual parameters).
This model is a quite complicated, thus, we use separated reference nets representing individual projects.

Modelling of resources

Every activity requires human resources with different characteristics. We need to choose the most suitable resources for those activities, thus, we decide according to their attributes: work experiences, required abilities, availability, wage, skills (values from interval 0-1) etc. Lower values mean resources have better preconditions to attend the project (resources with more experiences, higher level of needed skills which enable to manage an activity faster and in a higher level of quality).

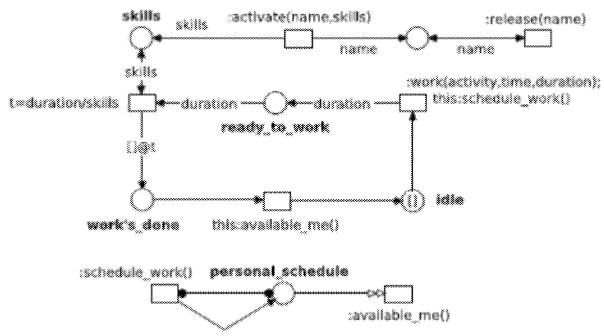


Fig. 7: Reference net of human resource (resource.rnw)

Reference net describing a resource provides methods *:activate(name,skills)* and *:work(t,d).* By this way, concrete resource is called by an activity. Parameters *activity* and *duration* informs resource of activity type and estimated time to realization. Resource checks its availability for the activity by method call *:schedule work*. If a resource is available (its schedule is empty), real time for the activity is calculated (including level of skills, experiences etc.). Unused resources are specified by “idle” state.

Connection to other tools

There are many tools for project portfolio management and control. Typically, the often used tool for complex administration of project portfolio is e.g. Microsoft Project (connection between activities and resources timetables). MS Project does not provide any open interface to direct data intervention into a project, but it provides a way for data import saved in several standard formats, e.g. XML, HTML, MPX, CSV and others. The most suitable (for our purpose) is MPX (Microsoft Project Exchange). This file consists of individual records (information about projects). These records and their order are predefined.

```
MPX,Microsoft Project,4.0,ANSI
30,Project
40,Unique ID,Name,Standard Rate
50,1;John,,$20/h
60,Unique ID,Name,Duration
70,5,"programming",50h
75,1,1,40h
```

Example above describes import/export of a resource with id 1 (noted by 50), name *John* and standard wage per hour is 20 dollars. Than follow: activity (n.70) *programming* with id 5 and resource assigning (n. 75); resources with id 1 on the previous activity with duration 40 hours.

CONCLUSION

The paper presents a new approach to project portfolio modelling and control by object oriented Petri nets and Renew tool usage.

This model is actually under the process of further development and extension. Further refinements on the new parameters used in the model more specific characteristics and attributes of individual projects and resources will be included and tested in future researches. Future work includes enlarging the model with respect to a real situation which can occur in projects.

ACKNOWLEDGMENTS

This work was supported by the Research Plan No. MSM 0021630528 and the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

REFERENCES

Archer N. and Ghasemzadeh F. 2004. *Project portfolio selection and management*, in: P.W.G. Morris, J.K. Pinto (Eds.), The Wiley Guide to Managing Projects, Wiley, New York.

Artto K.A. and Dietrich P.H. 2004. Strategic Business Management through Multiple Projects. In: Morris Peter, W.G., Pinto, J.K. (Eds.), The Wiley Guide to Project, Program, and Portfolio Management. Wiley, Inc., 1-33.

Bible M. and Bivins S. S. 2012. Mastering Project Portfolio Management. J Ross Publishing. ISBN 1604270662.

Billows D. 2005. *Project Management Best Practices.* Hampton Group. ISBN 0-9716820-3-8, 36-53.

Burke R. 2006. *Project Management: Planning and Control Techniques.* John Wiley & Sons. ISBN 0958273316.

Gareis R. 2004. *Management of the project-oriented company*, in: P.W.G.Morris, J.K. Pinto (Eds.), TheWiley Guide to Managing Projects, Wiley, New York.

Girault C. 2003. *Petri Nets for Systems Engineering*. Springer. ISBN 3540412174.

Goodman A. 2006. *Process-based software project management*. AUERBACH. ISBN 0849373042.

Gray C. and Larson E. 2005. *Project Management: The Managerial Process*. Irwin/McGraw-Hill. ISBN 0072978635.

Charvat J. 2003. *Project management methodologies: selecting, implementing and supporting methodologies and processes for project*. New Jersey: John Wiley.

Chen C.H., Ling S.F. and Chen W. 2003. *Project scheduling for collaborative product development using DSM*, International Journal of Project Management 21, 291–299.

Janoušek V. and Květoňová Š. 2009. *Object-Oriented Petri Nets-Based Modeling of Resources in Project Engineering.* In: Computer Aided Systems Theory - EUROCAST 2009. Springer Verlag. ISBN 978-3-642-04771-8.

Janoušek V. and Květoňová Š. 2009. *On the Multilevel Petri Nets-Based Models in Project Engineering*. In: International Workshop on Petri Nets and Software Engineering 2009.

Kendall G. and Rollins S. 2003. *Advanced Project Portfolio Management and the PMO*, J. Ross Publishing, Boca Raton, FL.

Kerzner H. 2009. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley & Sons. ISBN 0470278706.

Kummer, O., Wienberg, F. and Duvigneau, M. 2009. *Renew - User Guide*. University of Hamburg, Department for Informatics.

Lee B. *Multi-project management in software engineering using simulation modeling*, Software Quality Journal 12 (1), 59–82.

Meredith J. R. 2012. *Project Management: A Managerial Approach.* John Wiley & Sons. ISBN 1118093739.

Miguel I. 2004. *Dynamic Flexible Constraint Satisfaction and its Application to AI Planning*. Springer-Verlag.

Moldt D. 1995. *OOA and Petri Nets for System Specification. Application and Theory of Petri Nets*; Lecture Notes in Computer Science, Italy.

Morris P. and Pinto J. 2007. *The Wiley Guide to Project Control*. Wiley. ISBN 0470226846.

Murata T. 1989. "Petri Nets: Properties, Analysis and Applications". In *Proceedings of the IEEE*, Volume 77, Issue 4, 541-580.

Nabrzyski J., Schopf J. and Weglarz J. 2005. *Grid Resource Management State of the Art and Future Trends*. Kluwer Academic Publishers.

Rozenberg G. 1991. *Advances in Petri Nets.* Springer. ISBN 0387538631

Schwalbe K. 2005. *Information Technology Project Management,* Fourth Edition. Course Technology. ISBN 0619215267.

WEB REFERENCES

Microsoft [online]. Available at: http://www.microsoft.com (2012)

Project Management Institute [online]. Available at: http://www.pmi.org (2012)

Renew [online]. Available at: http://www.renew.de (2012)

Robertson, K. 2012. *Project / Program / Portfolio Management – What Does it Really Mean?* [online]. Available at: http://www.klr.com/articles/Articles_PM_project_program_portfolio_mgmt.pdf

BIOGRAPHIES

MICHAL NĚMEC was born in 1988, Vítkov, Czech Republic. Presently, he is a student of Master Study Programme Management and Information Technology at Brno University of Technology, Faculty of Information Technology, Brno, Czech Republic. His main research interests are project management, software engineering and website development.

ŠÁRKA KVĚTOŇOVÁ was born in 1981, Brno, Czech Republic. She has studied Economics and Management at the Brno University of Technology, Faculty of Business and Management, Brno, Czech Republic. In 2009, she finished her PhD. studies at Brno University of Technology, Faculty of Information Technology, Brno, Czech Republic. Presently, she is a lecturer at BUT FIT, Brno. Her main research interests are software engineering, processes and project management with focus on Petri Net techniques and their application for software projects management.

Industry Project Evaluation with the Analytic Hierarchy Process

Sebastian Zimmer
Matthias Klumpp
Hella Abidi
Institute for Logistics and Service Management
FOM University of Applied Sciences Essen
Leimkugelstraße 6, 45141 Essen, Germany
www.fom-ild.de

KEYWORDS

Analytic Hierarchy Process (AHP), Project Evaluation, Power Plant Industry

ABSTRACT

In the power plant industry the decision whether to bid for a tender and work out a proposal or not holds considerable consequences for the plant manufacturer. Adequate project evaluation is crucial. The criteria which influence the success of the evaluation and selection of promising projects are highly diverse. The scales of these criteria are qualitative as well as quantitative. Also the importance of the different criteria differs significantly. Some criteria are negligible whereas others have a dominant weight. In this paper the Analytic Hierarchy Process (AHP) is presented and discussed with regard to the qualification of supporting the bid decision. Based on the three basic project evaluation criteria rrealization chance (go), award chance (get) and profitability a AHP hierarchy was developed which covers all relevant criteria. By means of a small example the analysis, evaluation and the deduction of recommended proceedings is presented.

INTRODUCTION

In the power plant industry success and failure depend on few substantial large projects. In addition to that, these projects appear discontinuously on the market which involves problems with the workload of the plant manufacturer. The power plant industry furthermore is a mature industry, which means that competition is strong and profit margins are low.
Sales in the power plant business have relative high resource requirements. This is mainly due to two reasons. One is that only 5 to 10 % of the proposals lead to an order. Most of the other projects are for different reasons postponed or never realized. The other projects cannot be won. The other reason is that due to the individual customer requirements and the technical complexity of power plants the proposal preparation for a binding offer involves a comprehensive basic engineering design. This requires a high expenditure of time and resources. In the power plant industry proposal costs may account for 2 to 5 % of the total project volume (Jiayi and Mattfeld 2007). For different internal and external reasons some proposals may be more promising than others. Key for successfully winning contracts is the adequate and differentiated allocation of resources to different proposals. The basis for this is an effective and structured analysis and evaluation of different tenders and their environments.

PROJECT EVALUATION IN THE POWER PLANT INDUSTRY

The decision whether to bid for a tender and work out a proposal or not holds considerable consequences for the plant manufacturer. Massive project losses are often caused by inadequate project evaluation. For this reason the project evaluation phase has a special position in the complete tendering procedure (Backhaus 2007).
The success of a project evaluation can be seen as the ratio between the supposed project profit margin, the probability of winning the contract and the proposal costs (Heger 1998). The reasons why the evaluation and selection of projects before the bid decision are necessary are (Schwanfelder 1989):

- Limited resources,
- Proposal costs,
- Risk evaluation.

Limited resources: In the power plant industry the preparation of a proposal for a binding offer involves a comprehensive basic engineering design. For this task highly qualified engineering and management staff is necessary. Furthermore time is limited in the tendering phase. In times of a low project workload staff can be shifted to some extent from project work to proposal work. Anyway this measure can balance high proposal resource requirements only to a certain amount.
When the resource requirements exceed the allocated capacity the required accuracy suffers. This has far-reaching impact on the project success for several reasons. During proposal preparation risks are identified and if possible also limited. When the tender is not completely read and studied by qualified and experienced staff critical details will be not considered. Assumptions and alternatives will not be worked out and required work packages or scope of supply will not be calculated. Finally the proposal documents may be incomplete, incorrect or simply in the wrong language.

All these things may lead to a disqualification, missing limitation of risks and wrong total cost calculation. The consequences are either that the project is given to a competitor and the proposal preparation was in vain or that essential risks were not limited and certain scopes of supplies were not calculated which may lead to a negative project conclusion (Schwanfelder 1989).

Proposal costs: Depending on the size of the intended project and the extent of the required proposal, proposal costs in the power plant business are high. These costs include (Backhaus 2007):

- Acquisition costs like salary and travel costs of sales representatives;
- Project work costs like basic engineering costs, quantity determination costs and pricing calculation costs;
- General sales costs.

As described before proposal costs may account for up to 5 % of the total project volume. These costs have to be covered by the overhead. That means that the ratio between won and lost contracts has a high impact on the overhead structure. The more successful the tender evaluation and selection, the lower the number of prepared proposals and proposal costs. Furthermore lower overhead costs lead to a larger pricing range which again improves the probability of winning contracts (Schwanfelder 1989).

Risk evaluation: The risks for power plant manufacturers are highly diverse in terms of source, impact, probability of occurrence and the avoidance possibilities. For this reason the bid decision includes serious risks. Risk in general can be defined as the probability of a future deviation between a planned and an actual result. Every economic activity with the objective of earnings involves risks. However already in the project evaluation risks must at least be identified if possible avoided and if not possible minimized or assured (Rosenkranz and Missler-Behr 2005, Hempel and Offerhaus 2008).

In order to increase sales success and consequently the success of the company the ratio between lost and won contracts must be improved. The key is the individual and structured evaluation and selection of projects. The criteria which influence the success of the project evaluation and selection are highly diverse. The scales of these criteria are qualitative as well as quantitative. Also the importance of different criteria differs very much. Some criteria might be negligible whereas others might have a very dominant weight. It can be concluded that a multi-criteria decision-making problem is present and that an analytic and systematic approach is necessary to resolve this task.

The typical evaluation procedure for a power plant manufacturer, depicted in figure 1 basically consists in three steps.

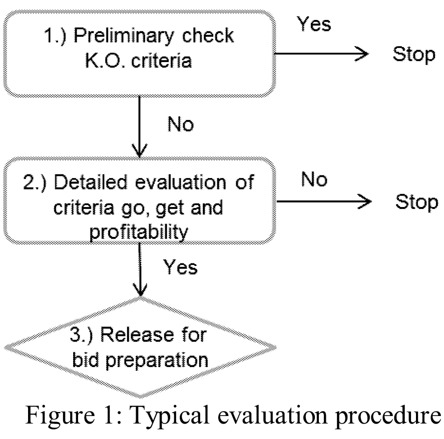


Figure 1: Typical evaluation procedure

The first step in this procedure is a preliminary check for K.O. criteria. These criteria are among others:

- Financing,
- Price level,
- Technical requirements.

Financing: A strong K.O. criterion is the financing of the client. When it turns out that a financial close is not possible or very unlikely the procedure is stopped.

Price level: For a German plant manufacturer the price level is a strong pre-selection criterion. Due to a better cost structure plant manufacturers from China for instance can offer much lower prices compared to German plant manufacturers. (News release U.S. Bureau of Labor Statistics 2009). With such a price level profit margins and risk margins can't be considered. Consequently a German plant manufacturer makes a pre-selection based on the criterion price level or indirectly on the fact that Chinese bidders are tolerated by the building owner.

Technical requirements: The third K.O.-criterion are the technical requirements from the tender. The key competence of a power plant manufacturer is the EPCC project execution of large scale coal fired power plants. All processes are oriented towards the particular requirements for these plants. For this reason tenders deviating from the specific own capabilities are rejected. When a project passes the preliminary check of K.O.-criteria a more detailed evaluation is executed in the second step. The evaluation of projects in the power plant industry is commonly based on three basic criteria:

- Realization chance (go)
- Award chance (get)
- Profitability

Realization chance (go): The criteria in this category are related to the probability of realization. This criterion is also called go-criterion. The result is the realization chance in percentage.

Award chance (get): The get-criteria are related to the probability of winning the project. The result is the award chance in percentage.

Profitability: The third basic evaluation criterion is the expected profit. The main question here is whether it is possible to make profit with this project.

When the project passes the second step of the evaluation procedure, in the last step the release for the proposal preparation is given.

THE ANALYTIC HIERARCHY PROCESS

The Analytic Hierarchy Process (AHP) was developed in the seventies by Thomas L. Saaty. Generally defined the AHP is a procedure for structuring and dissolving multi-criteria decision problems (Meixner and Haas 2002). As the name implies, objectives and criteria are structured analytically in a hierarchical order. Analytic means that the decision problem is analyzed mathematically by means of logical conclusions (Zimmermann and Gutsche 1991). Furthermore the name stresses the procedural character of AHP (Götze 2008). Different alternatives are compared with regard to criteria in a pairwise mode with a fundamental scale of absolute numbers which has been proven in practice and validated theoretically. This procedure converts individual preferences into ratio scale weights. Like this for each alternative a linear additive weight can be obtained which can be used to rank the alternatives and thus support the decision-making (Forman and Gass 2001). In figure 2 the three main levels of the typical AHP hierarchy objective, criteria and alternatives are depicted.

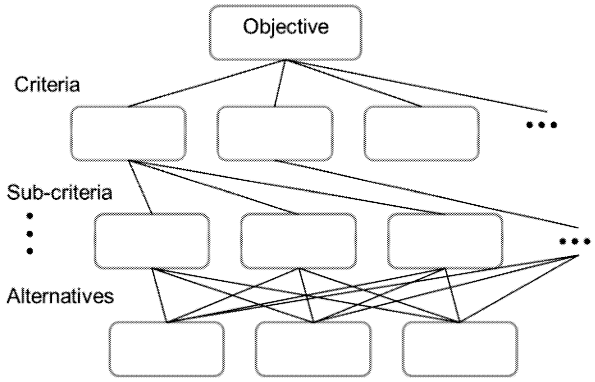


Figure 2: Analytic Hierarchy Process typical hierarchy

For the application of the AHP four axioms must be fulfilled (Ishizaka and Lusti 2006, Harker and Vargas 1987, Saaty 1986, Zimmermann and Gutsche 1991):

- Reciprocal condition
- Homogeneity
- Dependence
- Integrity

Reciprocal condition: Given two alternatives or criteria i and j from a finite amount of elements. The intensity of preference i over j must be reciprocal to j over i.

$$a_{ij} = \frac{1}{a_{ji}} \quad \text{for all } i,j \in A \qquad (1)$$

That means that the decider behaves reciprocal. When the criterion i is twice as important as j, j is half as important as i. Consequently for n elements

$$\frac{1}{2} * n * (n-1) \qquad (2)$$

paired comparisons are necessary.

Homogeneity: When two alternatives $i,j \in A$ are compared with regard to criteria $i,j \in A$ one alternative cannot be infinitely better than the other one.

$$a_{ij} \neq \infty \quad \text{for all } i,j \in A \qquad (3)$$

In case of an infinitely better alternative with regard to one criterion the other alternatives must not be considered and the decision problem is not existent.

Dependence: All elements in a hierarchy are allocated to different levels. A level influences and is influenced only by the next higher or lower level. The elements within a level must not influence each other.

Integrity: The hierarchy contains all relevant criteria and alternatives. When adding or leaving out criteria or alternatives the hierarchy and thus the result might change.

The general AHP procedure is depicted in form of a flow diagram in the figure 3.

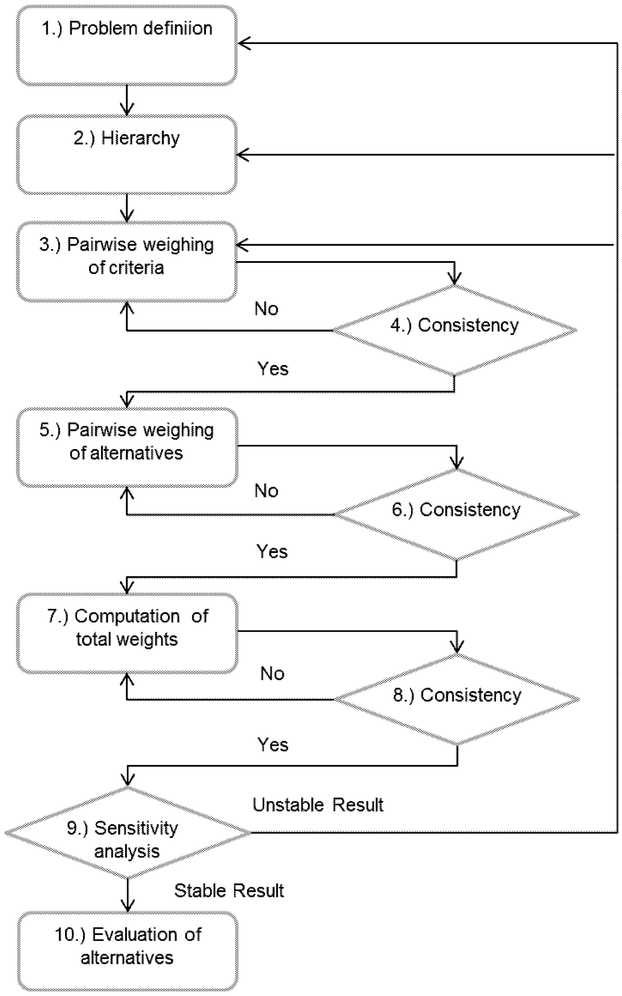


Figure 3: Analytic Hierarchy Process flow diagram

In the first step of the procedure the decision problem is defined. In the second step the decision problem is subdivided into different criteria, sub-criteria and alternatives. The elements of one level must belong to the same category. That means they are comparable to each other. Furthermore it is necessary that the evaluations are independent from evaluations in the same or in other levels of the hierarchy. Finally it has to be assured that all relevant criteria and alternatives are considered (Götze 2008). In the steps 3 to 5 each criteria of one level is compared to all other criteria of the same level. The results are summarized in a matrix (Zimmermann and Gutsche 1991). Then for the criteria of each level partial weights can be calculated. When all pairwise comparisons are consistent (step 4 and 6) the total weights can be calculated in step 7. Afterwards the total consistency is calculated in step 8. When the consistency is good a sensitivity analysis can be carried out. For this purpose the criteria weights are changed to check whether the model is stable. When the stability is good a final evaluation of the single alternatives is carried out. When the consistency in the steps 4, 6 and 8 is not good, or the results from the sensitivity analysis are not satisfying, the complete process is iterated with optimizing step 1 to 3 (Meixner and Haas 2002).

Problem definition and hierarchy

In the first step the decision problem is defined. At the end of this step the problem should be described precisely in one sentence: “The decision problem, which forms the evaluation and selection of projects in the power plant industry.” Based on the problem definition the AHP hierarchy is developed. The definition of the problem and the development of the hierarchy forms the basis for the complete decision-making process. The achieved quality in these steps of the procedure determines the quality of the overall result. For this reason special attention and a high proportion of time has to be given to this part (Meixner and Haas 2002). Like shown in figure 2, the decision problem shall be subdivided individually into different criteria, sub-criteria and alternatives. The elements of one level shall be of the same category. That means they shall be comparable to each other. Furthermore it is necessary that the evaluations are independent from evaluations in the same or in other levels of the hierarchy. Finally it is assumed that all relevant criteria and alternatives are considered (Götze 2008). There is a wide range of different types of decision problems existing. Possible structures of an AHP hierarchy might be (Meixner and Haas 2002):

- Utility hierarchy
- Cost hierarchy
- Risk hierarchy
- Probability hierarchy

Decision problems often require the combination of different types of hierarchies. This is also the case for the project evaluation in the power plant industry. All the objectives listed above can be used for such an evaluation. The utility of a project evaluation can be interpreted as winning a contract with good conditions. Cost optimization may be allocated to the costs for the preparation of the proposal. Risks are involved in all entrepreneurial activities.
In literature different AHP structures are proposed for the evaluation of projects in the construction industry. In order to reduce complexity the criteria are subdivided into internal and external criteria. Amad and Minkarah suggest a classic internal and external structure. The internal criteria are again subdivided into company related and project related criteria (Admad and Minkarah 1988). This structure possesses the strength that the sources of the criteria are highlighted. Project evaluation can be structured with the two main criteria risk and opportunity (Dikmen and Birgonul 2006). This structure holds the advantage that general key success factors are addressed directly.

Scale of measurement

In the steps 3 to 5, each criteria of one level is compared to the other criteria of the same level. The results are summarized in a matrix (Zimmermann and Gutsche 1991). For this purpose a scale of measurement is introduced. AHP allows the comparison of quantitative criteria as well as qualitative criteria. For this a common scale is required. The 1 to 9 point scale proposed by Saaty is shown in the table 1.

1	Equal Importance	Two activities contribute equally to the objective
2	Weak Importance	
3	Moderate Importance	Experience and judgment slightly favor one activity over another
4	Moderate plus	
5	Strong Importance	Experience and judgment strongly favor one activity over another
6	Strong plus	
7	Very strong or demonstrated Importance	An activity is favored very strongly over another; its dominance is demonstrated in practice
8	Very, very strong Importance	
9	Extreme Importance	The evidence favoring one activity over another is of the highest possible order of affirmation

Table 1: 1 to 9 point scale (Saaty and Vargas 2001)

The 1 to 9 point scale has proven to be an acceptable scale of measurement. The reason is that “a 1 to 9 scale can accurately portray an individual’s intensity of preference” (Harker and Vargas 1987). The scale values are considered to be relative values. 1 is allocated to equal importance which means that it represents the zero point (Meixner and Haas 2002).

Pairwise comparisons

The ratio between the compared criteria is:

$$a_{ij} = \frac{w_i}{w_j} \quad (4)$$

Considering the first axiom the reciprocal is:

$$a_{ij} = \frac{1}{a_{ji}} \quad (5)$$

The resulting ratio comparisons are depicted in

$$\begin{pmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \cdots & \frac{w_2}{w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \cdots & \frac{w_n}{w_n} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = n \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \quad (6)$$

In order to obtain the term $n * w$ the matrix is multiplied by w on the right (Saaty and Vargas 2001).

Priorities

One objective of the AHP is the derivation of weights and priorities from paired comparisons. There are different methods available to determine the priority vector (w). One is Eigen Vector Method (Saaty and Vargas 2001). In table 2 the paired comparisons are linked with normalizing the ratios. First the column sums (c) of the evaluation matrix have to be formed. Then the ratios (a) are divided by the column sums (c). The result is the normalized matrix scaled to 1. The priority vector (w) of the corresponding elements is achieved when the row sum (r) of the normalized matrix is divided by the number of elements (n). With this procedure corresponding to the hierarchy level the individual priority vectors can be achieved (Meixner and Haas 2002).

	Evaluation Matrix				Normalization					Weight
	a_1	a_2	....	a_n	a_1	a_2	....	a_n	r_i	
a_1	$a_{11} = 1$	a_{12}	....	a_{1n}	a_{11} / c_1	a_{12} / c_2	....	a_{1n} / c_n	r_1	$w_1 = r_1 / n$
a_2	$a_{21} = 1/a_2$	1	....	a_{2n}	a_{21} / c_1	a_{22} / c_2	....	a_{2n} / c_2	r_2	$w_2 = r_2 / n$
⋮	⋮	⋮		⋮	⋮	⋮		⋮	⋮	⋮
a_n	a_n	a_{2n}	....	$a_{nn} = 1$	a_n	a_{22} / c_2	....	a_{2n} / c_2	r_2	$w_2 = r_2 / n$
c_i	$c_1 = \sum_{i=1}^{n} a_{i1}$	$c_2 = \sum_{i=1}^{n} a_{i2}$	....	c_n	1	1	....	1	n	1

Table 2: Weight calculation with Eigen Vector Method

Consistency

One of the major advantages of the Eigen Vector Method over other multi criteria decision making methods is the ability to deal with inconsistencies. This is particularly important due the inability of human beings to have a 100 % precise judgment (Harker and Vargas 1987). Another source of inconsistencies may be the 1 to 9 point scale. In some cases this scale is not precise enough which can lead to inconsistencies (Nitzsch 1993, Rommelfanger and Eickemeier 2001). In the steps 4, 6 and 8 in the AHP procedure, the consistency of the pairwise comparisons made in has to be checked.

The evaluation matrix is consistent when $a_{ij} * a_{ij} = a_{ik}$ (Götze 2008). For the evaluation of the consistency first the row sum of the average matrix (ri□) has to be calculated. This term is obtained by forming an average matrix depicted in table 3.

	a_1	a_2	....	a_n	$\bar{r}_i$
a_1	$w_1 * a_{11}$	$w_2 * a_{12}$	....	$w_n * a_{1n}$	$\bar{r}_i = \sum_{i=1}^{n} w_i * a_{1i}$
a_2	$w_1 * a_{21}$	$w_2 * a_{22}$	....	$w_n * a_{2n}$	$\bar{r}_2$
⋮	⋮	⋮		⋮	⋮
a_n	$w_1 * a_{n1}$	$w_2 * a_{n2}$	....	$w_n * a_{nn}$	$\bar{r}_n$

Table 3:1 Calculation of average matrix

Then each Eigen Value has to be calculated with:

$$\lambda_i = \frac{\bar{r}_i}{w_i * a_i} \quad (7)$$

For n elements that means:

$$\begin{pmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \vdots \\ \bar{r}_3 \end{pmatrix} \div \begin{pmatrix} w_1 * a_{11} \\ w_2 * a_{22} \\ \vdots \\ w_n * a_{nm} \end{pmatrix} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_3 \end{pmatrix} \quad (8)$$

The Principal Eigen Value (λ_{max}) of the average matrix is:

$$\lambda_{max} = \frac{\sum_{i=1}^{n} \lambda_i}{n} \quad (9)$$

For consistent evaluation λ_{max} is always n (Saaty 1990).
The Consistency Index (CI) is obtained with:

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (10)$$

CI is compared with a Random Consistency Index obtained as an average over a high amount of reciprocal matrices of the same order. This Random Consistency Index is depicted below.

n	1	2	3	4	5	6	7	8	9	10	11
RI	0,00	0,00	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49	1,51

Table 4: Random consistency index

The Consistency Ratio is calculated with:

$$\mathbf{CR} = \frac{\mathbf{CI}}{\mathbf{R}} \quad (11)$$

Saaty suggests rethinking the evaluation of the pairwise comparisons when CR is about or above 10 % (Saaty 1990).

Sensitivity analysis

Effects on the rank of alternatives by changing criteria weights are analyzed in the sensitivity analysis. By means of continuous shift of criteria weights, the limits which lead to a change in the rank of alternatives are determined. A stable result is obtained when marginal changes in the paired comparisons do not lead to a change in the rank of alternatives. When marginal changes in the paired comparisons lead to a change in the rank of alternatives the procedure has to be reviewed. In some cases even the problem definition has to be reworked (Meixner and Haas 2002, Weber 1993).

Appraisal

The AHP is a very popular and widely-used multi-criteria decision-making method. The main advantages of the AHP are listed below:

- Structure complexity
- Measurement on a ratio scale
- Interdependence of preferences
- Synthesize results
- Consistency check
- Process repetition

Structure complexity: Better than any other multi-criteria decision-making method the AHP helps to structure complexity. The hierarchical structure and the deductive approach with an objective, criteria and sub-criteria is the most common and natural way for humans to deal with complexity. This structure supports the decider with regard to an analysis of the decision problem (Forman and Gass 2001).

Measurement on a ratio scale: Another advantage of the AHP is the measurement on a ratio scale. The measurement on a ratio scale allows the formation of ratios and proportions among ratio scaled values. In the AHP the decider gives judgments regarding a ratio for criteria and different alternatives or other criteria (Forman and Gass 2001). On the one hand this enables comparison of criteria with qualitative as well as quantitative scales. On the other hand this procedure facilitates the limited human capability to rational judgment. For this reason for decision problems with more than three criteria AHP unfolds its full potential (Girmscheid 2010).
Interdependence of preferences: In contrast to the utility theory based methods, the AHP can deal with interdependences of criteria. The substitution condition is not to be fulfilled. For this reason a complex preference independence check is not necessary (Saaty 2001, Schneeweiß 1991).

Synthesize results: Analysing and structuring the decision problem is the first step to reduce complexity. The overall objective is structured in several criteria and sub-criteria. The alternatives are evaluated with regard to these criteria. The next step is the synthesis. The individual results are put together and combined into one total result. The ability of AHP to measure and synthesize the multitude of factors in a hierarchy is outstanding (Forman and Gass 2001).

Consistency check: Another advantage of AHP over other methods is that a consistency check is integrated in the procedure. The reliability of decisions and evaluations with a low consistency is lower than for decisions and evaluations with a high consistency. Only decisions and evaluations with a defined consistency are acceptable. Other decisions and evaluations might be at random. This is particularly important for multi-criteria decision-making problems with a large number of criteria (Meixner and Haas 2002).

Process repetition: "The AHP enables people to refine their definition of a problem and improve their judgment and understanding through repetition" (Saaty 2001). In other words AHP helps people to get a better understanding of the problem. The main critics of the AHP are listed below:

- Fundamental scale
- Rank reversal
- Computation complexity

Fundamental scale: Frequent points of criticism of the AHP are related to the measurement scale. The fundamental 1 to 9 scale is used to measure the importance of one criterion over another. The rationality of this scale is not legitimated. On the one hand other scales might lead to a rank reversal of the alternatives.
On the other hand the definitions of the preferences are not clear. In practice it is often difficult to differentiate for instance between moderate or strong importance (Zimmermann and Gutsche 1991, Holder 1990).

Rank reversal: When the decision-maker adds a further irrelevant alternative to a set of alternatives it can happen that the ranking of the major alternatives changes. This phenomenon is called rank reversal (Saaty and Vargas 1984). In the 1990s there was a big discussion whether AHP is completely flawed and the rankings obtained are arbitrary (Dyer 1990). Today it is common sense that rank reversal can handled with adequate measures like a correct structure. For the discussion and the technical details of this aspect of AHP more information is provided in the literature (Barzilai and Golany 1994, Gass 2005).

Computation complexity: With an increasing number of criteria in one level the number pairwise comparisons and individual calculations increases exponentially (Götze 2008).

PROJECT EVALUATION WITH THE AHP

Criteria and Hierarchy

For the first level criteria in the AHP hierarchy the basic project evaluation criteria realization chance (go), award chance (get) and profitability are chosen. All other relevant criteria for the evaluation of projects in the power plant industry can be allocated to these three basic criteria. In the figure 4 the AHP hierarchy of the project evaluation with the defined criteria and all relevant sub-criteria is depicted.

Figure 4: Project evaluation AHP hierarchy

In table 5 the individual criteria are defined and a ratio scale is allocated to each individual criterion. The individual weightings (W) of the criteria and sub-criteria are also listed. The data was obtained in a workshop with managers of the sales department of a German power plant manufacturer.

Criteria name	Definition	W	Criteria scale
Realization chance (go)		**0,43**	
Financial close	How sure is the project financing of the client	0,35	Closed 100 Likely 50 Unlikely 0
Politics	Influence of politics/ public opinion on realization of the project. For instance influence on permit procedure or subsidies for the project. Political stability during tendering.	0,17	Positive 100 Neutral 50 Negative 0
Demand	How strong does the client require the power plant with regard to the realization of his objectives? Are there alternative investments?	0,08	High 100 Average 50 Low 0
Client rating and standing	What is the client's rating?	0,05	Positive 100 Neutral 50 Negative 0
Permitting	What are the chances that all relevant permits will be given?	0,35	Closed 100 Likely 50 Unlikely 0
Award chance (get)		**0,43**	
Competition	Number of potential serious competitive bidders.	0,31	2 33 1 50 0 100
Market condition	Other projects currently out to tender in the same market	0,11	Many 100 Average 50 Few 0
Relationship with the client	How was the relationship with the client in the past	0,11	Good 100 Fair/ No 50 Bad 0
Price Level	How is the market price level compared to the own estimated price	0,31	Higher 100 Even 50 Lower 0
Qualification	Is the plant manufacturer qualified (technical references, management, resources, financial resources) to execute the project?	0,05	Minimum 0 Solid 50 Excellent 100
Market knowledge and presence	Does the plant manufacturer know how the market where the client operates. Is the plant manufacturer active on this market e.g. branch office, core market?	0,11	No 0 Average 50 Yes 100
Profitability		**0,14**	
Price Quality	Estimated award price in comparison to own rock bottom price	0,31	High 100 Average 50 Low 0
Value added	Own products and services related to total products and services under the contract	0,31	High 100 Average 50 Low 0
Required realization time	Required realization time compared to the time schedule	0,14	Tight 0 Normal 50 Convenient 100
Conditions of payment	Prevailing cash position	0,06	Positive 100 Neutral 50 Negative 0
Project risk	Result of risk assessment	0,14	High 0 Average 50 Low 100
Strategic relevance	Is it, for some reason, not required to be profitable e.g. prestige of the project, market share, follow up projects, new technologies?	0,04	High 100 Average 50 Low 0

Table 5: Evaluation criteria realization chance award chance (get) and profitability

As indifference judgment could have been made for the main criteria realization chance and award chance the consistency of the main criteria is perfect. The consistency of the pairwise comparisons of the sub-criteria is also given.

The results obtained in an evaluation of different projects by means of the criteria listed in table 5 can be represented in two different ways: (a) Absolute evaluation score: An example of the depiction of the evaluation results for four different projects with an absolute score is shown in figure 5. Essentially this shows a ranking of the projects. The absolute scores of the weighed main criteria add up to an absolute evaluation score. Project 2 would be the preferred project.

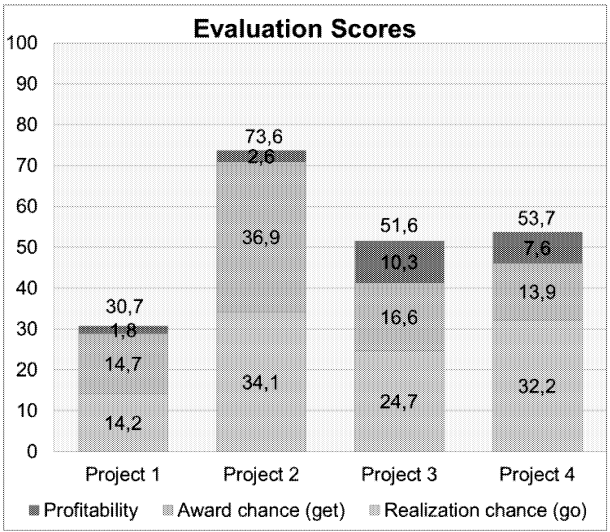


Figure 5: Absolute evaluation score

(b) Screening portfolio: In the screening portfolio shown in figure 6 the unweighed criterion award chance is plotted versus the criterion realization chance. The profitability is represented by ball size. Like this a screening portfolio can be developed where recommendations can be deduced.

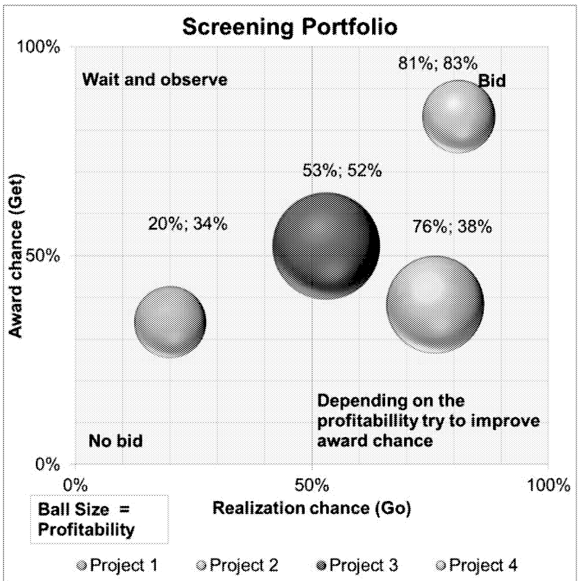


Figure 6: Screening portfolio

CONCLUSION

Before, AHP is presented and discussed with regard to qualification of supporting bid decisions in the power plant industry. Based on the three basic project evaluation criteria realization chance (go), award chance (get) and profitability an AHP hierarchy was developed. Individual weightings were obtained in a case study within a German power plant manufacturer. As an example the analysis, evaluation and deduction of recommended proceedings is presented. For obtaining, processing and recording information it is recommended to apply IT based systems. Further, different criteria data sets for relevant market segments and clients can be developed. For each new project evaluation, one can access the data base and use it for the evaluation. Based on this data base also standardized recommended proceedings with regard to resource allocation might be developed. Because of additional complexity, judgment uncertainty has not been considered so far. In the case study it turned out that the certainty of the judgments covers the complete range and this might have considerable impact on results. Thus it is recommended to discuss the incorporation of uncertainty in the evaluation method and to weigh up the improved quality of the results against additional complexity in the procedure.

BIBLIOGRAPHY

Admad I., Minkarah, I. A. (1988): *Questionnaire Survey on Bidding in Construction*. In: Journal of Management in Engineering, 4 (3), p. 229-243

Backhaus K. (2007): *Industriegütermarketing*. München: Verlag Vahlen

Barzilai J., Golany B. (1994): *AHP rank reversal, normalization and aggregation rules*. In: Information Systems and Operations Research 1994, 32, p. 57-64

Dikmen I., Talat Birgonul M. (2006): *An analytic hierarchy process based model for risk and opportunity assessment of international construction projects*. In: Canadian Journal of Civil Engineering 2006, 33, p. 58-68

Dyer, J. S. (1990): *Remarks on the Analytic Hierarchy Process*. In: Management Science 1990, 36, p. 249-258

Forman H. E., Gass S. I. (2001): *The Analytic Hierarchy Process - An Exposition*. In Operations Research 2001, Vol. 49, No. 4 ,P. 469-486

Gass S. I. (2005): *Model World: The Great Debate- MAUT Versus AHP*. In: Interfaces 2005, Vol. 35, P. 308-312

Girmscheid G. (2010): *Angebots- und Leitfaden für Bauunternehmen*. Berlin, Heidelberg, New York: Springer Verlag

Götze U. (2008): *Investitionsrechnung: Modelle und Analysen zur Beurteilung von Investitionsvorhaben*. Berlin: Springer Verlag

Harker P. T., Vargas L. G. (1987): *The theory of ratio scale estimation: Saaty's Analytic Hierarchy Process*. In: Management Science, 33 (11), p. 1383-1403

Heger G. (1998): *Anfragebewertung*. In: Kleinaltenkamp M., Plinke W. (eds.): *Auftrags und Projektmanagement*. Berlin: Springer Verlag

Hempel M. Offerhaus J. (2008): *Risikoaggregation als wichtiger Aspekt des Risikomanagements*. In: Risikoaggregation in der Praxis, Deutsche Gesellschaft für Risikomanagement e.V. Berlin: Springer Verlag

Holder R. (1990): *Some comment on the Analytic Hierarchy Process*. In: Operations Research, 41, p. 1073-1076

Ishizaka A and Lusti M (2006): *How to derive priorities in AHP: A comparative study*. In: European Center of Operations Research, 14, p. 387-400

Jiayi Y, Mattfeld D. C. (2007): *Entscheidungsunterstützung für die Ressourcenallokation im internationalen Großanlagenbau*. In: *Management logistischer Netzwerke*. Berlin, Heidelberg, New York: Springer Verlag

Meixner O., Haas R. (2002): *Computergestütze Entscheidungsfindung. Expert Choice und AHP - innovative Werkzeuge zur Lösung komplexer Probleme*. Frankfurt, Wien: Redline Wirtschaft

Nitzsch R. (1993): *Analytic Hierarchy Process und Multiattributive Werttheorie im Vergleich*. In: Wirtschaftswissenschaftliches Studium, 3, p. 111-116

Rommelfanger H. J., Eickemeier S. H. (2001): *Entscheidungstheorie: Klassische Konzepte und Fuzzy-Erweiterungen*. Berlin, Heidelberg, New York: Springer Verlag

Rosenkranz F., Missler-Behr M. (2005): *Unternehmensrisiken kennen und managen. Einführung in die quantitative Planung*. Wiesbaden: Gabler

Saaty T. L. (1986): *Axiomatic Foundation of the Analytical Hierarchy Process*. In: Management Science, 32, p. 841-855

Saaty T. L. (1990): *How to make a decision: The analytic hierarchy process*. In: European Journal of Operational Research, 48, p. 9-26

Saaty T. L. (2001): *Decision Making for Leaders. The Analytic Hierarchy Process for Decisions in a Complex World*. Pittsburgh: RWS Publications

Saaty T. L., Vargas L. G. (2001): *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*. Boston et al: Kluwer Academic Publishers

Schneeweiß C. (1991): *Planung 1. Systemanalytische und entscheidungs theoretische Grundlagen*. Berlin, Heidelberg, New York: Springer Verlag
Schwanfelder W. (1989): *Internationale Anlagengeschäfte.* Wiesbaden: Gabler
U.S. Bureau of Labor Statistics (2011): *International comparisons of hourly compensation costs in manufacturing 2009*.
URL: http://www.bls.gov/news.release/pdf/ichcc.pdf
Weber, K. (1993): *Mehrkriterielle Entscheidungen.* Wien, München: Oldenbourg
Zimmermann H.-J. und Gutsche L. (1991): *Multi-Criteria Analyse: Einführung in die Theorie der Entscheidungen bei Mehrfachzielsetzungen.* Berlin: Springer Verlag

MODEL BASED DECISION SUPPORT SYSTEM FOR COMMUNAL WATER NETWORKS

Marcin Stachura
Warsaw University of Technology
Institute of Automatic Control and Robotics
Ul. Sw. A. Boboli 8, 02-525 Warsaw
E-mail: m.stachura@mchtr.pw.edu.pl

Bartłomiej Fajdek
Warsaw University of Technology
Institute of Automatic Control and Robotics
Ul. Sw. A. Boboli 8, 02-525 Warsaw
E-mail: b.fajdek@mchtr.pw.edu.pl

Jan Studziński
Polish Academy of Science
Institute of System Research
Ul. Newelska 6, 01-447 Warsaw
E-mail: jan.studzinski@ibspan.waw.pl

KEYWORDS

Decision support, water distribution system, modeling, simulation

ABSTRACT

In the paper an idea of a computer aided decision support system for the maintenance of water distribution system is presented. The system uses mathematical models for computer simulation, optimization and control of all links composing the complex system of drinking water management realized in waterworks. The developed methods and algorithms are briefly described. With the system proposed an active control of waterworks could be carried out, whereas the systems applied up to date in waterworks work without any models. They make only a passive control of waterworks' separated objects, i.e. they try to stabilize the parameters of different processes on the given levels of values. The aim of the paper is to indicate a new direction in the field of computer aided decisions making for maintenance of waterworks.

Research was partially financed by National Center of Research and Development (NCBiR) in a project no. 14-0011010/2010

INTRODUCTION

Municipal water and sewage systems

Municipal water and sewage systems typically consist of four subsystems: water supply, water distribution system, sewage network and waste water treatment plant. These 4 subsystems are commonly treated separately as independent units although they create in reality 1 complex system consisting of 4 modules sequentially connected.

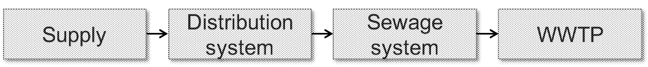


Figure 1: Municipal water and sewage systems block diagram.

Water distribution system seems to be a key element in this system. Its load affects the performance of pumping stations of water intake, determines the hydraulic load of the sewage network and treatment plants. Correct management of water supply network decides on the appropriate quality and quantity of water supplied to customers. Therefore, one of the fundamental tasks in the water supply is the effective management of the network. This task could be divided into a number of subtasks, such as:

- supplying drinking water,
- taking care of water quality,
- supply water under sufficient pressure,
- fault prevention, in case of emergency, its location and fast removal
- planning and implementing network rehabilitations,
- modernization and expansion of the network,
- prevention of water loss,
- planning price of water for the next calendar year, etc.

All these tasks are performed in each water company. Usually they are treated independently and are performed by different divisions of a company. Actually, these tasks are interrelated and should be implemented taking account of this interdependence. Furthermore, their proper implementation depends on a common set of information which may be obtained by installing the adequate water distribution monitoring system.

Management of water supply company

Despite the ongoing computerization of the polish water companies, its management is still based on an individual's experience and intuition. To a lesser extent it is based on the IT systems which are not treated as tools to support decision-making processes. In (Bogdan, Kaczmarska and Studziński 2005) the state of computerization of the national water supply monitoring, digital maps and hydraulic calculations is presented .The paper shows that even in the purely technical operations a lack of integration of different tools and functions is observed. Currently, domestic water companies do not have integrated decision support systems. In general, the incomplete water supply monitoring systems are used to collect data and perform simple analysis of it. The problem of water supply network management concerns also an optimization of its structure and parameters, energy-efficient operation, proper distribution of water of adequate quality, detecting and locating faults, development of the plants. The existing water distribution networks in Poland, currently do not apply the load forecasting models, or the water supply system or decision support systems controlling the network.
The introduction of a decision support system for the water supply network control will allow predicting a load and

improve operational control of the network. In particular, it would help to perform more efficient control of pumps in the network, monitoring water levels in reservoirs and respond to failures in the network.

A COMPUTER SYSTEM FOR WATER SUPPLY NETWORK MANAGEMENT

System description

Decision support systems can be described as tools that combine the possibility of collecting and processing large amounts of data, using a variety of models and intelligent application of the collected data and knowledge. Decision support systems are widely applied in many areas of life, especially in solving problems for which there is no straight solution.

Decision support systems are modern IT tools that provide managers with reliable information, both at strategic and operational level. Users of the system receive the instruments, which in a short time to help transform the data distributed in a variety of enterprise information systems in a consistent information. Systems of this class enable you to monitor indicators of historic, early detection of potential threats and opportunities, as well as forecasting possible business scenarios.

The basic tool supporting a management of a water company is the monitoring system. In general, a monitoring system is a computer system composed of three basic components: equipment for measuring water pressure and flow, data transmission system and a visualization software. A water supply monitoring system can operate in a water supply company as an autonomous system. If it is adequately accurate, it is able to perform several tasks independently, and help to significantly improve network management. However, it is more advantageous to implement a comprehensive, enterprise decision support system, in which monitoring would be the only one of its elements. Effective application of a monitoring system assumes the possibility to work in parallel with a hydraulic model of a water distribution network.

To use data gathered from a monitoring system an adequate branch database is necessary. The structure of this branch database comprises the intake and water treatment station, and water supply system. Creation of a database means defining the objects in specific systems within the water supply and sewage network and their attributes. Such a database could, in the future, become a standard for the water supplying companies which, nowadays, create their own databases individually, within the water supply and sewage network separately.

An idea of an integrated, computer aided decision support system for the city water supply system was formulated subsequent to above stated observations. The central function of water in this system is performed by a monitoring system.. The system is based on a hydraulic model that calculates flows and pressures in the network. It uses mathematical models of the analyzed network, used for simulations and estimating non measured parameters, as well as optimization algorithms used for facilitating the functioning of objects and generating scenarios for taking actions in the regular operation or in the cases of failure.

Energy-saving pumps management in the water-supply system as well as supervising the level of water in retention tanks are a problematic issue of primary importance. At the beginning the study aimed at evaluating whether the parameters of water pressure and flow fall within the norm of tolerance of the load of water-supply system, were carried out. In order to do so, an expert system for signaling the necessity of switching the pumps on was created.

System functions

The proposed decision support system performs several functions for managing production of water and the management of a water supply company.

The hydraulic model of a water distribution system.

The basic computational tool supporting the management of water distribution network is a hydraulic model that calculates water flow and pressure in a network. This model is described by linear and nonlinear algebraic equations, similar to the equations describing the balance of intensity and tension currents in electrical networks (Todini and Pilati 1987). A mathematical description results from the first and second Kirchhoff's law known from electrical engineering. For the formulation of equations of a model, a structure of an investigated network has to be known. Basically, it consists of links (pipes), nodes (pipe junctions), pumps, valves and storage tanks or reservoirs. Hydraulic model calculates the water flow in each pipe, the pressure at each node, the height of water in each tank, and the concentration of a chemical species throughout the network during a simulation. In addition, the water age can also be simulated (Rossman and Boulos 1996).

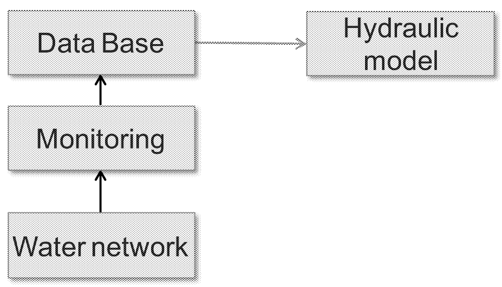


Figure 2: Block diagram of the system: hydraulic model.

Measurement points planning.

Selection of the correct measurement points for monitoring the water supply network is a non-trivial task to solve. Several different algorithms might be used to estimate the best positions of the monitoring devices. One of them, not necessarily the best, is an algorithm as given in: (Straubel and Holznagel 1999). It consists of the following four steps:

1. Multi criteria optimization for determining the number and location of measurement points, with the following criteria: minimum number of points, maximum sensitivity of leak location, minimum installation cost of measuring points
2. Execution of numerical simulations using a hydraulic model of water distribution system, for a standard state of a network. For a given average network load, nodal pressures are determined at all the nodes and flows in all sections of the network.

3. Sorting potential locations in order of their sensitivity to pressure changes in a network.

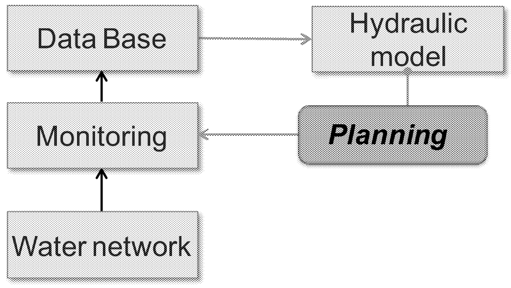


Figure 3: Block diagram of the system: measurement points planning.

Fault and unauthorized consumption of water detection

In the case of normal operation of the network there is a typical pressure distribution on the nodes. A failure is an unusual situation that effects in the unusual variation in the nodal pressures. The decision support system is detecting faults in two ways: by comparing the actual demands with declared ones (from a database, for points where monitoring devices are installed), and by comparing the actual demands with values calculated by a hydraulic model (for those that are not monitored).

Often, water is stolen from a network. This means that the production of water exceeds its documented consumption. In the case of lack of a monitoring system, it can only be concluded that such a situation occurred. Conversely, if there is a precise monitoring system and a precise hydraulic model, a comparison of recorded (or modeled) demands and water flows can be made. In such a situation it is possible to locate the places where significant disparities between consumption and production are visible.

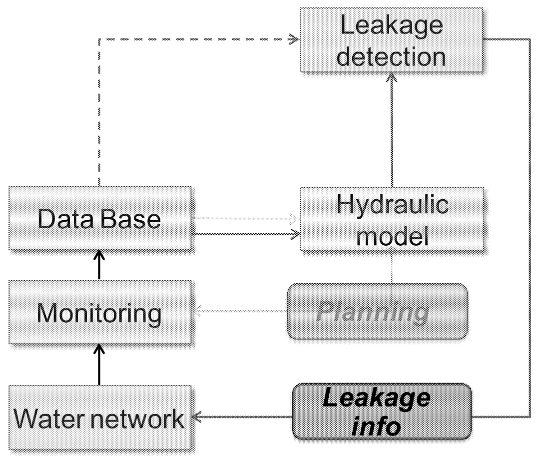


Figure 4: Block diagram of the system: leakage detection.

Control of water flow speed in a network

The problem relates to improving the quality of water circulating in the water distribution system. Water fed into a network after treatment, may lose its taste and smell, if it stays too long in a network. This might happen particularly in the case of old network made of steel pipes, cast iron or cement, which corrode over time, or are overgrown with different types of algae. To avoid deterioration of water quality, an operator (or control algorithm) should adequately enforce its rapid flows in the network through the appropriate pump control or the state of valves installed on a network. A precise hydraulic model of a network is a crucial component for the determination of adverse water outages and the implementation of such control.

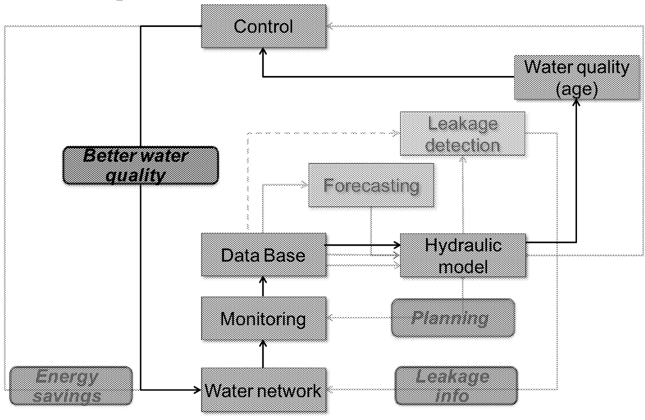


Figure 5: Block diagram of the system: control of water flow speed in a network.

Prediction of possible faults

The problem of faults forecasting is a complex issue. Successful prediction of possible faults allows the execution of any revitalization of a network, before a fault occurs. This means that it is possible to avoid the costs associated with uncontrolled water leakage and a lack of water supply to customers.

Basically, there are three methods of predicting the possible faults. The first group of methods is based on estimating a risk of a fault and proposing the activities for the case of fault. The second group of methods is based on predicting faults and elimination of possible causes of their occurrence, namely a prevention of accidents. Finally, a third group of methods is based on locating the faults that just occurred so as to minimize losses caused by them. In the second and third group of methods for the key role is played by a monitoring system and a hydraulic model of the network.

Network load forecasting.

A network load changes periodically, depending on season, day of a week or time of a day. These changes affect a control of a network, in particular filling and draining expansion tanks located on the network. When the bigger demands are forecasted the tanks should be filled in advance, while they can be emptied in case of lower demands for water. For the pumps are very energy consuming, then a reduction of the energy costs of only few per cent could make notable savings in the waterworks during the year time. Some problems these problems are already investigated and in (Waterworth 2002) an algorithm is discussed to minimize the energy consumption of waterworks pumps by their time distributed switching on and off and under additional consideration of their wear time and the necessary exchange against the new machines. But an implementation of such efficient pump scheduling is possible only when an exact forecast model of the water network load is known and to develop that model one has to consider the intake station, the water reservoirs and the water network as one integrated water system.. So, forecasting a network load means to prepare a precise numerical model that is capable to predict demands of the future load.

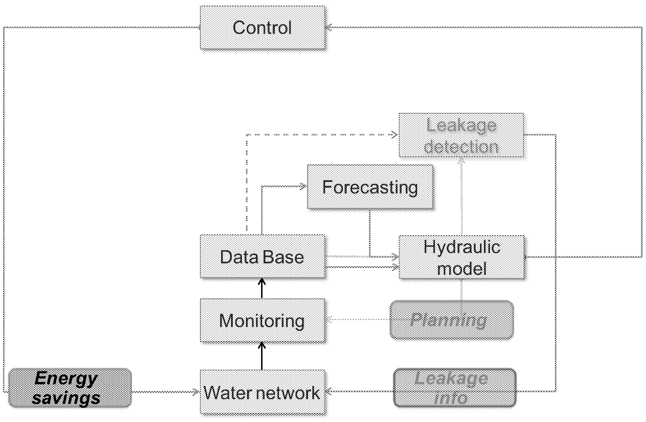


Figure 6: Block diagram of the system: Network load forecasting.

Planning a network rehabilitation

The method to counter the risk, or minimize the probability of fault in a water distribution system is implemented in the form of a complex integrated IT system. It generates scenarios for a work of revitalizing water distribution system for pipes that have been previously identified as most prone to failure. Revitalization planning allows to perform the following tasks:

- identification of technical condition of water distribution system and its individual parts,
- identifying the most vulnerable for faults sections,
- generating revitalization scenarios,
- identifying investment possibilities of a company.

During revitalizations planning numerical simulations of a hydraulic model are carried out. This allows to predict a network load in different scenarios of revitalization.

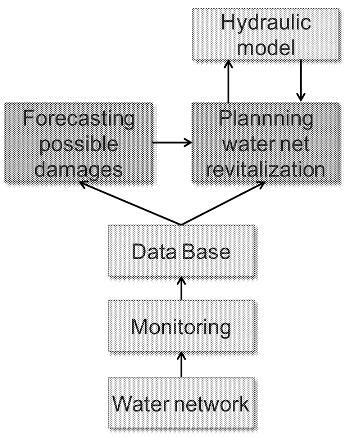


Figure 6: Block diagram of the system: forecasting possible damages and planning a network rehabilitation.

CONCLUSIONS

National water supply companies are usually government based partnerships governing the city sewage and water supply networks consisting of a water intake station, water supply network, sewage network and a sewage treatment plant. These objects, usually separately administered, are, in fact, interconnected in one system; however the functioning of each object influences the work of others. The present state of computing in water supply companies leaves a lot to be desired. The system is usually modernized by installing incomplete monitoring systems in water supply networks or drawing numerical maps of water supply networks. Mathematical models for calculating hydraulic water supply systems are rarely used. Operating the pumping stations at water intakes and water supply networks as well as coordinating the filling up of the equalizing tanks and aeration of sewage is usually performed by automatic regulation systems ensuring the keeping of the regulators as set. Companies use no models of water distribution networks, estimation algorithms, optimization and operational management, which simplify both technically and financially the operation of these objects.

With a view to complete decision aiding, an integrated computer system should be introduced, one equipped both with overt and covert knowledge. That is why the computer aided decision making system in the city water supply and sewage network, shall be enriched with numerical models and algorithms.

FUTURE WORK

In the near future an implementation of the proposed system is planned in Głubczyce (town in southern Poland, app. 13 000 citizens.). Water supply network in Głubczyce consists of the following elements:

- water supply distribution mainlines with a length 34.8 km and
- water supply connections with a total length 12.7 km

Further research concerning the development of the models and methods will be proceeded on the basis of experience gained from the mentioned pilot application.

REFERENCES

Bogdan L., Karczmarska D., Studziński J.2005 "Computerization of waterworks in Poland – current state and perspectives". In: Applications of Informatics in Environment Engineering and Medicine (Studziński J., Drelichowski L., Hryniewicz O., Eds.) PAS SRI, Series Systems Research, Vol. 42, Warsaw 2005, pp. 157-169.

Straubel R., Holznagel B. 1999 "Mehrkriteriale Optimierung fuer Palnung und Steuerung von Trink- und Abwasser-Verbundsystemen". In: Wasser•Abwasser, 140, No. 3, 1999, pp. 191-196.

Rossman, L.A. and Boulos, P.F. 1996. "Numerical methods for modeling water quality in distribution systems: A comparison", J. Water Resour. Plng. And Mgmt, Vol. 122, No. 2, 137-146.

Todini, E., Pilati, S. 1987. "A gradient method for the analysis of pipe networks" In: International Conference on Computer Applications for Water Supply and Distribution, Leicester Polytechnic, UK, September 8-10.

Waterworth G. 2002 "Efficiency modelling in the cost reduction of water pump maintenance". In: Proceedings of QRM'2002, Ed. G. J. McNulty, PEP London, pp. 141-144.

BIOGRAPHIES

MARCIN STACHURA was born in Szczecin, Poland. He studied at Maritime Academy of Szczecin, in 2004 he received a B. Sc. degree in the field of studies in the field of Mechanics and Machine Construction. Next he moved to Warsaw, and studied at Warsaw University of Technology. In 2007 he received M. Sc. degree in the field of Automatic Control and Robotics. After studies he started to work at the

University. He received Ph. D. degree in the field of Automatic Control and Robotics in 2011.

BARTŁOMIEJ FAJDEK was born in Warsaw, Poland. He studied at Warsaw University of Technology, in 2007 he received a M. Sc. Degree in the field of Automatic Control and Robotics. After studies he started to work at the Warsaw University of Technology.

JAN STUDZIŃSKI was born in Warsaw, Poland. He studied at Warsaw University of Technology, in 1989 he received a Ph. D. degree in the field of Automatic Control and Robotics, in 2005 a D. Sc. degree. Currently he works at the Institute of System Research of the Polish Academy of Sciences.

EVOLUTION MODELING ALGORITHM OF BIOCHEMICAL NETWORKS

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

Biochemical networks, evolution modeling, mutation operators, feedback.

ABSTRACT

Biological systems are extremely complex, nonlinear and self-organized, self-regulated systems that can adjust to the environmental changes due to underlying genetic mechanisms. The key task of systems biology is to understand cellular processes, functions and/or properties of biological system better. It can be done by means of the structure, behaviour and dynamic exploration of biochemical networks, such as genetic regulatory, metabolic, protein interaction and signalling networks. In this paper we focused on the modeling of biochemical networks evolution that occur as a result of genetic alterations and offer coterie an evolution modeling algorithm of biochemical network structure.

1 INTRODUCTION

Biochemical networks allow representation of many biological processes such as metabolism, gene regulation, signal transduction [11] and describing of intracellular molecular interactions of biological system by using available experimental data [16]. To represent biochemical networks directed and undirected graphs are used [10-13] that consist of nodes and edges. Nodes represent genes, proteins, chemical compounds or small molecules and links (edges) represent various types of interactions or associations between pair of nodes, e.g. metabolic events, protein/protein-nucleotide interactions, regulatory relationships or signalling pathways [14, 12]. A variety of sequence databases have been compiled to catalog information [3]. This available information about biological entities and their interactions enable us to consider different organisms as biological systems which are controlled by the genetic information [24].

There is a growing interest in understanding the principles of biochemical network evolution and many network growth models have been proposed to investigate this issue [17]: such as random growing models, duplication-mutation models, random static network models, aging vertex network models, small-world network models, random Boolean network models [26]. All this called network models are related to the scale-free networks, except small-world network model and tie basic algorithms of network growth to a real biological basis, naturally reproducing the scale-free [26] network architecture [19]. Above mentioned models for a network growth basis take only two types of mutations into account – gene duplication (whole genome duplication, locally confined gene duplication, retrotransposition) and gene loss [20], which are the most important drivers of network evolution. Gene duplication implies the addition of a network node and also the addition of links. With the loss of a gene not only the node but also all associated links are lost. But there are many other types of genetic mutations that can influence the process of network evolution. They do not modify a gene as whole, but modify a gene or its regulation in a way that results in link addition or loss. These genetic changes can be point-mutations, insertions or deletions, or mutations that affect the regulation of the gene [20]. As Yamada with colleague [20] note by investigating metabolic and protein interaction networks the evolution of network nodes and links is coupled to the genetic material of a cell, but the links can change over time even if nodes are unaffected, the rewiring of links can occurs without gene duplication and link changes might occur more frequently than node changes.
Biochemical network evolution can be realized due underlying genetic mechanisms. We did not found any software tool that allows to link up (to unit different level data) the genome data (gene sequences) with biochemical network components and exploration of their evolution.

In this research paper, we focused on the modeling of biochemical networks evolution that occur as a result of genetic alterations based on set of assumptions and their exploration.

2 DNA MUTATIONS

All information about biological system about how, when and where to produce each kind of protein [4], about course of processes and their regulation is carried in the genetic material, called Deoxyribonucleic acid (DNA). The genetic information carried by DNA resides in its sequence, the linear order of nucleotides and is divided into discrete functional units - genes, which typically are 5000 to 100000 [4] nucleotides long. Most bacteria have a few thousand genes (Bacterium E.Coli have 3 200 genes [27]); humans, about 25000 [27]. The genes that carry instructions for making proteins commonly contain two parts: a coding region that specifies the amino acid sequence of a protein and a regulatory region that controls when and in which cells the protein is made.

Mutations are alterations in the genetic material of cells. When the cell divides, any changes due to mutations of the original DNA sequence during the DNA replication process, will be copied and passed on to the next generation of cells. In case of a single-celled organism the mutations are passed to the next generation of organism [4]. Mutations can affect genes that encode proteins, genes encoding non-coding RNA, regulatory sequences and recognition sites. Mutations within the gene that encode proteins, can defect them producing mutant proteins and alter the cell's function. As note Clark and Pazdernik [8], *somatic mutations* that result in unregulated cell growth are responsible for the emergence of cancer, but other merely result in particular cell lines or organs. Mutations that occur within those genes whose products are tRNA, rRNA, or other non-translated RNA molecules may have drastic effects on ribosome function, splicing, or other vital processes. Furthermore, mutations may fall within promoter sequences or other regulatory sites that do not actually encode any gene product. Nonetheless, such regulatory sites are important for the gene expression and altering them may have major effects [8].

There are many types of mutations that can be neutral, useful and harmful by a factor of influence consequences, dominant and recessive by a factor of manifestation. The mutations can be divided in three main classes (see Fig.1.): 1) genome mutations, 2) chromosomal mutations; 3) genes mutations. Some affect a single base or nucleotide while others affect large segments of DNA.

More than one codon encodes a given amino acid. These redundant codons usually differ at the third position. For example, the amino acid serine is encoded by UCU, UCC, UCA, and/or UCG. This redundancy is a key to accommodating mutations that occur as DNA is replicated and new cells are produced. By allowing some of the random changes in DNA to have no effect on the ultimate protein sequence, a sort of genetic safety net is created. [29]. When the codon for an amino acid is mutated to give a stop codon (UAA, UAG or UGA), a *nonsense mutation* occur. When the ribosome translates the mRNA it comes to mutant codon and stops. The protein is made only partly and usually cannot fold properly. Such misfolded proteins are detected and degraded by the cell. Nonsense mutations are often lethal if they affect important proteins [8].

The effect of deletions or insertions depends greatly on how many bases are removed or inserted. Deletions may remove critical segments of DNA (part of a gene, an entire gene, several genes) or largely functionless regions of DNA. Point deletions or insertions may have major effects due to disruption of the reading frame if they fall within the coding region of a gene. Deletion for the DNA sequence for a whole gene means that no mRNA and no protein will be made. If the protein is essential, then the deletion will be lethal in a haploid organism [8]. As argue Clark and Pazdernik [8], point deletions and insertions may have major effects due to disruption of the reading frame if they fall within the coding region of a gene. If a single base or two bases are inserted or removed, the reading frame for all codons following the insertion or deletion will be changed. The result will be a completely garbled protein sequence. Such frameshift mutations usually completely destroy the function of a protein, unless they occur extremely close to the far end. Insertion or deletion of three bases ads or removes a whole codon and the reading frame is retained. If the deleted or inserted amino acid is in a relatively less vital region of the protein, a functional protein may be made. Deleting or inserting more than three bases will give a similar result as long as the number is a multiple of three [8].

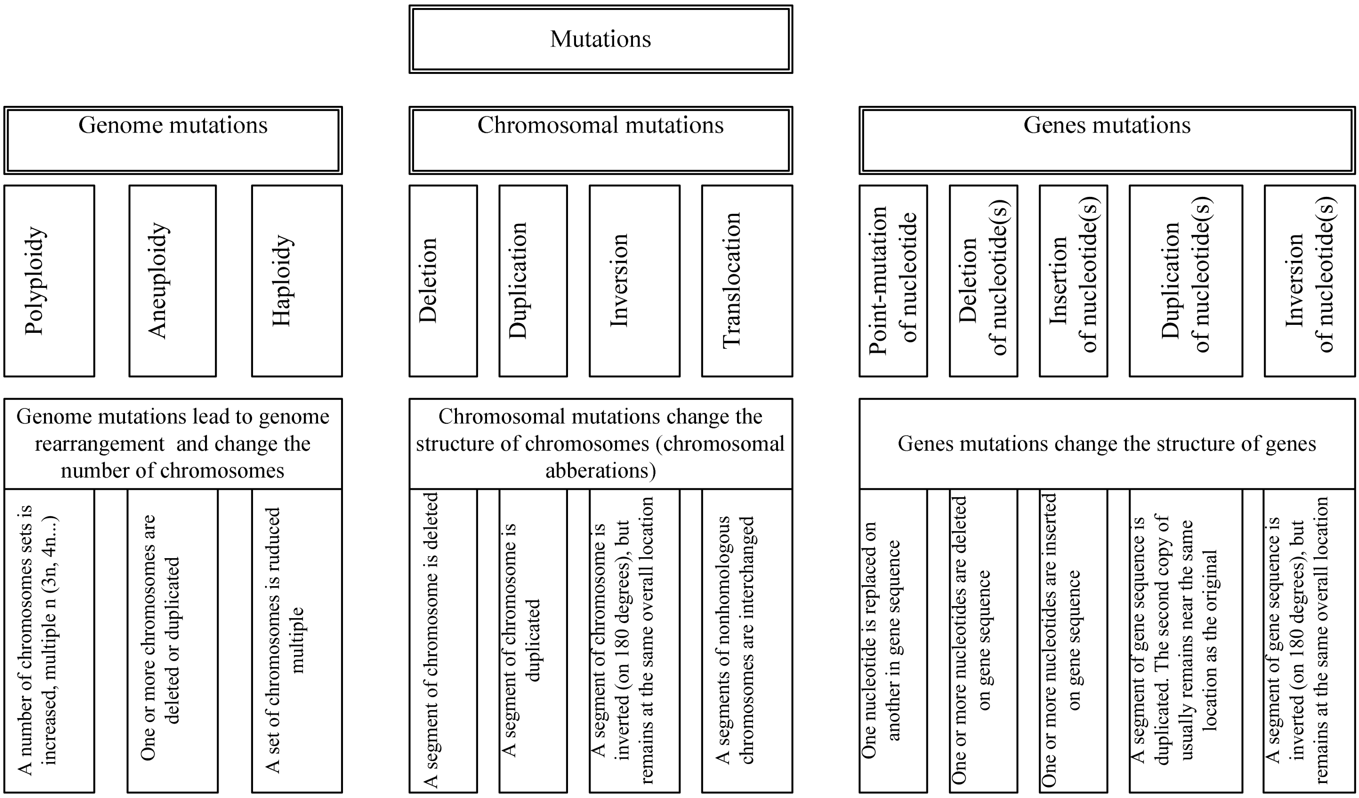


Fig. 1.:Types of mutations

The rate at which mutation occur is generally very low (approximately 2.5×10^{-8} mutations per gene per generation in humans [26], but eukaryotes and bacteria $3\cdot10^{-3}$ [6]) but can increased by environmental influences, such as chemicals, radiation [4, 8], transportable elements [1], extremely temperature [8], viruses and other factors.

3 EVOLUTIONARY MODELING OF BIOCHEMICAL NETWORKS

In several years different abstract models have been developed to investigate network structure, dynamics and evolution. Changes that were made to networks are constrained to the addition and deletion of nodes, and the addition, deletion and rewiring of links. These network level changes are biologically grounded and correspond to such types of chromosomal mutations as duplications and deletions of genes, and whole genome duplication such as polyploidy.

One of such known models is Artificial genome [2] that was developed for Gene Regulatory Networks (GRNs) modeling. This model use network level description of the genome. By modeling an artificial genome at the level of nucleotide sequences and mapping it to a regulatory network, biologically grounded mutation operators are mapped to network level mutations [1].

In AG model a genome is presented as a sequence of numbers, analogous to the string of nucleotides forming a real genome [5]. A GRN can then be extracted from this sequence using a simple set of rules based on biological mechanisms. This model allows a range of mutation operators to be applied to the genome sequence whose effect can then be observed at the level of the regulatory network. Biological mutations act upon the underlying nucleotide sequence; changes at this level do not necessarily have a one-to-one correspondence with network changes. Therefore, the mutation operators currently used to evolve artificial GRNs represent only a subset of the types of mutations that occur in nature [1].

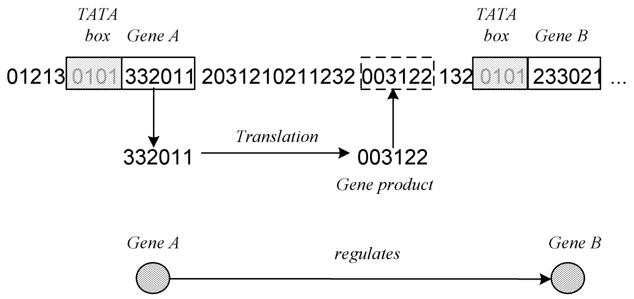


Fig. 2.: Reil's Artificial genome model [1]

To explore network growth and evolution in some cases at the theoretical level binary networks are used. Boolean network model [26] is a most popular type of network models. It consists $G(V,F)$ of a set of nodes $V=(x_1, x_2, ..., x_n)$, directed links between them and of a list of Boolean functions $F=(f_1, f_2, ..., f_n)$. In Boolean models each variable $x_i \in \{0,1\}$, $i=1,2...n$ can only attain two values (0/1 or on/off). For example, these values represent whether a gene is being expressed, or the concentration of a protein is above a certain threshold, at time t [21]. The nodes are assumed to have binary activation (on or off), at any given time step. Each directed link can be characterized as either an inhibition or activation [2, 21]. The behaviour of each network node x_i is described by Boolean function f_i based on its inputs.

This and other in practice applied models and approaches for exploring of biochemical network and their evolution have several drawbacks. First, these approaches in most cases consider the growth of biochemical networks in general leaving outside important properties. One of such properties is essentiality of given process. Not all processes in biological system are equally important. Part of them is vital without which the biological system cannot support its activity and maintain viability. Second group of processes define quality of existence but do not influence viability. Third group of processes concerns the external (visual) - features. For this reason, these processes or process property should be defined and conditionally divided. The influence of changes in genome sequence that defines and regulates the first group processes and others group's processes are not equal. Mutations even insignificant of genes sequences of the first group can be much more dangerous, make considerably changes to behaviour of biological system and have serious consequences on the system as a whole.

Biological systems are robust thanks to existence of not only feedback mechanisms, but also alternative mechanisms [28]. Systems have alternative feedback or control loops that perform to execute process in case of loss or blocking of the main mechanism. We assume that the vitally important processes are conservative. They pass in all biological systems and have very small number of alternatives or do not have them at all. We assume that conservative processes should be described with very similar gene sequences in very large number of biological systems and allow small differences in genetic material.

Second drawback of applied approaches is lack of connected genome, except the Artificial genome model. Network level changes are only theoretically explained by the existence of genome level mutations but are not introduced as result of such mutations.

Third drawback of applied approaches is remoteness from real process of evolution and application generally only of two types of mutations – genes deletion and duplication. There are many different types of mutations possible as above is described that need to be considered under different conditions. External perturbations of specific types can increase frequency of certain types of mutations. For example, X-rays tend to produce multiple mutations and often yield rearrangements of the DNA, such as deletions, inversions and translocations [8]. The influence of other types of mutations on structure and dynamic of biochemical network needs to be explored.

Fourth drawback of existing models is their limited application. These models are preferably provided for network structure evolution and less for network dynamic exploration during the evolution.

The goal of systems biology and field of our interest is to understand the mechanisms of how biochemical networks generate particular cellular functions and processes in response to environmental stresses or genetic changes [15].

To achieve this issue we offer to execute process of evolution to a given biochemical network and perform structural analysis [22, 23].

4 EVOLUTION MODELING ALGORITHM

Evolutionary changes of biochemical networks occur in their structure as result of genome level mutations. As genes define and regulate the operation of each type of biochemical network, that way to explore evolution of biochemical networks on their structure we should connect genome sequence data with higher level data of biochemical interactions and processes. It is possible via:

1) connecting genes including their sequence data with network components by attaching to network component (node or link) gene that regulates or defines it;
2) executing simulations of genes and whole genome evolution by pressure of the mutations;
3) generating (mapping) structure changes based on gene and whole genome sequence changes.

We propose an algorithm for evolutionary modeling of biochemical network structure that includes six consecutive stages (see Fig.3.).
The first stage of algorithm is definition or input structure data of initial network that includes three main substages: 1) data definition of network nodes and links, manually or loading some existing model, for example SBML model; 2) initial genome definition of an organism generating automatically the sequences of test genes or loading genome from existing database; 3) genome connection to network attaching the genes to the network components (nodes and/or links).

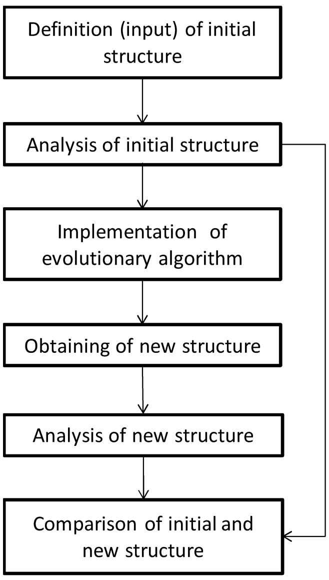


Fig. 3.: Algorithm of evolutionary modeling of biochemical network structure

Analysis of initial network structure provides calculation of structure topological parameters such as node (in, out)degree, (in, out)degree distribution, count of nodes etc., determination of network motifs such as feedback or control loops, and saving of this analysis data for further comparison.

To implement the evolutionary algorithm that provide obtaining of genomes of the next generations of organism, first some parameters need to be defined (see Fig.4.). Evolutionary algorithm will execute n times and generate genomes of n generations (see Fig.5.). To get genome of the first generation organism the initial organism genome is used, but to get genome of the t generation is used genome of the t-1generation.
To get the genes of the new t generation it is provided to generate 10 copies or alternative forms of each gene in consequence of implementation of genetic operations or mutation operators from which one alternative form is selected according to predefined criterions (see Fig.6.). After implementation of evolutionary algorithm for comparison chosen genomes of the j generations are compared with initial organism genome and estimated. Each gene is compared with its initial sequence and with all other genes sequences of initial genome to calculate the concordance coefficients. Based on concordance coefficients and essentiality level of the genes, network structure changes are generated (see Fig.7.).

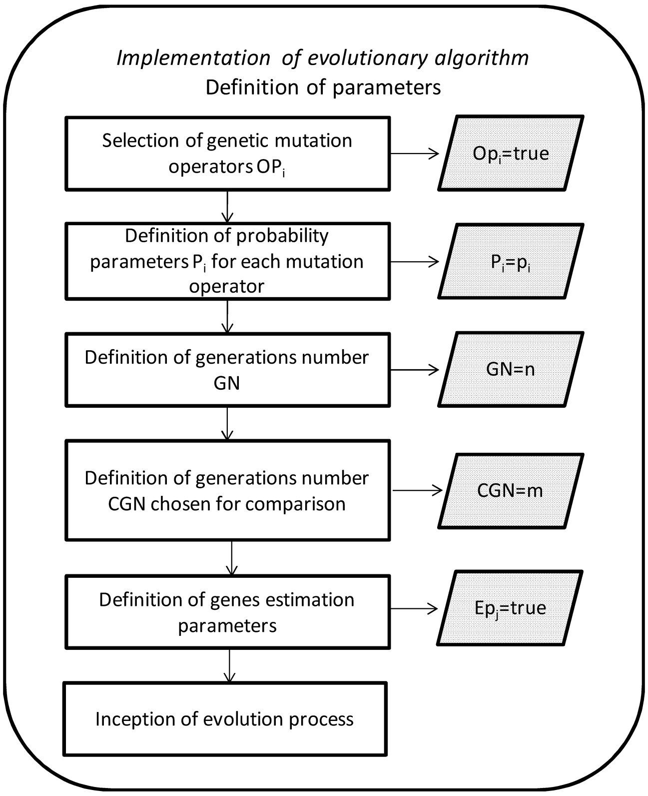


Fig. 4.: Implementation of evolutionary algorithm (1)

For example, if the link have highest essentiality level Es=1 and gene that regulates given link, concordance coefficient is Rgk=0.72 (concordance to the initial gene sequence), than link will be deleted. When network structure changes are completely introduced, the analysis of structure should be performed – calculation of topological parameters and determination of network motifs. The parameters and motifs of initial and new network structure should be compared with a goal to find similarities and differences, and to draw the conclusions about influence of chosen types of mutations.

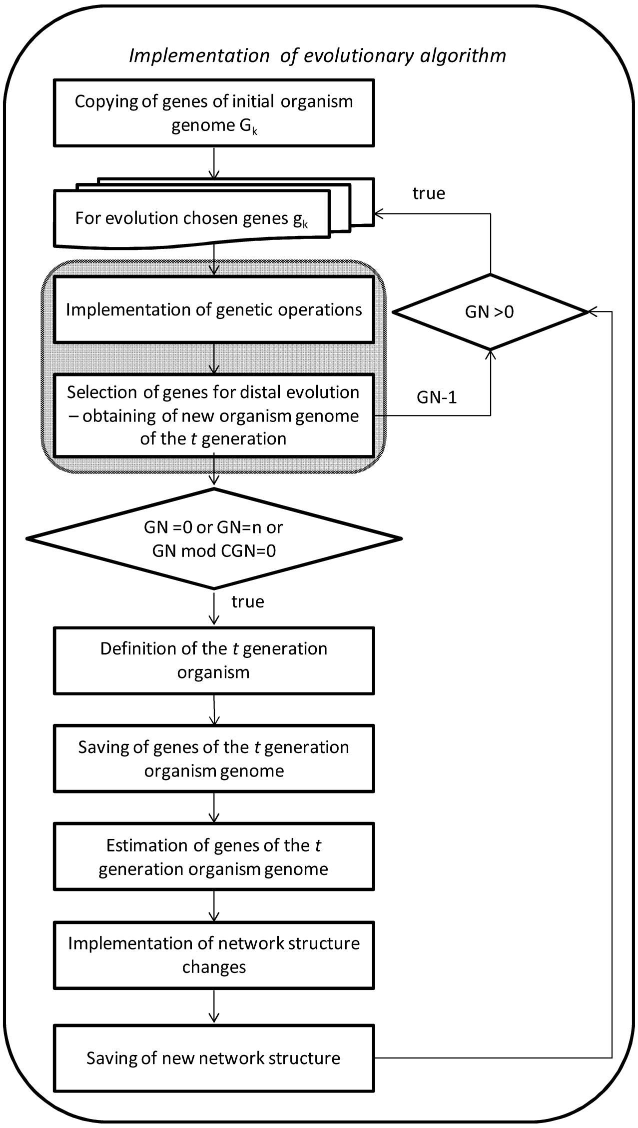


Fig. 5.: Implementation of evolutionary algorithm (2)

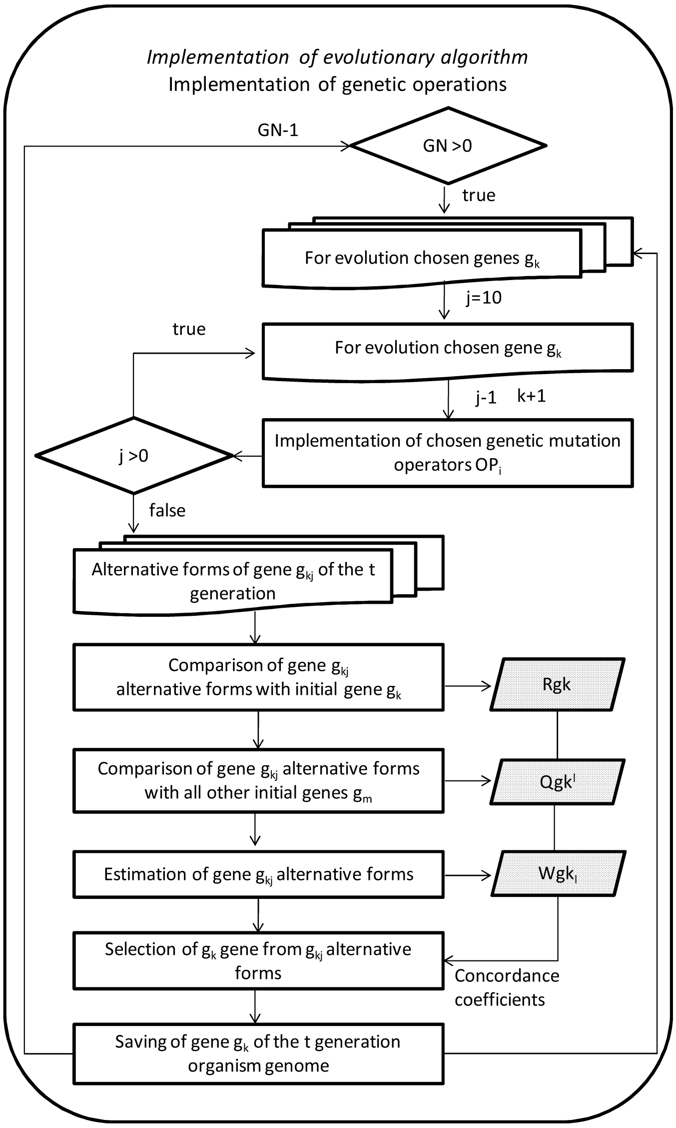


Fig. 6.: Implementation of evolutionary algorithm (3)

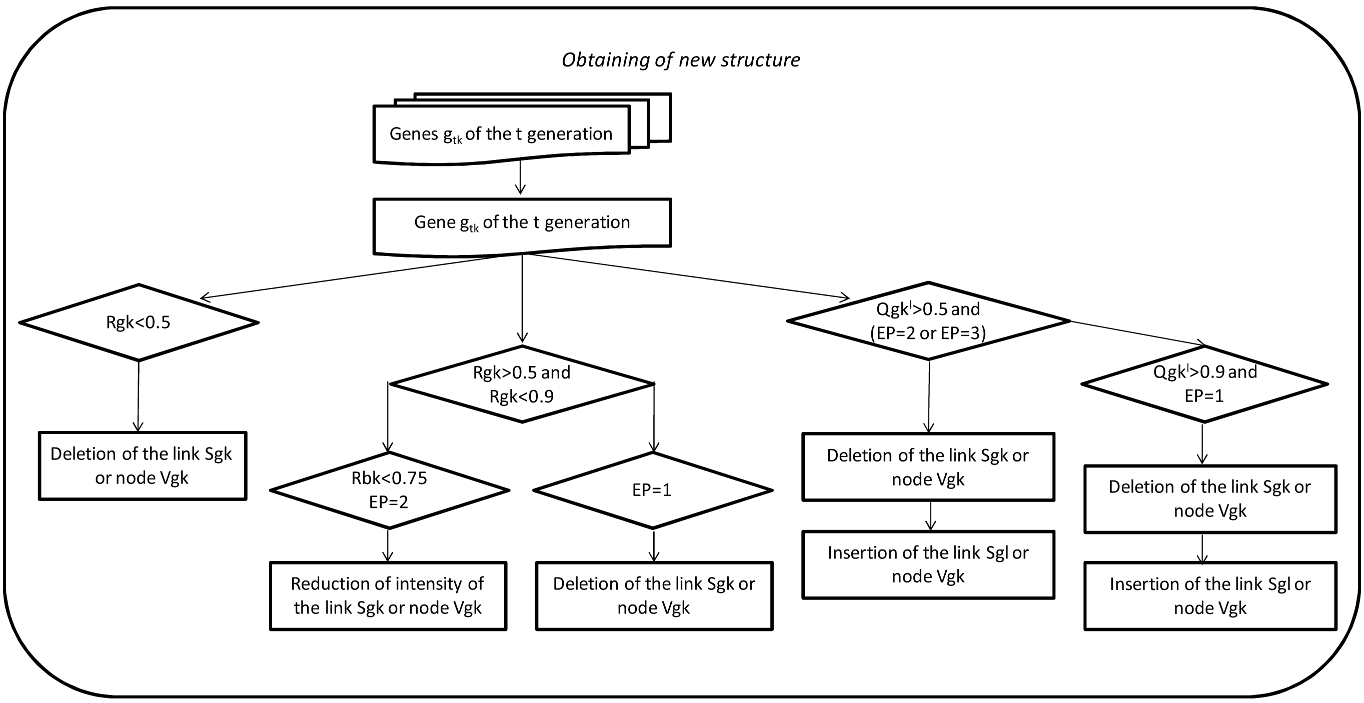


Fig. 7.: Obtaining of new structure

5 CONCLUSIONS

Several approaches of evolution modeling of biochemical network structure have been proposed by different authors. We offer an evolution modeling algorithm of biochemical network structure that is based on following assumptions:

1) Processes or links in biological system are not equally important. For this reason processes are divided on three groups by essentiality level.

2) Under influence of mutations the protein amount that gene introduced can be reduced. It can influence rate of chemical reaction to, in which protein takes part. For this reason is entered the intensity parameter for network components.

3) Genetic mutations act upon the underlying nucleotide sequence and do not always have a one-to-one correspondence with network level changes. To represent an influence of various types of mutations that occur in nature we offer to establish concordance coefficients for each gene that determine on which cases network level changes should be introduced. Biologically grounded mutation operators are mapped to the network level changes.

4) Than higher is essentiality level of process that stronger requirements are defined for gene sequence changes and concordance coefficients values. We assume that regulating genes of vital processes (1.essentiality level processes) can mutate no more than on 10%, it means, to differ from initial gene sequence, otherwise, adjustable node or link is deleted. Regulating genes of middle essentiality level processes (2.essentiality level) can mutate to 25%. In a case as regulating gene sequence changes from 25 to 50%, than intensity parameter of adjustable link or node is reduced. If the regulating gene sequence mutates more than on 50%, than adjustable link or node is deleted.

Our proposed algorithm for evolutionary modeling of biochemical network structure provides possibility to connect genome to the network components and to generate network level changes based on alterations that occur during evolution process, implementing genetic mutation operators. The proposed algorithm can be used also for dynamic exploration of biochemical network during the evolution.

6 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”.

7 REFERENCES

[1] Watson J., Geard N., Wiles J. (2004) Towards more about mutation operators in gene regulation studies, Biosystems vol.76, No 1-3, pp. 239-248

[2] Hallinan J.S., Jackway P.T. (2006) Network motifs, Feedback loops and the Dynamics of Genetic Regulatory Networks

[3] Maier R.M., Pepper I.L., Gerba C.P. (2009) Environtmental Microbiology (Sec.Eds.), Academic Press, Elsevier inc., pp.266-270

[4] Lodish H. et al. (2004) Molecular Cell Biology (5-th ed.), W.H.Freeman and Company, 973 pp.

[5] Reil T. (1999) Dynamics of gene expression in an artificial genome – implications for biological and artificial ontogeny. In: Floreano D., Nicoud J.-D., Mondana F. (Eds.), Advances in Artificial Life: 5-th European Conference (ECAL 99) Vol. 1674 of LNAI. Springer-Verlag, Berlin, pp. 457-466

[6] - Drake J.W., Charlesworth B., Charlesworth D., Crow J.F. et al.(1998) Rates of Spontaneous Mutation. Genetics, Vol.148 ,pp. 1667-1686

[7] Venter J.C. (2001) The sequence of the human genome. Science, 291, No 5507, pp. 1304-1351

[8] Clark D.P., Pazdernik N.J. (2012) Molecular Biology: Understanding the Genetic Revolution (2nd eds.). Elsevier Academic Press, 928 pp., pp.720-766

[9] Project Vibrio: Bacterial Codon Optimisation. Avialable at: http://2010.igem.org/Team:Cambridge/Bioluminescence/Bacterial_Codon_Optimisation [27.03.2012]

[10] Klamt S., Saez-Rodriguez J., Gilles E. D. (2007) Structural and functional analysis of cellular networks with CellNetAnalyzer . BMC Syst Biol, Vol. 1,

[11] Hu Z., Mellor J., Wu J., Yamada T., Holloway D., DeLisi C. (2005) VisANT: data-integrating visual framework for biological networks and modules. Nucleic Acids Research 2005, Vol.33, pp.352-357

[12] - Hu Z., David M. Ng., Yamada T., Chen C., Kawashima S., Mellor J., Linghu B., Kanehisa M., Stuart J.M., DeLisi C. (2007) VisANT 3.0: new modules for pathway visualization, editing, prediction and construction. Nucleic Acids Research, June 22, 2007

[13] - Zinovyev A., Viara E., Calzone L., Barillot E. (2008) BiNoM: a Cytoscape plugin for manipulating and analyzing biological networks. Bioninformatics applications note. Vol. 24, no.6, 2008, pp. 876-877

[14] - Suderman M., Hallett M. (2007) Tools for visually exploring biological networks. 2006. Published by Oxford University Press. August 25, 2007. Available at: http://bioinformatics.oxfordjournals.org /cgi/reprint/btm401v1.pdf [30.03.2012]

[15] - 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

[16] - Zhang A. (2009) Protein Interaction Networks: Computational analysis. Cambridge university press. 2009. pp.4, 44

[17] - Kwon Y.K., Cho K.H.: Analysis of feedback loops and robustness in network evolution based on Boolean models. BMC Bioinformatics 2007, Vol.8, pp.1-9

[18] - Alon U., Surette M.G., Barkai N., Leibler S. (1999) Robustness in bacterial chemotaxis. Nature, 1999, pp.168-171.

[19] - Albert R., Barabasi A.L. (2000) Topology of Evolving Networks: Local Events and Universality. Physical rewiev letters 2000, Vol.85, pp. 5234-5237

[20] - Yamada T., Bork P. (2009) Evolution of biomolecular networks – lessons from metabolic and protein interactions. Nature reviews, Molecular cell biology 2009, Vol.10, pp.791-803

[21] - 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

[22] – Rubina T., Stalidzans E. (2010) Software Tools for Structure Analysis of Biochemical Networks. Proceedings of the 4-th international scientific conference Applied information and Communication Technologies AICT'2010, ISBN 978-9984-48-022-0, Jelgava, Latvia, pp. 33-49

[23] – Rubina T., Stalidzans E. (2010) Topological features and parameters of biochemical network structure. ISC'2010, 8-th Industrial Simulation Conference. A publication of EUROSIS, Budapest, Hungary, pp. 228-236

[24] - 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

[25] - Nachman M. W., Crowell S. L. (2000) Estimate of the Mutation Rate per Nucleotide in Humans. Genetics 156, pp. 297-304

[26] - Rubina T., Stalidzans E. (2012) Evolution of control loops of biological systems. Proceedings of the 5-th international scientific conference Applied information and Communication Technologies AICT'2012, ISBN 978-9984-48-065-7, Jelgava, Latvia, pp. 317-324

[27] - U.S. Department of Energy Genome Program's. Biological and Environmental Research Information System (2008) Human Genome Project. Genomics 101: A primer. Avialable at: http://www.ornl.gov/hgmis/publicat/primer/ (04.04.2012)

[28] - Kitano H. (2004) Biological robustness. Nature reviews, genetics, Vol. 5, 2004, pp. 826-837

[29] - NCBI. What is a genome? Avialable at: http://www.ncbi.nlm.nih.gov/About/primer/genetics_genome.html

SOFTWARE DESIGN SIMULATION

SHARING LIMITED RESOURCES IN SOFTWARE PROCESS SIMULATIONS

Štěpán Kuchař
IT4Innovations
VŠB – Technical University of Ostrava
17. listopadu 15/2172
70833 Ostrava-Poruba, Czech Republic
E-mail: stepan.kuchar@vsb.cz

David Ježek, Jan Kožusznik, Svatopluk Štolfa
Department of Computer Science
VŠB – Technical University of Ostrava
17. listopadu 15/2172
70833 Ostrava-Poruba, Czech Republic
E-mail: david.jezek@vsb.cz, jan.kozusznik@vsb.cz, svatopluk.stolfa@vsb.cz

KEYWORDS

BPM method, software process, human-based process, process simulation, Petri nets, shared resources

ABSTRACT

This paper describes the use of the BPM Method simulations for estimating the appropriate number of resources in the software process to balance its cost and duration. Extensions to the BPM Method that are necessary to model the sharing and managing of limited resources are defined and their conversion to the Petri Nets formalism is suggested to specify the simulation semantics. Balancing formulas and ratios for process cost and duration are provided and used for a local software development company's software process simulations to find a balanced number of workers in specific roles in the process.

INTRODUCTION

Business processes represent the core of each organization's behavior (Madison 2005). They define a set of activities that have to be performed to satisfy the customers' needs and requirements, roles and relationships of the employees that are needed for actually performing these activities and objects that are consumed or produced by these activities (Šmída 2007). The software process is also a business process that is highly dependent on human creativity, competencies, experience and interaction (Dutoit et al. 2006). Human-based processes tend to be more uncertain and risky than automated processes performed by machines and properly trained employees are one of the main sources of the company's competitive advantage (Hatch and Dyer 2004).

Due to this uncertainty and the scale of the software process it is problematic to exactly predict the duration, cost and quality of the process and to balance these three indicators (Rus et al. 1999). This problem can be solved by running simulations of the process that can work with stochastic parameters and probability distributions to approximately model the uncertainty. After running many simulations their results can be statistically evaluated and can provide an approximation of the duration, cost and quality of the process.

As the quality of the process mainly depends on the capability and maturity of the process (Persse 2006) and competency of its workers (Hatch and Dyer 2004), the duration and cost greatly depend on the number of resources allocated in the process. But how many resources should be allocated to strike a balance between these two indicators? This paper tries to answer this question by running and evaluating simulation experiments for the software process of a local middle-sized software development company.

BPM METHOD

A modeling and simulation method that is able to sufficiently model human-based processes was needed to execute these experiments. For these purposes we used the BPM Method (Vondrák et al. 1999a; Vondrák et al. 1999b) that already provides simulation environment with stochastic parameters (Kuchař and Kožusznik 2010). This method defines three basic models of the process – architecture of the process, objects and resources utilized in the process and the behavior of the process. The most important one of these models for performing simulations is the behavioral model. This model is called the Coordination model and it specifies the behavior of the process as a sequence of activities. It also specifies what resources the activities demand and which artifacts they consume and produce. Alternative flow in the coordination model is enabled by multiple activity scenarios and concurrency of the activities can also be modeled. This model can also be converted to a Petri net to provide exact semantics for performing simulations (Kuchař and Kožusznik 2010).

The Coordination model is visualized by the Coordination diagram and a simple example of this diagram is shown in Figure 1.

This diagram describes a part of the Software Construction subprocess. The Designer and Developer places describe the roles of employees in the process and the Task place defines the task objects that serve as input for the Construction activity. System block can be constructed when the Task is created and the Developer resource is available. The yellow arrow shows that the Developer is responsible for executing the Construction activity. By completing this activity the state of the Task changes to implemented, new System block is created and the Developer is ready to implement another task.

The subsequent activity is Code verification that is performed by the Designer and consumes the implemented Task and created System block. This activity can end up in two ways. The first scenario signifies that the constructed code is correct and its outputs are marked by number 1. The second scenario shows that there were errors in the implementation and the process will continue by reporting and repairing the error. Outputs of the second scenario are marked by number 2.

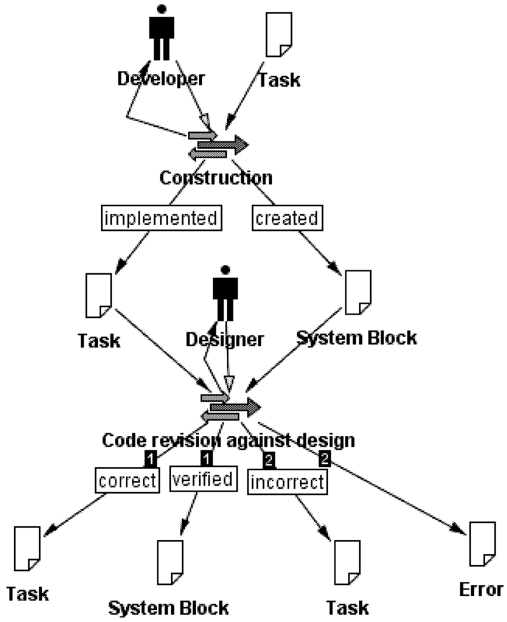


Figure 1: Coordination Diagram

PARALLELISM IN THE BPM METHOD

The previous example shows that activity scenarios are used to branch the workflow to a number of alternate branches but no specific modeling elements for parallelism are contained in the BPM Method. Parallel flows in the BPM Method can be modeled in several ways:

1. One activity can be performed concurrently many times, but each such activity instance consumes appropriate input object instances. If there were for example three Developers available and five Tasks created, then the Construction activity in Figure 1 could be performed three times concurrently. Each activity instance would consume one Developer and one Task leaving two Tasks for the Developers that finish their Tasks first.
2. Several different activities in the same process instance can be performed in parallel by structuring the model correctly. The correct structure is based on the AND-split structure of the Petri nets where multiple branches start from one activity and AND-join structure where multiple flows merge into one activity.
3. Every process instance runs concurrently with other running process instances be they of the same process or of another processes. Each process instance is independent from other instances and only shares common resources (Kuchař and Kožusznik 2010).

SHARING LIMITED RESOURCES

All of these concurrency methods have to deal with sharing limited number of resources. The first one is directly solved by standard Petri nets behavior because each resource type for one activity is contained in one input place of this activity. When this activity starts all its input resources are consumed and are not available until it finishes. The only extension in the BPM Method needed here was to enable the multiple-server semantics for each activity in the process (Boyer and Diaz 2001).

The second concurrency method can be solved directly by sharing the same input place among all activities in the parallel branches that utilize resources in this place. This will fill the more complex models with a lot of additional arcs but it does not matter for the automatic processing in the simulations.

The third concurrency method is the most problematic because it needs to share resources between multiple instances of the process. This can be implemented by introducing the pools of limited resources to the objects in the Coordination model. Objects within the same shared pool are then linked together. When one of the resources in the pool is used to perform an activity, it is taken from all linked objects. After the activity finishes, the used resource is returned to the pool and thus returns to all linked objects. To add this behavior to the BPM Method's simulation engine we had to find a way to convert the idea of these pools into the Petri nets formalism to be compatible with the rest of the simulation model. This problem was solved by adopting the notion of fusion places (Huber et al. 1991).

Fusion places are special places in Petri nets that always share all their tokens. When a token is consumed from any fusion place, all linked fusion places lose that token. When a token is added to any fusion place, all linked fusion places gain that token.

Each shared pool in the BPM Method corresponds to a set of linked fusion places that are shared between all concurrent simulation instances of the process. These fusion places are then set as both the input and output places for all activities that require shared resources from the appropriate shared pool. Figure 2 shows the Coordination diagram with shared pools and Figure 3 depicts the corresponding Petri net with fusion places.

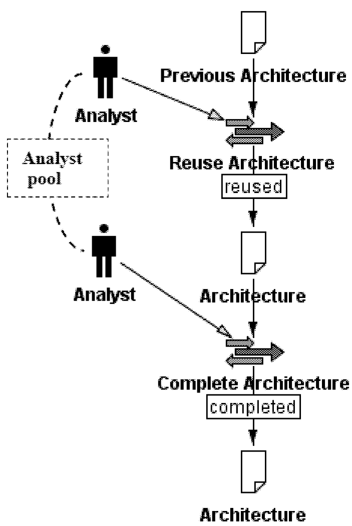


Figure 2: Shared Resource Pools

Both Analyst objects in the Coordination diagram are taken from the shared pool of analyst resources. This pool is then represented as a set of two linked fusion places in the Petri net which serve as both input and output place for the Reuse Architecture and Complete Architecture activities. Standard places for both Analyst objects are still used to ensure correct flow of the process if they were activated by previous activities.

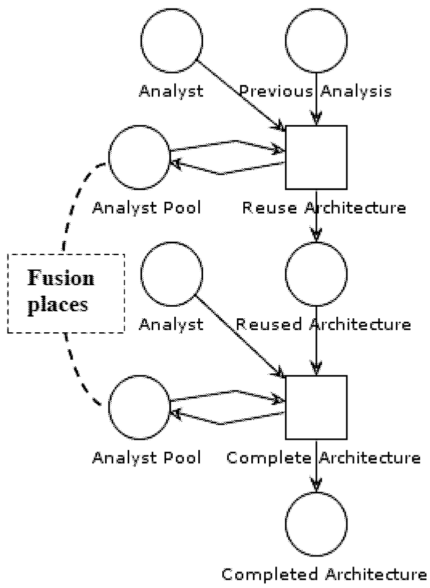


Figure 3: Petri Net with Fusion Places

Fusion places can also be used to solve the second concurrency method instead of the pure Petri net solution mentioned earlier. This usage is preferred because it simplifies and clarifies the process diagram.

CHAINED EXECUTION

Notation of the Coordination diagram can also describe chained execution (Aalst 1998). Chained execution defines that the same resource is used to perform several subsequent activities in one process instance. In this way experience acquired by the resource during the first activity can be easily reused in subsequent activities (e.g. when all activities work with the same set of data). The BPM Method models the chained execution by using one object as an output place for one activity and at the same time as an input place for subsequent activity. When this happens the same shared resource will be used for both activities.

Chained execution can be easily converted to a Petri net by adopting some colored properties (see Colored Petri nets, e.g. (Jensen 1998)), coloring the chained shared resource and then matching the color when choosing the resources for all activities in the chain.

RESOURCE UTILIZATION AND UNAVAILABILITY

The last extension added to the BPM Method is determining the utilization of each shared resource in the process. This utilization is measured by simply counting up the time when the resource is performing any activity.

Utilization is an interesting result of the simulation but it is not very useful in optimizing the performance of the process. When optimizing the number of resources in the process we are not interested in answering how long one resource was doing something in the process, but rather how long did the process have to wait for the resource when it was needed to perform another activity. One resource cannot perform two activities at the same time but activities and processes run concurrently and they very often need the same resource to be able to continue their run (e.g. one developer is needed to implement a new feature in one system and at the same time needs to repair a fault in another system). When this happens the resource has to somehow perform these tasks sequentially:

- finish the first task and then start the second one, or
- pause the first task and return to it after finishing the second one, or
- switch back and forth between these tasks.

In either way one task will have to wait for the completion of the other (or partial completion in the case of the third option). It is therefore important to be able to simulate and measure these waiting times. The BPM Method can easily model this, but is able to model only the first sequencing option. Whenever an activity is enabled but the resource is not available, the BPM Method counts and notes the time needed for the resource to become available to perform the activity. Total waiting time for one resource is then computed by adding up these noted times for this appropriate resource.

CASE STUDY

We have cooperated with a middle-sized software development company from Czech Republic to model and improve their software development processes. Having made some improvements in the process (see (Kožusznik et al. 2011)) a question of how many employees should be allocated to an instance of this process arose. By extending the BPM Method's modeling and simulation software tool called BPStudio with the mechanisms described in previous chapters it was possible to run simulation experiments with varying number of employees and analyze the results to find the best configuration.

The first step was to identify the roles in the process that would influence the total duration of process execution. This can be done by running a simulation with small number of resources in different roles and then running a simulation with more resources. By watching waiting times of these resources bottlenecks in the process can be identified. When the waiting time is high and is changing with the number of resources then the process could benefit from adding resources to this role. Utilization and waiting times for one of these simulations performed in this case study is shown in Figure 4.

This simulation showed that only three out of five roles – Developer, Designer and Tester – have significant waiting times so further simulations focused on finding optimal number of employees in these roles. It is also important to note, that all workers of one role have same skills and are therefore interchangeable. This can be also seen on the approximately equal utilization and waiting times of the developers in Figure 4.

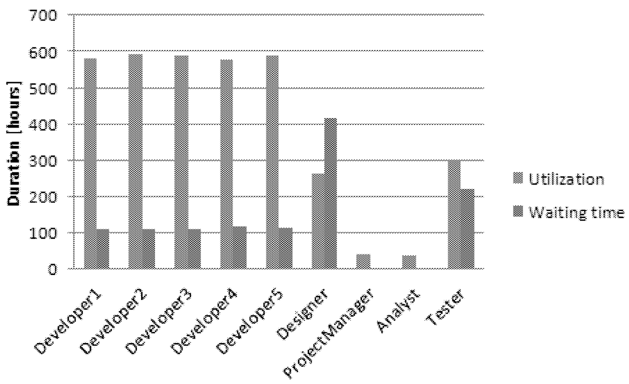


Figure 4: Utilization and Waiting Times of Resources

The second step was to run simulations with all resource configurations that were feasible in the process. Developers are the most utilized role in the process and by adding more workers their waiting time decreases slowly and so the range of possible developers was set to 2-20. Figure 4 also shows that the lone designer is overburdened with work because designers also play a role of team leaders in this process. Based on this information the range of feasible designers was set to 1-5. The tester has also a lot of unavailable time on his hands but not so much as the designer, so the number of testers was set to 1-4.

After establishing the feasible ranges of workers in each role, 200 simulations for every combination of these resource numbers were performed to get the adequate statistical image of the simulations considering their stochastic properties and risks in the process. These combinations were divided to 20 groups with the same number of designers and testers for better understanding and more transparent analysis (i.e. one group was for 1 designer, 1 tester and 2-20 developers; another group was for 1 designer, 2 testers and 2-20 developers, etc.). Finally a statistical analysis was done for each of these groups concerning the duration and total cost of the process. Figure 5 shows the chart with duration results of the group with 2 designers and 2 testers.

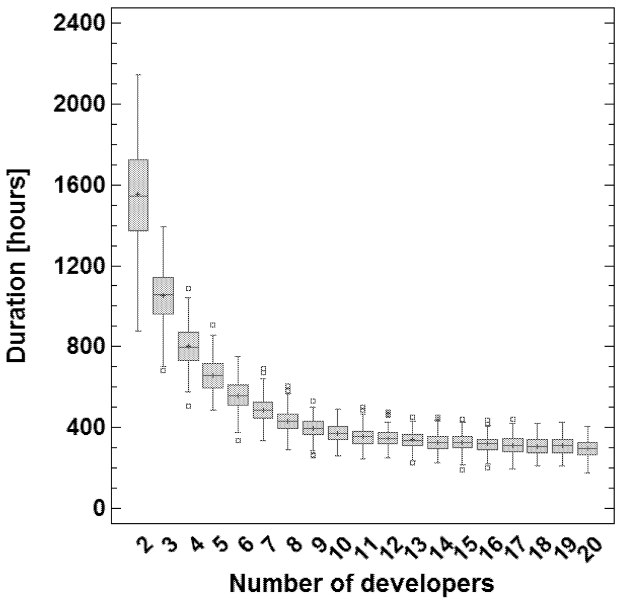


Figure 5: Durations for 2 Designers and 1 Tester Group

The best duration average in this group has the configuration with 20 developers because the waiting time of these developers still has not decreased to 0. But statistical comparison of this configuration with the other configurations shows that it is not significantly better than the configurations with 16, 17, 18 and 19 developers (at 95% confidence level). It is therefore better to use the lower number of developers, because such unallocated developers can be sent to work on another project. The chart also shows that starting around the 10 developers mark the slope of the decrease is very small and the process does not benefit much from adding more developers.

This trend is also confirmed by the total process cost chart of the same group in Figure 6. The chart shows that the cost of allocating additional developers to the process is higher than the decrease in the duration for 10 and more developers. The best configuration optimizing the total process cost in this group is 9 developers with 8 and 10 being statistically comparable. This means that 8 developers should be used to free the ninth developer for other projects.

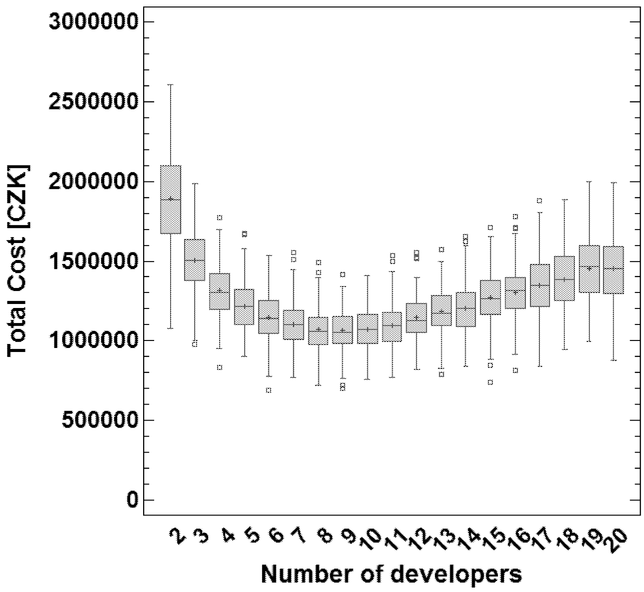


Figure 6: Total Costs for 2 Designers and 1 Tester Group

After analyzing all groups in this manner the best local configurations can be compared with each other in the same way to find the best global configuration that minimizes either the duration or the process cost. The best global configuration minimizing the process duration is 3 designers, 3 testers and 19 developers with 289.08 hours and 1 568 254 CZK. The best global configuration minimizing the process cost is 1 designer, 1 tester and 8 developers with 637.64 hours and 1 376 698 CZK.

These configurations optimized only one indicator without worrying about the other. But how to find a configuration that balances both? The first step is to normalize average values of both indicators for each configuration in each group by using the following formula:

$$navg(c_i) = \frac{avg(c_i) - min(avg(c_1), \ldots, avg(c_n))}{max(avg(c_1), \ldots, avg(c_n)) - min(avg(c_1), \ldots, avg(c_n))}$$

where n is the number of configurations in actually examined group and c_j for $j \in \{1, \ldots, i, \ldots, n\}$ describes actually examined indicator's values for all simulations of the *j*-th configuration in actually examined group.

The balancing value of each configuration is then:

$$balance(c_i) = \alpha * navg_{duration}(c_i) + \beta * navg_{cost}(c_i)$$

where α∈<0,1> is the duration indicator weight, β∈<0,1> is the cost indicator weight and α + β = 1.

The best balanced local configuration of each group is the one that minimizes this balancing value. These best balanced local configurations can then be collected to one common group and among them the best balanced global configuration can be identified by the same procedure.

The final problem is to correctly set the α and β weights so that they really balance the duration and cost. It is clear that if α=1 and β=0 then only duration is taken into consideration and the configuration with the lowest duration is declared the best. On the other hand if α=0 and β=1 then only process cost is considered in choosing the best configuration. The first idea that comes to mind is setting α=0.5 and β=0.5 so that both are of the same weight but this might not balance these indicators correctly. The balancing value with these weights is very sensitive to duration and cost indicators with different variances. For example if the duration varied from 100 to 900 hours and the process cost only ranged from 1 000 000 to 1 000 010 CZK then the loss of 400 hours would be balanced by the loss of 5 CZK only. The solution to this problem is to set the α and β weights in accordance with the range of the duration and cost indicators. This has to be done for each configuration group separately because ranges can be different in each group.

To set the weights correctly a range ratio has to be computed for both indicators that describe how much the maximum average value is greater than the minimum average value by the following formula:

$$ratio = \frac{max(avg(c_1), \ldots, avg(c_n)) - min(avg(c_1), \ldots, avg(c_n))}{min(avg(c_1), \ldots, avg(c_n))}$$

This ratio is then directly used to evaluate the α and β weights using the following formulas:

$$\alpha = \frac{ratio_{duration}}{ratio_{duration} + ratio_{cost}} \qquad \beta = \frac{ratio_{cost}}{ratio_{duration} + ratio_{cost}}$$

Using these weights the best balanced global configuration came out to be the same as the best duration global configuration – 3 designers, 3 testers, 19 developers. Weights used in the global comparison were α=0.82 and β=0.18. For example by setting α=0.5 and β=0.5, the best global configuration would be 2 designers, 2 testers and 13 developers because it prefers lower process cost at the expense of longer process duration.

CONCLUSION AND FUTURE WORK

This paper described a method that uses simulations for estimating the number of resources in different process roles that equally balance the duration and cost of the final process instance. Duration or cost of the process can also be prioritized when computing the balancing value by changing the ratio of balancing weights. This could be used for finding resource configurations that support the need for processes with the shortest possible duration (e.g. for taking advantage of high market demand) or lowest possible cost (e.g. for small start-up projects).

Real processes can also be limited by specific budget or specific time. Such limitations could be easily implemented into the indicated analysis method by filtering out the configurations that exceed these limits.

All simulations in this case study assumed that all workers of one role have same skills and therefore have similar levels of utilization, waiting times and performance. Real workers have different skills that change the way they are allocated. Our future work will be focused on introducing these differences to process simulations.

ACKNOWLEDGEMENTS

This research has been supported by the internal grant agency of VSB-TU of Ostrava - SP2012/26 „An utilization of artificial intelligence in knowledge mining from software processes“ and supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

REFERENCES

Aalst W.M.P. van der. 1998. “The Application of Petri nets to Workflow Management”. *The Journal of Circuits, Systems and Computers 8* (1), 21-66.

Boyer M. and Diaz M. 2001. „Multiple enabledness of transitions in Petri nets with time“. In *9th International Workshop on Petri Nets and Performance Models*. IEEE, 219–228.

Dutoit A.H.; McCall R.; and Mistrik I. 2006. *Rationale Management in Software Engineering*. Springer.

Hatch N.W. and Dyer J. H. 2004. “Human capital and learning as a source of sustainable competitive advantage”. *Strategic Management Journal* 25 (12), 1155-1178.

Huber P.; Jensen K.; and Shapiro R. 1991. “Hierarchies in coloured petri nets”. *Advances in Petri Nets*, vol 483. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 313-341.

Jensen K. 1998. “Coloured Petri nets: basic concepts, analysis methods and practical use”. *Monographs in Theoretical Computer Science*, 2nd corrected printing edn. Springer-Verlag.

Kuchař Š. and Kožusznik J. 2010. “BPM Method Extension for Automatic Process Simulation”. In *8th Industrial Simulation Conference 2010*, Budapest, Hungary, 80-85.

Kožusznik J.; Štolfa S.; Ježek D.; and Kuchař Š. 2011. “Petri Net Based Simulation for SPI”. In *17th European Concurrent Engineering Conference/7th Future Business Technology Conference 2011*, London, United Kingdom, 33-39.

Madison D. 2005. *Process Mapping, Process Improvement and Process Management*. Paton Press.

Persse J.R. 2006. *Process Improvement Essentials: CMMI, Six SIGMA, and ISO 9001*. O’Reilly Media.

Rus I.; Collofello J.; and Lakey P. 1999. “Software process simulation for reliability management”. *Journal of Systems and Software 46* (2-3), 173-182.

Šmída F. 2007. *Zavádění a rozvoj procesního řízení ve firmě*. Grada Publishing, a.s.

Vondrák I.; Szturc R.; and Kružel M. 1999. “Company Driven by Process Models”. In *European Concurrent Engineering Conference Proceedings ECEC ’99*, Erlangen-Nuremberg, Germany, 188-193.

Vondrák I.; Szturc R.; and Kružel M. 1999. “BPM – OO Method for Business Process Modeling”. In *ISM ’99 Proceedings*, CSSS, Rožnov pod Radhoštěm, 155-163 .

Generic Source Code Migration Using Decompilation

Lukáš Ďurfina
Jakub Křoustek
Petr Zemek

Faculty of Information Technology, IT4Innovations Centre of Excellence,
Brno University of Technology, Božetěchova 1/2, 612 66 Brno, Czech Republic
email: {idurfina, ikroustek, izemek}@fit.vutbr.cz

KEYWORDS

source code migration, analysis, transformation, retargetability, reverse engineering, decompilation, Lissom, LLVM, LLVM IR

ABSTRACT

Many core parts of software applications have been written several decades ago, and they are still essential in several existing projects. However, maintaining these parts is difficult and expensive. The source language itself may be either ancient and ineffective, or its toolchain does not support new platforms and features. Source code migration to a newer language is often the only solution. In the past, several single-purpose converters have been created. However, such solutions are more or less limited to a particular language and computational platform. In this paper, we present a generic solution of this problem. The proposed solution is independent of any particular source or target language, or on the target platform. We utilize an existing retargetable decompiler and the LLVM Compiler System for the conversion process. At present, the Python language is used as the target language, while other languages are under development. As can be seen from the experimental results, we are able to migrate source code from many particular languages (e.g. C/C++, D, Fortran) into a functional equivalent and highly readable code in Python.

INTRODUCTION

Long-lasting software projects have to deal with problems related to their implementation language or its compiler toolchain. Their original source code is typically written in an ancient language or some of its dialects. Maintaining such code represents a very difficult task because debugging and testing tools may not be available or their functionality may be limited (see Lerew 1999, Dickens 2002). Enhancing or porting to newer platforms is usually impossible because the original language does not support many nowadays common features (e.g. modular design, object-oriented development, multi-threading). The original toolchain is usually not developed anymore and its support is over.

One approach on how to preserve application's development is a migration of its sources. A common solution is a manual rewriting of the complete source code into a modern programming language. However, such a process is vastly time consuming and error prone. Moreover, proper debugging and testing takes usually more time than the physical rewriting itself.

In the past, several source code converters have been developed. However, most of them were limited to a particular input and output programming language, and often to a particular target architecture as well (see Feldman (1990), Kontogiannis et al. (1998), Dickens (2002)). There is no appropriate generic solution which allows source code migration between multiple programming languages.

In this paper, we present such a solution. We use an existing retargetable decompiler devolved within the Lissom Project (2012) for this purpose. The decompiler is originally intended for reverse translating binary applications into a high-level-language (HLL) source code. More precisely, we use only its back-end for source code migration because the front-end part, which transforms machine code into an internal code representation, is not needed for this high-level transformation. Binary code migration is possible as well, but it is beyond the scope of this paper.

The LLVM Compiler System (2012) is used for parsing the input source code and transforming it into the *LLVM bitcode file format*, which is further translated by our own converter. We support migration from a broad range of languages (e.g. Objective-C/C++, D, Fortran) into the Python Programming Language (2012). Support of other output languages is under development (e.g. the C language).

Currently, we are able to automatically generate a highly readable HLL code while still preserving functional equivalence of the input source code.

The paper is organized as follows. The next section describes existing solutions of source code migration. Then, we briefly discuss the retargetable decompiler developed within the Lissom research project. Afterwards, our source code migration solution is presented. Experimental results are given in the subsequent section. A discussion of future research closes the paper.

RELATED WORK

We can find several projects focused on migrating source code. Each of these projects exploits its own migration techniques between two particular languages. However, only a few of them allow migration from more than one programming language. Most tools described below use techniques based on parsing the original source code and its consequent transformation and compilation into the target language.

A typical example of a migration from an insufficient language to another one is discussed in Kontogiannis et al. (1998). Other tools for migrating Fortran applications to C (`f2c`) and Java are proposed in Feldman (1990) and Dickens (2002), respectively. A conversion between all of these languages is also presented in Dickens (2002). In Zhong et al. (2010), a system for translating Java programs into C# is proposed.

Yang (2003) introduced a tool more similar to reverse engineering: a translator from low-level assembly code into C or Cobol. Finally, Mills and Peterson (1998) introduced a converter between ADA 95 and VHDL, which can be used for more effective hardware/software co-design and partitioning.

An alternative approach, which we follow in the rest of our paper, is converting source files into a unified intermediate representation, which is then transformed into the target language. The LLVM Compiler System (2012) provides such a representation: the LLVM assembly language, called *LLVM IR*. First, a source file may be converted by a suitable compiler, such as `llvm-gcc` (see The LLVM Compiler System (2012)), whose front-end supports the source language. Indeed, instead of emitting assembly code for a specific processor, it emits LLVM IR. After that, an LLVM back-end may be created for each target language.

Even though LLVM primarily aims for converting LLVM IR into assembly code, back-ends for emitting high-level code can be created. Currently, LLVM contains a C back-end for transforming LLVM IR into C. However, the resulting code is rather a direct translation of the input LLVM IR so it lacks high-level constructs. This makes the result a lot less readable and maintainable. As explained later in the paper, we aim to reduce this disadvantage.

As another example of this approach, we can mention Emscripten (2012). It is a compiler able to transform LLVM IR into JavaScript. It is used for transforming C/C++ code for running on the web. A similar project is llvm-js-backend (2012), which represents a general LLVM back-end for producing JavaScript code. Finally, let us point out that there are also ways of generating LLVM IR from self-created languages. For example, the parser generator ANTLR (2012) provides such a way (see ANTLR's page on LLVM (2012)).

RETARGETABLE DECOMPILER AND THE LISSOM PROJECT

In this section, we briefly describe the concept of an automatically generated retargetable decompiler, which is developed within the Lissom Project (2012), and which is used as a core of our source code migration solution. Our solution is build on the top of The LLVM Compiler System (2012). The LLVM assembly language, LLVM IR, is used as an internal representation of decompiled applications in particular decompilation phases. See Ďurfina et al. (2011) for a detailed description.

The decompiler consists of three basic parts—a *front-end*, a *middle-end*, and a *back-end*, see Figure 1. The front-end is the only platform-specific part of the decompiler because it is automatically generated based on the target architecture model. This model is realized in the architecture description language ISAC (Masařík 2008), which is also developed within this project.

At first, the decompiled application is converted from a platform-specific executable format into an internal file format. Afterwards, it is processed by the front-end. The front-end translates the application into an LLVM IR instruction sequence, which characterizes its behavior in a platform-independent way.

After that, the LLVM IR program representation is optimized in the middle-end using many built-in optimizations available in LLVM and our own passes (e.g. optimizations of loops, constant propagation, control-flow graph simplifications) and emitted as the target HLL in the back-end. These two parts are discussed below in a greater detail.

In comparison with existing decompilers, our solution is not limited to any particular architecture. It should be noted that at present, there is no other competitive method of automatically generated retargetable decompiler.

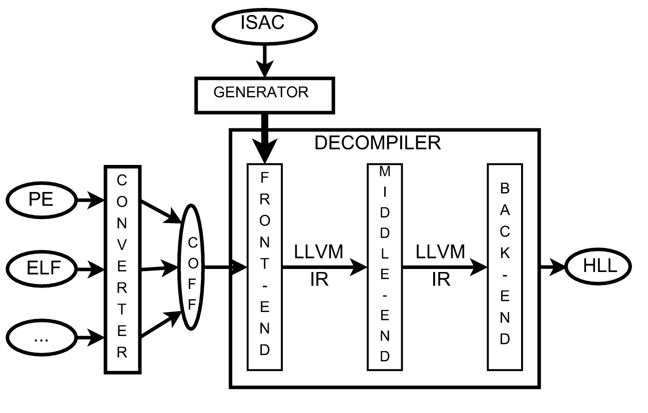


Figure 1: The concept of our retargetable decompiler.

THE MIGRATION PROCESS

This section describes our solution of code migration. It is divided into four subsections. First, we translate the given source file, written in a source language, into LLVM IR. Then, a pre-processing phase optimizes this IR. After that, we convert the optimized IR into the resulting HLL, which is then further neatened during a post-processing phase.

Translation to LLVM IR

As an input, we have a source code in a source language, which can be C, Fortran, etc. We use `llvm-gcc` with an appropriate front-end or `clang` from Clang (2012) to translate this source code into LLVM IR. Since we translate directly the source code (not a machine code), variable and function names are usually kept untouched. This provides a more readable output. Currently, among supported languages belong, e.g., C, C++, C#, D, Objective-C, Objective-C++, Haskell, Ada, Fortran, Pascal, see The LLVM Compiler System (2012).

Pre-processing

After we have an LLVM IR representation of the input source file, we run several optimizations over it to produce a more appropriate code for the successive conversion. Indeed, in the successive stages, we need the LLVM IR in a special form. For example, induction variables and loop exits have to be canonicalized (see Passes in the LLVM Compiler System 2012). We use the `opt` tool from LLVM for this purpose, which includes several built-in analyses and transform passes.
We also use our own passes. For example, if variable names are altered during the translation, we give them more readable names. For instance, induction variables of loops are named `i`, `j`, `k` and the variable holding a result of a function (if any) is named `result`.

Conversion to a high-level language

As LLVM IR is pretty low level, we first convert it into a uniform high-level form of representation. It is in the form of an abstract syntax tree, and it is independent from the target HLL. During this conversion, we identify high-level constructs, such as loops and conditional statements, which were lowered to labels and branches during the translation into LLVM IR.
After that, we run several optimization passes over our representation, like various data-flow analyses to clear the resulting code. The optimized code is then emitted in the selected HLL.
Currently, we support only the Python programing language. However, other output languages are planned to be supported in the future.

Post-processing

After the emission into the target language is completed, an additional post-processing phase is done to further improve the readability of the code. These modifications are done on textual level, and include the elimination of redundant brackets and simplification of conditions. For example, “`if not (a == b):`” is simplified into “`if a != b:`”.

EXPERIMENTAL RESULTS

In this section, we present three experimental results of code migration from various languages into Python. In each example, we give a source code, migrated code, and several notes concerning the migration. At the end of this section, the third result is compared with an existing single-purpose converter.

First example

In the first example, we migrate an implementation of the Ackermann function in C. The source code is given in Figure 2. The migrated code in Python can be seen in Figure 3.

```
unsigned ack(unsigned m, unsigned n) {
    if (m == 0)
        return n + 1;
    else if (n == 0)
        return ack(m - 1, 1);
    else
        return ack(m - 1, ack(m, n - 1));
}
```

Figure 2: Source code in C.

The following three facts should be observed: (1) `llvm-gcc` turned one recursive call into an iteration. While both versions are functionally equivalent, during code migration, a one-to-one translation is often desired. If this is the case, turning off compiler optimizations might result into a more corresponding code. (2) The function's name and both arguments' names have been preserved. Compiler-generated temporary local variables have been given more readable names. Indeed, the original names for `orange`, `i`, `banana`, and `result` were `n_tr7`, `indvar`, `n_tr_be` and `n_tr_be_lcssa`, respectively. (3) In the intermediate LLVM IR code, there is no evidence of the `for` loop, just branches to labels. We used our and LLVM's analyses to structure the code into a high-level `for` loop. This makes the resulting code more readable.

```
def ack(m, n):
    if m == 0:
        return n + 1
    orange = n
    for i in range(0, m):
        banana = 1
        if orange != 0:
            banana = ack(m - i, orange - 1)
        orange = banana
        result = banana
    return result + 1
```

Figure 3: Migrated code from Figure 2 into Python.

Second example

In the second example, we migrated an implementation of the factorial function in the D programming language. The input source code is given in Figure 4. The resulting code in Python is available in Figure 5.

```
ulong factorial(int n) {
    if (n < 2)
        return 1;
    else
        return n * factorial(n - 1);
}
```

Figure 4: Source code in D.

Again, as in the first example, the compiler turned the input, recursive version into an iteration. The function's name has been preserved although its name has been mangled. The same holds for the argument's name.

```
def _D4fact9factorialFiZm(n_arg):
    if n_arg < 2:
        return 1
    orange = 1
    for i in range(0, n_arg - 1):
        orange = (n_arg - i) * orange
        result = orange
    return result
```

Figure 5: Migrated code from Figure 4 into Python.

Third example

In the third and last example, we translated a recursive implementation of the factorial function in Fortran, given in Figure 6, into Python. The resulting version can be seen in Figure 7.

```
recursive function fact (n) result (value)
    implicit none

    integer n
    integer value

    if (n <= 0) then
        value = 1
    else
        value = n * fact(n - 1)
    end if

    return
end
```

Figure 6: Source code in Fortran.

Contrary to the previous two examples, in this case, the recursion has been kept. Apart from this fact, the same observations as in the first example can be made.

```
def fact(n):
    orange = n
    if orange < 1:
        return 1
    banana = fact(orange - 1)
    return banana * orange
```

Figure 7: Migrated code from Figure 6 into Python.

A comparison to an existing solution

In the rest of this section, we compare the third result with the existing single-purpose converter: `f2c` (Feldman 1990). The C code migrated from a recursive factorial function, listed in Figure 6, is presented in Figure 8. In comparison with our solution, one of the disadvantages of `f2c` is a restriction to the Fortran77 dialect as the input language. Therefore, our input had to be properly modified. Also, we can mention that the function's argument is passed as a pointer and there is no check of its value. Such an intrusive change of program's behavior is dangerous and it can cause the application to crash during run-time. Moreover, using a pointer in this case is not needed. Both solutions generate new local variables with their own name schemes.

FUTURE RESEARCH

We conclude the paper by suggesting two areas for future research.
As for possible improvements of our source-to-source migration process, other HLLs can be considered, possibly including type information and conversions between types, like in C.

```
integer fact(integer *n) {
    /* System generated locals */
    integer ret_val, i__1;

    ret_val = 1;
    if (*n >= 1) {
        i__1 = *n - 1;
        ret_val = *n * fac_(&i__1);
    }
    return ret_val;
}
```

Figure 8: Migrated code from F77 into C by `f2c` (Feldman 1990).

Furthermore, one may consider implementing new front-ends for `llvm-gcc` to support other source languages. Since `llvm-gcc` is an open-source compiler, users who want to compile their sources in not-yet-supported languages would benefit from a front-end like this, too. Indeed, the front-end may then be used to compile source code into LLVM IR which can then be converted into machine code for a specific platform by an existing back-end.

In future research, we would also like to focus on binary code migration. Such a process may be very effective for porting existing applications to new platforms. Observe that binary code migration may become very useful if the original source code is unavailable.

We plan to use the decompiler's front-end for decoding and translating platform-dependent applications into LLVM IR. After that, we plan to re-use our built-in optimizations in the middle-end. Finally, we plan to use target-architecture LLVM back-ends for assembly code generation.

ACKNOWLEDGEMENTS

This work was supported by the research funding TAČR, No. TA01010667, BUT FIT grant FIT-S-11-2, by the Research Plan MSM0021630528, and by the IT4Innovations Centre of Excellence project CZ.1.05/1.1.00/02.0070.

REFERENCES

ANTLR, 2012. `http://www.antlr.org/`.

ANTLR's page on LLVM, 2012. `http://antlr.org/wiki/display/ANTLR3/LLVM`.

Clang, 2012. `http://clang.llvm.org/`.

Dickens T., 2002. *Migrating Legacy Engineering Applications to Java*. In *OOPSLA'02 Practitioners Reports*. ACM, New York, NY, USA, OOPSLA'02. ISBN 1-58113-471-1.

Emscripten, 2012. `https://github.com/kripken/emscripten/wiki`.

Feldman S.I., 1990. *A Fortran to C Converter*. *SIGPLAN Fortran Forum*, 9, no. 2, 21–22. ISSN 1061-7264.

Kontogiannis K.; Martin J.; Wong K.; Gregory R.; Müller H.; and Mylopoulos J., 1998. *Code Migration Through Transformations: an Experience Report*. In *Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, CASCON '98, 1–13.

Lerew E.W., 1999. *Migration of Legacy Test Programs to a Modern Computer Platform [for Avionics Testing]*. In *AUTOTESTCON '99. IEEE Systems Readiness Technology Conference, 1999. IEEE*. ISSN 1080-7725, 293–298.

Lissom Project, 2012. `http://www.fit.vutbr.cz/research/groups/lissom/`.

llvm-js-backend, 2012. `http://github.com/dmlap/llvm-js-backend`.

Masařík K., 2008. *System for Hardware-Software Co-Design*. Faculty of Information Technology BUT, Brno, CZ, 1st ed.

Mills M. and Peterson G., 1998. *Hardware/Software Co-Design: VHDL and Ada 95 Code Migration and Integrated Analysis*. In *Proceedings of the 1998 annual ACM SIGAda international conference on Ada*. ACM, New York, NY, USA, SIGAda '98. ISBN 1-58113-033-3, 18–27.

Passes in the LLVM Compiler System, 2012. `http://llvm.org/docs/Passes.html`.

Python Programming Language, 2012. `http://www.python.org/`.

The LLVM Compiler System, 2012. `http://llvm.org/`.

Ďurfina L.; Křoustek J.; Zemek P.; Kolář D.; Masařík K.; Hruška T.; and Meduna A., 2011. *Design of a Retargetable Decompiler for a Static Platform-Independent Malware Analysis*. *International Journal of Security and Its Applications*, 5, no. 4, 91–106.

Yang H., 2003. *Successful Evolution of Software Systems*. Artech House, Norwood, Mass. ISBN 9781580533492.

Zhong H.; Thummalapenta S.; Xie T.; Zhang L.; and Wang Q., 2010. *Mining API Mapping for Language Migration*. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1*. ACM, New York, NY, USA, ICSE '10. ISBN 978-1-60558-719-6, 195–204.

System Dynamic Simulation Engine on Java platform

Ľuboš Chupáč, Jan Kožusznik, Svatopluk Štolfa, Jan Korpas, Oldřich Vašut
Department of Computer Science
VSB - Technical University of Ostrava
17.listopadu 15, Ostrava-Poruba
Czech Republic
E-mail: {lubos.chupac, jan.korpas.st, jan.kozusznik, svatopluk.stolfa, oldrich.vasut }@vsb.cz

KEYWORDS
System dynamics, software development process

ABSTRACT

The paper demonstrates possibility of the JAVA technology for a creation of a system dynamics simulation. This paradigm is demonstrated on a model of software development process. The paper presents our experiences with a model creation. System dynamics simulation based on the JAVA technology has mainly benefits for academic purposes – it enables various experiments.

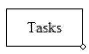
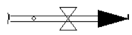
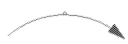
INTRODUCTION

Continuous simulation System Dynamics (Raffo, Vandeville, and Martin 1999; Ruiz, Ramos, and Toro 2004; Mildeová and Vojtko 2008) is one of the leading simulation paradigms. It is controlled by continuous time and change of parameters in process is modeled as system of differential equations.
Many mature simulation tools for system dynamics exists: ExtendSim (*ExtendSim Simulation Software by Imagine That Inc*), Vensim (Ventana Systems 2010), Anylogic (XJ Technologies 2011), Powersim (*Powersim Studio 8 | Powersim Software*). This presented paper shows using JAVA technology for a creation of a system dynamics simulation. This approach has benefits mainly for academic purposes: a required tool is available and there are not additional expenses; and it is open to customization required for various experiments.
This paper is organized as follows: *System dynamics* provides a brief overview of problematic; *The issue of software system development and changes* explains two approaches to software system development; *Reference model* describes model of a case study; *Summary of the model variables* defines variables used in a case study; following chapter *Transfers of measurement units their relationships* solves conversion among different units; *Creating reference model* shows creation of model in the simulation tool Vensim whereas *Implementation of the system dynamics using Java* demonstrate creation in programming language JAVA; and finally *Conclusion* summarizes positives and negatives of the simulation environment.

SYSTEM DYNAMICS

System dynamics model software process as a closed-loop version of a system (Figure 1). Input represents requirements, specifications and resources; otherwise output represents artifacts. The input is transformed to output and the operation is influenced by a controller that changes the behavior of the system.
System dynamic simulation approach was developed by Jay Wright Forrester – published in (Forrester 1961). It was adopted for the software process modeling purpose by Tarek Abdel-Hamid (Abdel-Hamid and Madnick 1991). This approach is also named continuous system modeling because the model is computed in respect to continuous time. It is the main difference from discrete-event modeling.
System dynamics model is composed from elements (adopted from (Madachy 2008)) defined in Table 1.

Table 1 Elements used in SD models

Element	Notation	Description
Level		It represents an accumulation over time – also called a stock or state variable.
Source/Sink		It indicates that flows come from or go to somewhere external to the process.
Rate		It is also called flow and it effect changes in levels.
Auxiliary	Only label	It is converter of input to outputs, it helps elaborate the detail of level and rate structures.
Information Links		It is used to represent the information flow that could influence value of some auxiliary or rate element.

The system is modeled as a structure of levels connected by flows and information links. Flows between levels represent process entities that are conserved within flow chains. Information links only provide data from auxiliary, flow or level to another auxiliary or flow (Figure 2). Individual events are not tracked; tasks are modeled by levels and the system could be described by system differential equations.

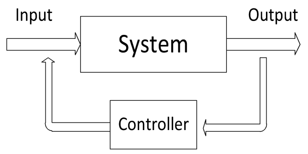


Figure 1
Closed-loop system

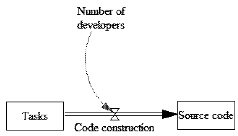


Figure 2
Code construction

It has a very simple parallel to the physical system of water basins (level) connected with valued pipes (rates). Pipes could be controlled based on the level of other water basins or other computed factors (auxiliary). Information about control is represented by an information link. The amount of level in a specific time T is equal to:

$$\int_0^T \left(\sum_{i \in Input} f_i(t) - \sum_{o \in Output} f_o(t) \right) dt$$

Functions $f_i(t)$ represent input flows to a level, while $f_o(t)$ represents output flows.

THE ISSUE OF SOFTWARE SYSTEM DEVELOPMENT AND CHANGES

Creation of software systems may be based on two approaches. The first one is the development of a new software system, which is not influenced by earlier products. The second approach is upgrade of the existing software system, which is in most cases a developmental change and improvement of the system during its life cycle.

Our simulation is based on the second approach where change of software system takes place. Software change can be understood as a very extensive and complicated process of which modeling is difficult, because it can very quickly lose the modeling of links between individual threads that make up this process.

For a basic understanding of software simulation changes, you should be familiar with the basic chain, which is a key point of the software system. As shown on Figure 3, these points are core requirements generation, planning requirements, modification of design, design review, code modification and testing.

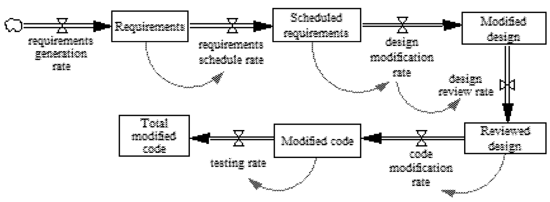


Figure 3 Change of software system the basic change.

It is obvious that the result of this simulation would be applied only in a completely inert and isolated environment where the simulated process is not influenced in any way by other factors. If we want to simulate the evolution of software system change as far as possible, we have to set the factors that correspond as closely as possible to reality and practice development. It is hard to imagine how a large number of factors operate on the outcomes of the simulation work, and it is just through the sub-processes which were introduced above. We could specify fractional random list of the most important sub-processes, such as requirements specification, planning and allocation of requirements, design, design review, approval, design, coding, testing and bug fixing, deployment, hiring resources, etc. Each of the sub-processes could be represented by a separate model large enough that participates in the main model, thereby extending the basic model of Figure 3.

There is illustrated how the basic chain can be extended with the sub-processes in the Figure 4. Even completed simulation model may not be the final model, there are number of factors which can enter into the change of software, and thus affect the final result. For example, the processes of employee turnover, employees gain experience over time, the effect of fatigue on employee productivity, etc.

As we mentioned above, this complex model for a reference model, which aim is to explain the process of the simulation is not good enough in the system dynamics. Orientation in this developed model would be complicated and whole process would be hard to understand. For our purposes, largely diminished model change of software will be considered, despite this fact that we provide sufficient insight into the modeled problem and express all the essentials.

REFERENCE MODEL

Our selected reference model will be described and discussed below. It is only part of the overall model simulating changes of software, specifically the chain developed to simulate the phase of planning tasks and design modifications. Before we proceed to create the reference model itself, it is necessary to define some basic properties of the modeled problem (Madachy 2008). This definition is a condition sine qua non, as it is necessary to create a framework under which we will be able to create a model that will fit our ideas and that will present a simulation of the desired problem. Basic features needed are: the purpose of the model, border of the model, key variables and finally a description of the desired behavior of the model. There are many methods and procedures for purposes of the definition, so it is up to creators of the model, which way they will choose.

The purpose of the selected reference model is to enable the simulation of situations that may arise during the process of planning tasks and design modification of the software change. We will be able to monitor the time required to complete this process, the number of errors

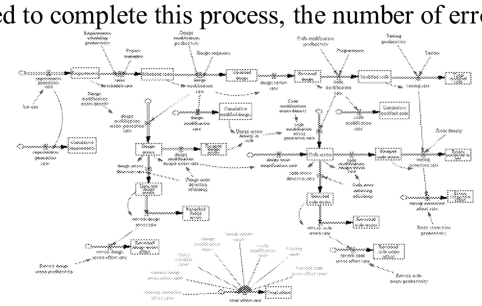


Figure 4 Developed basic chain over next sub-processes

which occur in the design phase of modifications, and what effort will be currently needed in different time periods.
Simulation will provide us valuable data that will be used to achieve maximum efficiency in the process of planning tasks and software modifications to change the software. It is clear that these data and the efficiency of this phase achieved a high degree of contribution on a global scale throughout the simulations of the software changes, as an integration of the reference model as a set of sub-processes into a coherent model of a complete development software system changes.
Another essential feature is the behavior of the model. Creator of the model for the definition of this property has to imagine how the process works in real environment and what factors affect the process. Our presented reference model begins with planning tasks. In the first phase the project managers will receive a list of demands for change the entire software system, where the main task is to plan the whole process of the change. Time needed is in a direct proportion depended on their performance. Planning change of process is followed by modification phase of the design, which is provided by a team of design engineers. The main of these engineers are steps leading to the required change in the software, based on the task list, which is received as input information. The outcome of their activity is therefore revised design, it is design changed against the original design, which existed before the desired change. Also in this case it is necessary to specify the value of the performance of workers determining the volume of tasks that they are able to handle for selected time unit.
In the real environment, all processes are affected by human errors or equipment failures, it is necessary to include these errors against isolated environment. These errors would logically appear only in the design modifications, and their number will be determined by density of error attributable to a single task. Errors will be further divided into two types. The first are mistakes that a team of design engineers reveal in the process of the design modification. It is easy to remove these errors at this stage. The second type are errors which are not detected by anyone in design modification and will affect the coding phase that which follows afterwards (further development is beyond of the scope).
The last watched metric is the result efforts on the modeling process. The result value includes efforts at the planning tasks in design phase, in design modification phase including reveal and further elimination of errors in design phase. This metric allows us to measure the actual effort required to process during the selected time.

SUMMARY OF THE MODEL VARIABLES

Input parameters

The input parameters are the values that define the behavior of the model (Šusta and Neumajerová 2004). These variables can be selected on demand and whole process change in software can be optimized during the simulation. Parameters can be divided into three types: employees, productivity and density of errors. Employees determine how human resources are needed at this stage, productivity defines the number of tasks that workers are able to process daily, and density of errors is the ratio of errors resulting from the processing of a task. All input parameters are clearly shown in Table 2.

Tasks scheduling productivity	Design modification errors density
Project managers	Design error detecting efficiency
Design modification productivity	Rework design errors productivity
Design engineers	

Table 2 Input parameters

Levels

The reference model is represented by three basic levels, which are Scheduled tasks, Design errors and Total effort. Other levels are derived from these essential. Table 3 shows the complete list of primary and secondary levels, which are used in the reference model. For a better overview, it is divided into three categories, the criteria are the names of measure units, which are marked the individual levels.

[Tasks]	**[Errors]**	**[Effort]**
Scheduled tasks	Design errors	Reworked design errors effort
Cumulative modified design	Escaped design errors Detected design errors Reworked design errors	Total effort

Table 3 Level list

The main flows

Even in the relatively limited reference model relatively high number of flows can be found that affects the individual level. For an example, several major flow models that affect us of maintaining the most important values of levels. The first flow *tasks schedule rate* provides planning of tasks which are then processed in process change of software. Another interesting flow *design modification errors generation rate* which generate errors in the phase of design modification. The third main flow is identified as *total effort rate* which count all effort made in the model.

The results

Output can be considered for all levels and flows, after the simulation finishes, we can monitor their outcome or progress in a time. For our purposes the most relevant outputs is *total effort*, *total effort rate*, *escaped design errors*, *reworked design errors*. The value of multiplier occurrence of errors in coding phase could be considered as an interesting *design errors density in code.*

TRANSFERS OF MEASUREMENT UNITS AND THEIR RELATIONSHIPS

In previous section mentioned labeling units of measure levels *Tasks*, *Errors* a *Effort*. It is common to identify the scope of software units in *SLOC* (Source Lines Of Code) in software engineering. In the reference model the basic parts of unbranched chain of the process of software changes (ie. planning phase for tasks and design modification) will be selected as metric units *Tasks*. For the purposes of this particular model we will assume the following:

- 1 Task = 60 SLOC
- 1SLOC = 1 Error
- 1 Task = 2.5% errors

In the process of generating simulation errors there were very low value results in this part of the chain and it is appropriate to convert units *Task* on *SLOC.* The result is ratio 1 Task = 1.5 false SLOC.
As pointed above one of the most important level is overall effort which is count of three flows *tasks schedule rate*, *design modification rate* a *rework design errors effort rate*. Therefore it is necessary to recognize the fundamental relationship between units Tasks and Errors with which these streams are working. The values mentioned in the two flows are expressed in a unit of the Task – this is number of Tasks handled per chosen time unit. On the other hand the third of these flows is working with a unit of Errors. To obtain the relevant value of the total effort is necessary for a third flow to convert units of Errors to Tasks, in the proportions: 60 Task Errors correspond to 1 Task.

CREATING A REFERENCE MODEL

We discussed all basic concepts system dynamics is working with process simulation of change in software, the properties that need to be selected before simulation were mentioned. We defined all inputs, outputs and variables of model and a graphical representation allows better understanding of the components and links that were mentioned earlier in the article on theoretical level.

Simulation tool Vensim

If we want to use advantages which give us the simulation and modeling we have a wide range offer of simulation tools which we can choose from.
For simulation of reference model, a simulation tool Vensim (Ventana Systems 2010) was chosen which is well known and widely used for the realization of system dynamics models. The advantage of this tool is that its version academic version Vensim PLE is free for researches and noncommercial purposes. On the other hand, it has limited functionality.

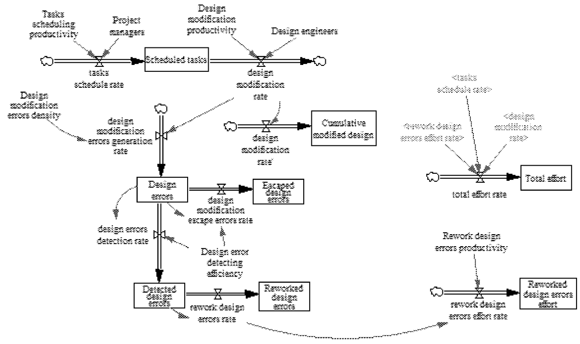


Figure 5 The reference model of change in software

Model simulation in Vensim tool

The aim of the selected simulation tool is to create a reference model, but not only from graphical view but also from functional point of view. However it is desirable to see the reference model in isolation, which allows us to perceive connections more specifically, relations and links between different elements of the model with which we have been familiar only theoretical. Graphic design model is shown on Figure 5.
In the previous description of the problem of simulation and modeling, we did not mention the equations, which are ~~a~~ the key artifact of the desired behavior of the entire model as well as their output values calculated by means of simulations. Creation of the equations, except for few exceptions (equations generated in the levels), is the creator issue, reflecting its specific requirements for the simulation. List of equations that can be used in the simulation to calculate the levels, flows and variables is shown below:

```
Detected design errors = INTEG (
        design errors detection rate - rework
design errors rate, 0)

Design modification errors density = 1.5

tasks schedule rate =
        Project managers * Tasks scheduling
produktivity

design modification escape errors rate =
        Design errors * ( 1 - Design error
detecting efficiency )
```

If you create a set of equations the reference model is fully ready to perform simulations. The actual simulation is of course always dependents on the form of equations that model makers assembled to the case.
In this article we will perform one of the countless simulations in the reference model. The scope of the equations developed is beyond the capability of this article. Some of the key values are discussed bellow.
In Table 4 the results are recorded after first ten steps of simulation reference model related to the flow *rework design errors rate, reworked design errors* and *total effort.* In the first three rows of columns with the value of levels *reworked design errors* are zero values which are the result of delaying the process errors generation, their

finding and consecutive removal. Graphical progress on this level of that delay is shown in Figure 6.

Time	Rework design errors rate	Reworked design errors	Total effort
0	0	0	0
1	0	0	24
2	1,44	0	48
3	2,448	1,44	72,024
4	3,1536	3,888	96,0648
5	3,64752	7,0416	120,117
6	3,99326	10,6891	144,178
7	4,23528	14,6824	168,245
8	4,4047	18,9177	192,315
9	4,52329	23,3224	216,389

Table 4 Results of chosen elements in Vensim

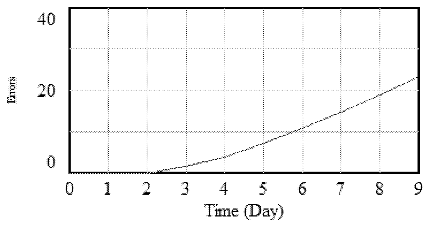


Figure 6 Rework design errors

IMPLEMENTATION OF THE SYSTEM DYNAMICS USING JAVA

Although many of tools exist that allow the creation of models using system dynamics and then run the simulation, it is still necessary to create a new tools from the beginning to the end. The reason may be charging many of simulation tools for their full versions, inadequate framework for processing or just a desire to create a tool sui genesis. If a company likes to create its own tool for creating and simulating the models it most likely run into a fatal lack of libraries and related packages. It is essential that the creation proceeded from the beginning and this library has to be created by himself.
In this paper, we decided for presentation purposes to create library in Java that will provide a simulation engine for system dynamics modeling tools. Our chosen approach is one of the possible ways of resolving this issue.

Library description

The library of simulation engine what we developed consists of several basic classes divided into two packages. The first package contains classes that represent items of system dynamics `Stock`, `Flow`, `Auxiliary` and their superclass `Item`. These classes from the first package contain basic properties to keep names of variables, equations and initial values for class `Stock`. The second package contains simulation class `Simulation` and two classes for returning simulation results `Result` and `ResultItem`. The first mentioned class provides the calculation of equations for simulation model of selected number of steps (description of the calculations will be given). Values of each step of the simulation are always available for each element. This makes it possible to display continuously the value of each element which is included in the simulated model.
During the development of this library there was a need to deal with the possibility of embedding simulation run in two modes. The first mode is to run simulation as whole by making the results after its complementation. The second is phased run of simulation for individual steps in which the results are always available after the step is complete. The whole process of phasing show great variability in the continuously results, they can be viewed in real time or step by step, always on the user`s request.

Solution

The most important part of the library is the area of equations calculation which are irreplaceable for the simulation and which the whole model of system dynamics consists from. There exist many procedures for equation calculation for example using of mathematical libraries. Our solution will calculate equation of model as a system of equations using the parsing functions. The chosen solution is based on principle of replacing the variables in the equations model by their values which may be numeric constants or equations. In the case when it is substituted for equation variable and their variables are substituted again until equation without variables arise. These new compiled equations are calculated and their final results are saved.
If the equations are represented as strings of characters we need to parse each of these numeric constants, variables and mathematical operations for reason of correct calculation their results. For these purposes it is good to use a Java and the library JEval that has all the methods needed for proper mathematical evaluation and calculate given equation.

JEval library

JEval is a library working with mathematical, string, boolean and functional formulas that can parse and evaluate. It includes a variety of mathematical operations and functions for example sin, cos, ln, ie... The major advantage is that library is provided library free (JEval).
For our need we use these methods: `replaceVariables` and `evaluate`. The first provides substitution of variables in equation by their value (number or equation). Using this method is necessary because in system dynamics often can happen that the calculation of variable is deeper in the structure of model and method `evaluate` does not solve this problem. This method only works with variables within single level. Therefore, first we need connect all together using `replaceVariable` to one imaginary level and thus equations can be easily calculated.
This library has also disadvantage, but their existence does not cause major complications. For example it is necessary to record all variables defined–within model in notation `#{variable_name}`.

Simulation model using Java library

Java library developed for simulation model using method of system dynamics which is described above satisfies all necessary requirements and functions allowed to run simulations of the reference model. Simulation is primarily used to verify proper functionality of our Java library and therefore the model will not be built in the user interface, but only with the instance of the `Simulation` class. Record and definition of the model is shown below:

```
Simulation simulation = new Simulation();
simulation.setStartSimulation(0);
simulation.setStepSimulation(1);
simulation.setEndSimulation(10);

Set<Stock> stocks = new HashSet<Stock>();
stocks.add(new Stock("Scheduled tasks",
        "(#{tasks schedule rate}-#{design
modification rate})", 0));
...

Set<Flow> flows = new HashSet<Flow>();
flows.add(new Flow("tasks schedule rate",
        "(#{Project managers}*#{Tasks scheduling
productivity})"));
...

Set<Auxiliary> auxuliaries = new
HashSet<Auxiliary>();
auxuliaries.add(new Auxiliary("Project managers",
"2"));
...

try {
  Result result = simulation.play();
}
catch (EvaluationException e) {
  e.printStackTrace();
}
```

Results of the simulation are recorded in Table 5.

Time	Rework design errors rate	Reworked design errors	Total effort
0	0	0	0
1	0	0	24
2	1,44	0	48
3	2,448	1,44	72,024
4	3,1536	3,888	96,0648
5	3,64752	7,0416	120,11736
6	3,99326	10,68912	144,17815
7	4,23528	14,68238	168,24471
8	4,4047	18,91767	192,31529
9	4,52329	23,32237	216,38871

Table 5 Simulation results

CONCLUSION

To verify the correctness of the procedure for creating libraries, we used our well-known reference model on which we perform simulation in Vensim. After the simulation which is using our library developed in Java we get the results and their values are shown in Table 5. Comparing the results with those onces originated from our simulation we can verify the identity of values from the both simulations. Therefore, we can declare that our chosen approach to create a library using Java is correct. Verification process using the results from the Vensim tool is regarded as authoritative, because it is a professional tool with a long tradition.

There were developed countless of new methods using the system dynamics in the simulation and modeling, it has still an indispensable role and it can be also currently used with no difficulty. System dynamics keeps pace with newly developed methods also due to the fact that it is compatible with the modeling tools that are currently available on the market.

Benefits of the article can also be certainly considered as initiative to create new simulation tools. The aim was to create a functional, simple and easy to use library that will have all the important properties for system dynamics simulation models. This library can be further used for the realization of its simulation tool, or it can just illustrate approach to system dynamics. I hope that this article extended awareness about system dynamics.

Acknowledgements. This research has been supported by the internal grant agency of VSB-TU of Ostrava - SP2012/26 „An utilization of artificial intelligence in knowledge mining from software processes“

REFERENCES

Abdel-Hamid, Tarek, and Stuart Madnick. 1991. *Software Project Dynamics: An Integrated Approach*: Prentice Hall, ISBN 0138220409.

ExtendSim Simulation Software by Imagine That Inc. Available from http://www.extendsim.com/.

Forrester, Jay Wright. 1961. *Industrial Dynamics*: Pegasus Communications, ISBN 1883823366.

JEval. 2012. *JEval* [cited 2.12. 2012]. Available from http://jeval.sourceforge.net/.

Madachy, Raymond J. 2008. *Software Process Dynamics*. 2nd ed: Wiley-IEEE Press, ISBN 0471274550.

Mildeová, S., and V. Vojtko. 2008. *Systémová dynamika*. 2nd ed. Praha: Oeconomica, ISBN 9708-80-245-1448-2.

Powersim Studio 8 | Powersim Software. Available from http://www.powersim.no/.

Raffo, D. M., J. V. Vandeville, and R. H. Martin. 1999. "Software process simulation to achieve higher CMM levels." *Journal of Systems and Software* no. 46 (2-3):163-172.

Ruiz, M., I. Ramos, and M. Toro. 2004. "Using dynamic modeling and simulation to improve the COTS software process." In *Product Focused Software Process Improvement*, edited by F. Bomarius and H. Iida, 568-581.

Šusta, M., and I. Neumajerová. 2004. *Cvičení ze systémové dynamiky*. 1th ed. Praha: Oeconomica, ISBN 80-245-0780-3.

Ventana Systems, Inc. . *Vensim® PLE for Windows Version 5.10b.* 2010. Available from http://www.vensim.com/software.html.

XJ Technologies. *AnyLogic version 6.1* 2011. Available from http://www.xjtek.com/.

MATLAB TOOL SUPPORT FOR SAFETY ANALYSIS OF HIGH CONSEQUENCE ARMING SYSTEM DESIGNS

Dan Slipper and Alistair A. McEwan
Department of Engineering
University of Leicester
University Road
Leicester, LE1 7RH

Wilson Ifill, Gordon Hunter, Roger Green and Richard Johnson
AWE, Aldermaston
Reading
Berkshire
RG7 4PR

KEYWORDS
Safety Analysis, Matlab, Simulation, Propagation

ABSTRACT

High consequence arming systems are designed to prevent unwanted external (or potentially internal) energy flowing to a critical component without authorisation. Hazard analysis of such systems can be a slow and difficult manual process, potentially repeated in various life cycle phases or on multiple design options. This paper details a simulation tool under development at AWE to provide a fast and repeatable analysis process. The Matlab tool provides an object-oriented modelling approach representing the system as a set of components, the topology in which they arranged and the hazards present in a given scenario. The simulation generates a set of possible paths along which different energy types could potentially propagate through the system. The state of the critical component as a result of these event sequences is also output, aiding identification of potentially unsafe system behaviour. We present an outline of the model development process, results from its use with a case study and demonstrate the advantages over manual analysis. A number of limitations of the current implementation are discussed, we then propose future work aimed at alleviating some of these issues.

INTRODUCTION

High consequence systems 'where failures can cause catastrophic results' (Davis 2000) pose a potential hazard throughout their operation, for our specific industry the catastrophic results would be radiological dispersal or nuclear yield. Where possible these hazards should be removed from the system during design to maintain assured safety. If this is not possible, the impact of such hazards should be reduced through the implementation of safety features. The effects of external *insults* 'physically measurable phenomena with the potential to detrimentally affect the system' (Johnson 2009) must also be considered to ensure that the system remains adequately safe through all environments. The Nuclear Weapon regulations and design principles (JSP 538 (Ministry of Defence 2008) and JSP 372 (Ministry of Defence 2011)) provide the high-level safety requirements for system designs which are based strongly on the concepts of Isolation, Incompatibility and Inoperability as described further in (Covan and Cooper 1998), (Plummer and Greenwood 1998), (Elliott 2005) and (Ekman et al. 1998). Due to the Comprehensive Nuclear Test Ban Treaty (Comprehensive Test Ban Treat Organisation 1996) full system tests are not possible. Software based modelling methods can aid analysis of the system by providing the user with a fast, thorough and repeatable process. We introduce a Matlab tool designed to automate a manual technique used at AWE (Johnson 2009), by identifying the possible paths from an insult to a critical component of the system. The tool allows a safety analyst to address non-design mode connectivity through the system, considering electrical and non-electrical hazards and transformations. The output from the tool is a list of the unsafe pathways through the system (some of which could be potentially missed using the manual method).

This paper provides an overview of the modelling technique and elements of a model, an example use of the tool on a simple example and some discussion and conclusions of how the tool can be improved in the future.

MODEL ELEMENTS

The elements used to model a system are the component responses, insults and a system topology. Figure 1 shows each of the stages and the output created by the simulation. When the model is executed it uses a path generator and insult propagator to analyse all possible paths then identifying ones which are potentially unsafe.

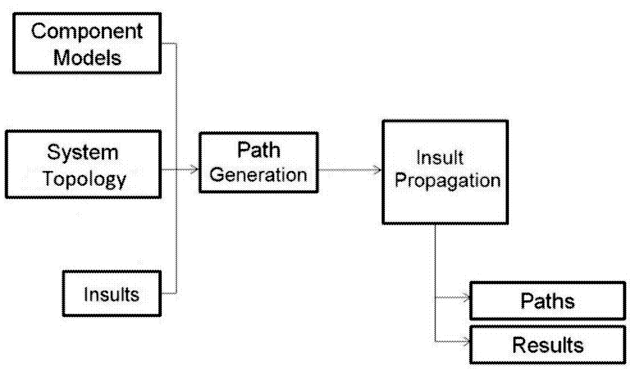


Figure 1: Elements of the Simulation Tool

Each of these stages is described in more detail in the following sections.

Component Models

Each component in the system model will either be a functional component or designed to provide safety. For either scenario energy may be generated by, contained by, or passed through a component. Potentially a component can have multiple responses, dependent on the type and magnitude of energy that is received. Although Matlab provides the ability to represent component responses by transfer functions, our safety analysis is concerned the point that safety is not assured. The component models are developed using a rule-based approach (Zadeh 1983), where statements represent the safety assured responses. Use of an object-oriented design allows each instance of a component to be created with its thresholds and output values as constructor arguments. Assigning these values from the main program code allows the simulation to be modified and re-run easily. In early system design stages (e.g. generating and testing system layout options) the information about component response functions would be difficult to provide accurately. The values assumed would be the estimated worst case for assured safety and based on expert judgement. This development process allows model specifications to be refined as further detailed design information becomes available.

Insults

To analyse a scenario of interest to the safety analyst an insult source should be provided. The insult could involve multiple types of energy being introduced at a given point. In Matlab the insult is captured as an object containing a vector (or array) of values. The object is passed between components along the paths that are generated by the system. The values in the insult vector are modified where appropriate as they pass through each component, until the critical component is reached.

System Topology

To capture components interacts the relationships between them should be modelled. This is to capture where the insult vector can be passed between components. In abnormal environments it is assumed that all components could potentially become coupled with all others unless a safety feature removes this connectivity (such as an area isolated by a barrier). The connections between components can be represented by an adjacency matrix (Kreyszig 2010). Capturing all of the interactions in this way aids the modelling of the system in Matlab. Each row of the matrix would be translated into an appropriate line of code detailing where each component could connect to.

Path Generation

To generate a list of all of the paths from the insult source through to the critical component, a breadth first search (Jungnickel 2008) of the system is performed. This algorithm identifies each connection from its current component then in turn repeats the process on each branching component. This generates a tree across the breadth of the graph rather than the depth. When a full chain is found that does not end with the critical component, it is removed from the list. The resulting paths are written to a file and provide the first part of the output.

Insult Propagation

The propagation element of the simulation iterates through each of the paths generated, taking an insult magnitude that is input to the system and then calculating the change to this vector based on the component responses. The components in the system are instantiated by calling their constructor with the appropriate values (setting thresholds and outputs based on the components responses). This could look like:

i1 = I(50); %electrical signal of 50V
a1 = A(40, 50, 80); %threshold of 40V, output 50V, 80 deg
b1 = B(..., ..., ...);

The final component determines if it is unsafe based on the magnitudes held within the insult vector (if applicable). The resulting output is a list of statements of whether the paths are safe or unsafe, linking to the order in which they were generated. Once this point has been reached the component generates a `safe' or `unsafe' response. In the current implementation only a single critical component is modelled.

MODELLING EXAMPLE

To demonstrate the use of this simulation tool a simple case study analogous to that of a high consequence arming system has been devised. The selected case study is of a car where the hazard being protected is the fuel tank, which provides a constant hazard of fire or explosion. Although vehicles are now designed with safety of fuel tanks in mind and appropriate measures are considered. Older vehicle designs have demonstrated this risk, for example the Ford Pinto was designed with the fuel tank rear of the axle, which according to (Schwartz 1991) limited the amount of crush space around the component. This design decision resulted in explosions from the fuel tank upon high speed impact to the rear of the vehicle. A few example components from a vehicle have been modelled to each have a number of responses to insulting energy which could in some way affect the fuel tank. These components as shown in Figure 2 are the: Fuel Tank (FT), Lighter (L), Brake Pedal (P), Brake Light (BL) and Battery (B). In reality, this combination of elements are unlikely to be collocated within the system, however in accident scenarios it must be ensured that they would be sufficiently isolated or have incompatible energy types.

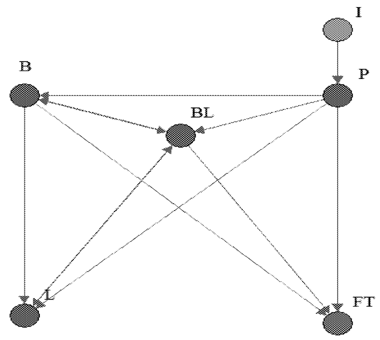


Figure 2: Potential Connections in the Example System

The scenario under test was a shock to the brake pedal mechanism in a drive by wire system, producing an electrical signal elsewhere. The case study was analysed manually to identify the expected results and compared against the output from the model to validate its results. The output gave 16 paths through the system, 2 of which were potentially unsafe in this scenario. It was noted during manual path generation that possible paths could easily be missed, even when generating possibilities with a small system of 5 components. This becomes far more difficult with a larger set of components and also with more potential energy types. The simulation itself only takes a matter of seconds to execute, providing a fast method to get repeatable results. To change the component specifications and test a range of insulting values only required changes to the constructor of the insulting component model.

DISCUSSION

The overall outcome from using the modelling techniques is that the generation and analysis of paths with use of a software based tool is faster and more accurate than the manual process. Removing the aspect of human error from the analysis process is an important motivation for this development. Use of the simulation tool allows repeatability of experiments with very minor changes to the code, which would require a full repetition of the analysis process if undertaken manually. Despite these advantages, the system is only in the early stages of becoming a useful tool to aid analysis in a way that supports design decisions. Ideally a number of system design options would be compared against each other to demonstrate which provides the highest assurance of safety. There are some limitations on modelling the real world with the concepts captured through this technique, such as component response functions. These are programmed as a safety assured response and it is thought that there is little added value at this stage to have a detailed system response for a full range of possible inputs.

RELATED WORK

Related work has been published from Sandia National Laboratories where the combination of Fault Tree Analysis, Event Trees and Finite Element Models has been described in (Carlson and Jones 1998). Failure modelling techniques exist with a similar concept of component specifications and responses. The Failure Propagation and Transformation Notation (FPTN), (Fenelon and McDermid 1993), provides a notation for capturing component responses and analysing ways through which a component failure can propagate between components. This has been developed further into a calculus (Wallace 2005), where different types of transfer of energy can occur (source, sinks, transformation or propagation). Our tool adds the path generation aspect to these existing methods, but utilises the existing concepts for propagation of insults. The methodology under development is expected to provide much more functionality for analysis of the system and to aid the whole design process.

CONCLUSIONS

The issues of speed, completeness and reliability of safety analysis with manual process can be improved through the use of software tools. Case study evidence has demonstrated that it is possible to analyse all paths through a system and identify which are of concern in a short time scale using the Matlab tool. This aids the design and analysis process of potential system designs, and future work aims to provide a full methodology to support this along with tackling some of the highlighted issues with the current tool.

REFERENCES

Davis, J. 2000. " Integrated Safety, Reliability, and Diagnostics of High Assurance, High Consequence Systems." Doctoral Thesis

Johnson, C.R. 2009. "Methodology for Designing and Analyzing High Consequence Arming Systems". In *Proceedings of the US Joint Weapons Systems Safety Conference* (Huntsville, Alabama), pages 552-561.

Ministry of Defence. 2008. *JSP538 - Regulation of the Nuclear Weapon Programme.*

Ministry of Defence. 2011. *JSP372 - Approving Authority Management Arrangements for the Trident Re-entry System.*

Cooper, J.A. and J.M. Covan. 1998. "Predictable Safety in the Control of High Consequence Systems". In *Proceedings of the 3rd IEEE High-Assurance Systems Engineering Symposium* (Washington, DC)

Plummer, D.W. and W.H. Greenwood. 1998. "The History of Nuclear Weapon Safety Devices". *In Proceedings of the AIAA/ASME/SAE/ASEE joint propulsion conference* (Cleveland, OH) (Jul)

Elliott, G. 2005. "US Nuclear Weapon Safety and Control." *In MIT Program in Science,Technology, and Society*

Ekman, M.E.; P.W. Werner; J.M. Covan; P. E. D'Antonio; and E. Perry. 1998. "A Thematic Approach to System Safety*" In Process Safety Progress 17:3.* American Institute of Chemical Engineers

Comprehensive Test Ban Treat Organisation. 1996. *Comprehensive Nuclear Test Ban Treaty.*

Zadeh, L.A. 1983. "Commonsense Knowledge Representation Based on Fuzzy Logic." . *Computer, 16(10):61-65*

Kreyszig, E. 2010. Advanced Engineering Mathematics. John Wiley & Sons.

Jungnickel, D. 2008. "Graphs, Networks, and Algorithms" Algorithms and computation in mathematics. Springer.

Schwartz, G.T. 1991, "The Myth of the Ford Pinto Case" Rutgers Law Review.

Carlson, D.D. and T.R. Jones. 1998. "An Integrated Tool Set for Software Safety Analysis". In *Journal of Systems and Software*, 21:279-290

Wallace, M. 2005. "Modular Architectural Representation and Analysis of Fault Propagation and Transformation". In Proceedings of the FESCA 141(3), 53-71

ACKNOWLEDGEMENTS AND COPYRIGHT

3D SIMULATION

REAL TIME 3D SIMULATION TOOL FOR NC SHEET METAL CUTTING AND PUNCHING PROCESSES

Aitor Moreno
Vicomtech
email: amoreno@vicomtech.org

Álvaro Segura
Vicomtech
email: asegura@vicomtech.org

Harbil Arregui
Vicomtech
email: harregui@vicomtech.org

Álvaro Ruíz de Infante
Lantek Investigación y Desarrollo S.L.
email: A.RuizDeInfante@lantek.es

Natxo Canto
Lantek Sheet Metal Solutions
email: N.Canto@lantek.es

May 15, 2012

KEYWORDS

Computer Integrated Manufacturing and Engineering (CIME), Industrial processes, Optimization, Interactive simulation, Real time simulation

ABSTRACT

In this paper we present a sheet metal Numerical Control (NC) simulation tool for cutting and punching processes, considering the internal representation of the sheet metal as a 2D complex polygon. The nature of the involved processes in the cutting and punching operations supports the utilisation of Boolean Operations between 2D polygons with fully established geometrical methods. However, straightforward utilisation of such Boolean Operations to support the material removal process leads to slow simulation times, since complexity of the sheet increases continuously. Some optimizations have been introduced to outperform the simulation times, such as the spatial subdivision and optimized methods to generate directly the swept area in arc movements. Results show that those optimizations are significant and have a direct impact in the simulation performance.

INTRODUCTION

In the machining industry, technological advances have led to increased productivity and business efficiency. Generally, these advances have been higher in mechanical technology, obtaining more efficient machinery, faster and more versatile, leading to better economical results. The management and optimization of resources is one of the most interesting challenges, since a small improvement (in time or resource consumption) results in a real savings in the processes that will produce thousands of parts. In the field of sheet metal cutting it is even more crucial because the very nature of the cutting process will generate wasted material, having to be returned to the melting industry for recycling.
Therefore, the simulation tools help to test and check the programs in the design phase, being verified all the necessary times before they are actually run in the actual machine. After a series of tests, including the optimization modifications, the program can reach an optimized state, good enough to be transferred to production. Any not fully tested Numerical Control (NC) program can cause or increase the risks and compromise the machinery, provoking partial breakages of parts, collisions between different machine parts, the sheet metal and the tool.

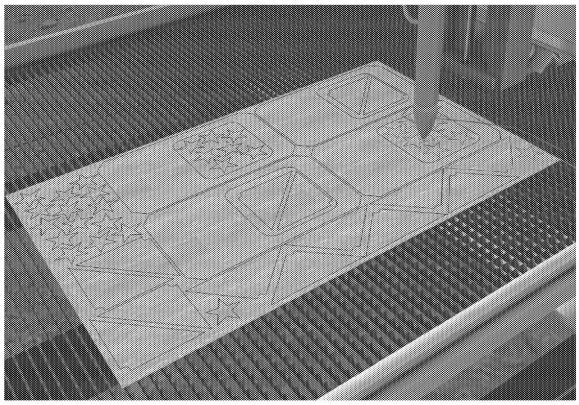


Figure 1: Sheet metal cutting simulation example.

This work presents a NC simulation tool (see Fig. 1) specifically aimed for the virtual representation of the metal sheet main machining processes (see Fig. 2), like cutting and punching. The first part of this work introduces the technological advances in the field, being the basis for the implemented simulation system, presented in the corresponding section. The performance improvements of the proposed architecture are addressed, followed by the conclusions and some future guidelines.

RELATED WORK

The NC machining simulation using Computer Graphics techniques is a widely extended research topic, where

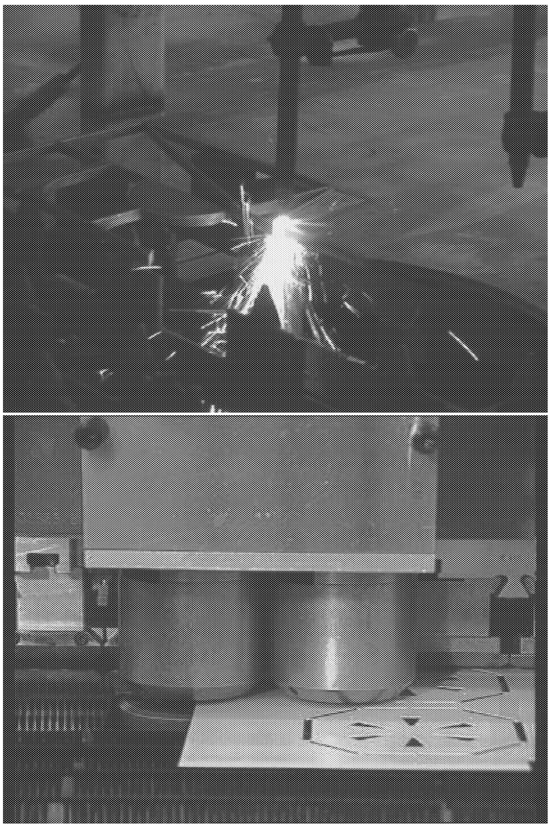


Figure 2: Real sheet metal cutting and punching operations.

the main issue is related to the representation of the dynamic parts of the simulation (in this case, the sheet metal is considered dynamic, as its geometry changes over time). Some traditional approaches do not store the geometrical information during the simulation, but they simply modify the drawing screen using an image-based approach.
Some techniques store the intermediate results in the computer's memory, having an internal 3D geometric representation of the object that is changed continuously during the simulation process. With these methods, a permanent representation is always available and provides free camera movement around the object, better geometric accuracy control, geometric based collision detection, etc.
Van Hook (Hook 1986) used an extended Z-buffer data structure (called a Dexel structure) for the graphical verification. In his work, a scan method to convert surface data into his Dexel (depth element) structure was presented. The Z values for the nearest and the farthest surface at each Dexel are stored in such depth elements. This technique has been extended by several authors (Zhu and Lee 2004).
Other representation methods in the Computer Graphics field are fundamentally geometric like *i)* Boundary Representation (B-Rep) *ii)* Constructive Solid Geometry (CSG), and *iii)* Hierarchical Space Decomposition (HSP).
Although B-Rep is the most used method for solid modelling in modern CAD systems, its straightforward use for machining simulation is not convenient due to the long time required for the dynamic simulations (Spence and Li 2001). A similar problem occurs with CSG representation, with computational costs of order $O(n^2)$, where n is the number of primitives (Stewart et al. 2003) being computationally expensive. A modern implementation of CSG representation through BSP (Binary Space Partitioning) trees has been ported to Javascript and to the Web with a great but non real time performance (Wallace 2012).
To cope with the complexity of the problem and the long time required in these approaches, the approximation of the exact geometry, and especially the partitioning of the object in suitable regions has been proposed by several authors (Stewart et al. 2003).
The most classic technique for volume partitioning is the voxel representation (classical octree, extended octree (Brunet and Navazo 1990), SP-Octree (Cano 2002)) that combines the space partitioning, solid representation and boolean operation support in a single definition. The sheet to be manufactured can be approximated by a very thin extruded plane, given that the machining program is limited to 2D movement over the sheet. This sheet representation provides a direct way to perform boolean operations between the moving tools and the planar sheet. The 3D boolean operation is simplified in a single 2D boolean operation between two 2D complex polygons, that is a well reviewed research topics (Vatti 1992, Preparata and Shamos 1985).
In this work, we present how an efficient and optimized sheet metal machining simulator has been designed and developed, with an internal geometrical core based on complex polygon boolean operations to support the material removal processes.

METHODOLOGY

This work is aimed to develop a simulation system for cutting and punching of sheet metal through a NC controller. The simulator has been developed as a prototype simulation software module and it is focused to obtain consistent geometric results and high graphics quality for the following sheet machining processes (see Fig. 3):

- Sheet metal cutting processes using laser, plasma, oxy-fuel or water jet.
- Punching processes with tools defined with basic or complex shapes.

By means of simulation techniques, this work tries to emulate the behavior of the machine tool to the computer. The simulation system takes as input a starting NC program (normally, G-code dialect) translated to

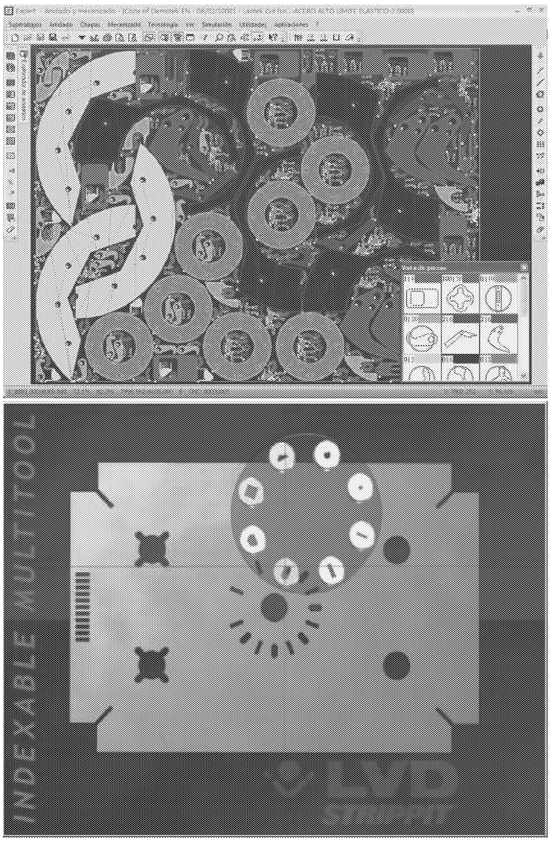


Figure 3: Nesting example for sheet metal machining (Lantek 2012) and a Virtual Punching 2D visualization (LVDGroup 2012).

a common and simplified XML format, listing all the movements that the machine will perform during the cutting or punching operation. The operation mechanisms differ between the cutting and punching process, but essentially, they are based on the removal of material from the sheet, so the internal module for such operations has been designed to be generic for such operations. The main significant difference is that in the cutting process the removal process is continuous while the torch is powered on, whereas in the punching process, the removal process is instantaneous when the punch is triggered.
In the following subsections, the system architecture and the main modules will be described.

System Architecture

The simulator is structured as a multi-layered architecture, each one encapsulating different methods and techniques. The low level layer involves the geometry calculations (Geometric Kernel) with the management of the boolean operations between 2D polygons as its main responsibility. The Clipper library (Johnson 2012) provides the functionality to calculate Boolean Operations between 2D polygons, as the basis for subtraction between a complex polygon representing the sheet and another polygon representing the sweep of a moving tool in a given period of time (either cutting or punching). This low level layer (green box in Fig. 4) provides the access to the graphic system, responsible for rendering the simulation results in the screen. In this work, we have chosen OpenSceneGraph (OpenSceneGraph 2012) as the graphics subsystem, which is used to draw on screen the result of the Boolean operations, the 3D machine models and all the virtual elements in the scene.
The middle layer of the architecture (red box in Fig. 4 provides a conceptual representation of the entities related to the NC machine domain, like Part, Tool, Machine and Axis, including all animation of all the moving elements. Also, in this layer the geometric sweep volume of the moving tools are calculated and passed to the lower level for the actual boolean operation with the sheet metal representation.
Over the cutting and animation layer, we have added the user oriented layer (blue box in Fig. 4), including the graphical interface of the prototype application and management of the XML files:

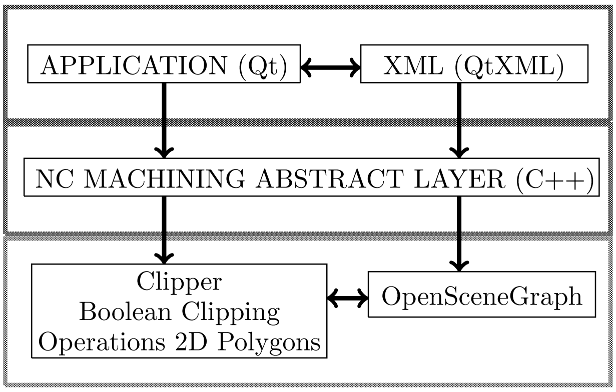


Figure 4: Modular Architecture

- Provide the user interface, including multi-language interface options.
- Implementation of the different navigation methods in the 3D virtual world, using the mouse.
- Manage and display the real NC instructions (highlighting the currently executing instruction) that will be loaded into the simulator through the intermediate and abstract XML representation.
- Manage and display the visualization options of the 3D objects corresponding to the individual parts of the cutting or punching machine.

Both the user interface and the XML parsing functionality have been implemented through the Qt library (blue layer in Fig. 4).

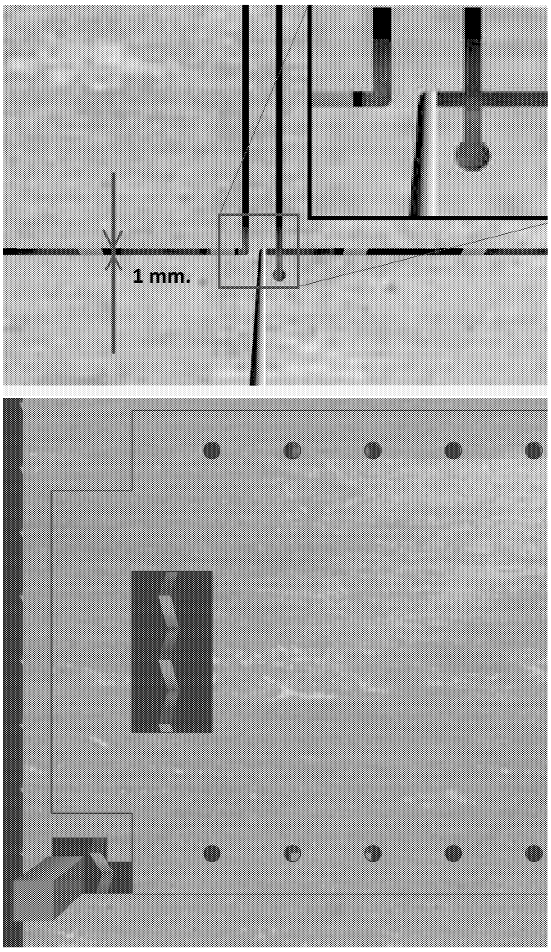


Figure 5: Close up 2D sheet metal machining and punching example.

Listing 1: Pseudo code for the subtraction methods for cutting or punching movements

```
OpBoolCut (in-out Obj sheet , in Mov m)
{
  Obj sweep = GenerateSweep (m)
  BoolOp res = PolyClip2D (sheet , sweep)
  sheet = res
}
OpBoolPunch (in-out Obj sheet , in Mov m)
{
  Obj c = getPunchContour (m)
  BoolOp res = PolyClip2D (sheet , c)
  sheet = res
}
```

Overview of the Sheet Metal Cutting Algorithm

The main elements in the system are the metal sheet and the tools that will translate the machining instructions into geometrical operations. As the tools moves on the sheet, the swept area has to be removed from the sheet, thus, after applying multiple subtraction boolean operations, the sheet will be modified, resembling the expected output in the real world.

The foundation of the methodology is a 2D polygon clipping system. The cutting element or torch (laser, plasma, waterjet, etc.) can be modelled as a cylinder of varying radius (see Fig. 5, top). In a time interval, the moving cutter intersecting the sheet metal sweeps a shape than can be represented by a polygonal contour with curves approximated by sequences of linear segments. In each time step, such swept polygon is subtracted from the polygonal representation of the sheet containing all previously removed contours, setting the result as the new sheet. Additionally, from this new updated geometric representation of the sheet, the graphic subsystem must be updated too, so the final rendering on the screen is updated consequently (see Listing 1).

The geometric representation of punch tools is more complex than the torches, since there are circular, square, rectangular or even arbitrary shapes for punching processes. But the simulation process is simpler as these tools are not activated in continuous mode. A *Punch* instruction activates the current active tool, making a hole in the sheet and returning to its original position. This process can be done multiple times, at a very fast pace, providing similar features to the sheet metal cutting processes (see Fig. 5, bottom).

Cutting and Punching Machine Specification

Cutting and punching machines are essentially different, but at the same time, they have similar components and a generic hierarchical structure can be defined.

The hierarchical structure of the 3D model of the cutting machines starts with the *Table*, as the static part of the machine. Over it, the *Bridge* moves in the X axis. The *Torch Support* is mounted over the bridge as the Y axis. The movements in the Z axis of the machine are performed by the *Torch*, fixed to the support structure (See Fig. 7). Although we have developed several 3D model for the different machines we have considered (oxy-fuel, plasma, laser and waterjet), they all follow the same hierarchical definition.

The hierarchical structure of the punching machine is significantly different from the cutting machine. In the punching machine, the sheet is the mobile element, while in the cutting machine the sheet is static. Therefore, the sheet is moved in the X and Y axis, while the punching stations give the punch movements in the Z axis. The implemented prototype of the punching machine has room for up to 20 individual punching tools. The individual punch geometries are loaded from an XML file.

The visualization of the punching machines requires playing with transparencies in the punching station and the bridge, since they hide completely the punching operation (See Fig. 6).

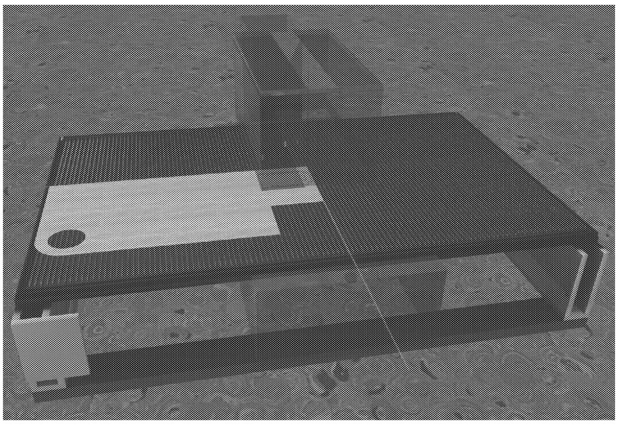


Figure 6: Simplified 3D model for the punching machine with transparent elements to ease the visualization of the punching process.

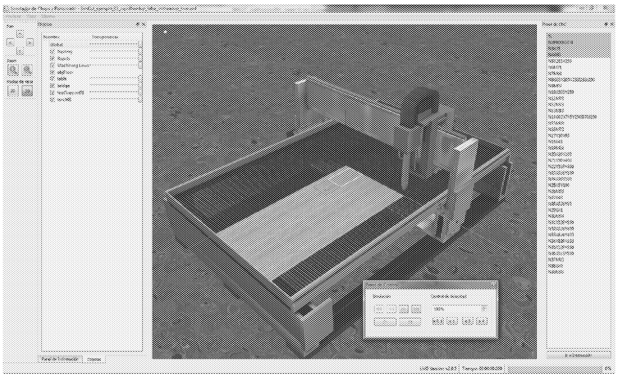


Figure 7: Prototype loaded with a program for cutting a rectangular metal sheet and configured for a single oxy-fuel torch.

User Interface

The user interface displays a simple and clean interface between the user and the simulation functionality. It has been developed using the Qt library, which provides the technology needed to create professional and high-quality graphical interface controls. The application GUI is structured as a set of docking and floating panels, which can be moved freely or docked to any side of the screen (see Fig. 7) with the 3D virtual world always in the central widget.
There are two ways of controlling the virtual simulation. One is by running a continuous simulation, where an animation factor is applied to control de simulation speed. The other is a fast mode, which will run the simulation in background as fast as possible, until the target instruction is reached. Combining both modes, users can go to a specific instruction of the NC code (shown in an independent panel) and from there, begin an animated simulation with the desired speed factor. The rest of the buttons in the Control Panel offer simple VCR functionality: *play and stop*, *play only one instruction*, and go directly to the previous or next instruction, the first or the last instruction of the loaded program.
The GUI provides the visibility options for all the objects in the virtual scene. Any object can toogle its visibility, but not all the elements can modify the transparency, as the lines (machining toolpaths and other helping elements) and the ground model. The camera properties and movements can be set up in another panel, including the toogle button to go to 3D or 2D mode and the Zoom and Pan functionality.
Finally, thanks to the multilingual support from Qt, the interface has been easily translated to several languages, with an easy and portable mechanism to add new languages.

PERFORMANCE OPTIMIZATIONS

As the NC program instructions are known before hand, just when the simulator is started, it is possible to preprocess some information. However, these actions cannot be performed if a full run of the simulation is requested as soon as the simulation is started. In this worst scenario, all the machining instructions must be run in the fastest way, optimizing the resources of the hosting computer. Additionally, the state of the simulator after the final instruction is reached must keep the interactivity of the application for further simulations.
In the following subsections, some applied techniques are presented to try to optimize the overall performance in term of simulation time and other resources consumption such as memory or hard disk.

Direct Contour Generation for Arc Sweeps

The circular movements of the tools were initially implemented by a subdivision into linear piecewise movements. We used a global variable in the system representing the number of subdivisions that a full circle would have. If this value is 32, a full circle movement produces 32 linear movements and similarly, a π radians arc will be decomposed in 16 pieces. The immediate drawback of such subdivision system is that the number of boolean operations grows significantly.
In order to avoid extra boolean operations, we have implemented a direct or native generation method for the arc movements. Using such functionality, the swept contour of a tool is generated directly, and thus, a single boolean operation is performed (see Fig. 8), resulting in a significant performance boost (see Table 1).
The subroutine *AddArcPoints* used in Listing 2, run as *p.AddArcPoints (p1, p2, c, CCW or CW)*, samples a given arc movement going from point *p1* to point *p2*, with point *c* as the center of the arc, in the given *sense*, clockwise or counter clockwise. The resulting sampled points are appended at the end of the *Polygon p*. The

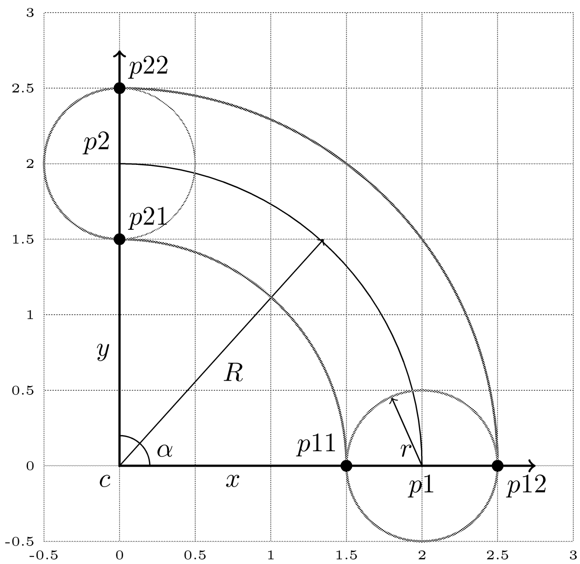


Figure 8: Sweep arc geometric construction for a tool with radius r moving from point *p1* to point *p2* in an arc movement. The sweep contour is constructed by sampling four partial arcs, using the points *p11*, *p12*, *p21* and *p22* as the limit points.

GetCorners subroutine calculates the point coordinates where the individual sectors of the arc are joined.

Listing 2: Pseudo code for the SweepArc function

```
SweepArc (in Point p1, in Point p2,
          in Point c, in float r,
          out Polygon p)
{
  GetCorners (pc11, pc12, pc21, pc22)
  p = Polygon::CreateEmptyPolygon()
  p.AddArcPoints (pc11, p21, c, CCW)
  p.AddArcPoints (pc21, p22, p2, CW)
  p.AddArcPoints (pc22, p12, c, CW)
  p.AddArcPoints (pc12, p11, p1, CW)
}
```

The actual method *SweepArc* is more complex than the pseudo-code shown in Listing 2, as there is a number of cases that should be treated one by one, e.g., when $p1 = p2$ or $r > R$. Additionally, a better consistency with the sense of the arc sampling is desired in order to get geometrically accurate contours for the tool sweep.

Spatial Subdivision

The efficiency of the polygon clipping algorithms depends directly on the total number of contours and points involved in the boolean operation (Leonov 1998):

$$O(n \times \log(n) + k + z \times \log(n)) \quad (1)$$

Table 1: Performance improvement using optimized sweep arc generation. A full simulation was performed with and without the optimization and the number of low level Boolean Operation and the time are presented in the columns *OpBools* and *Time*.

Arc Optimization	Movs	OpBools	Time (s)
No	560	1054	25
Yes	560	774	10

where n is the number of edges (points), z is the number of contours and k is the number of edge intersections.

As the simulation is performed, the working part gets more and more complex and consequently, the number of points and contours grows. In order to limit the number of points and contours that would increase the boolean operation time, a high-level partitioning system is added to the architecture.

This spatial partitioning decomposes the sheet metal into a set of smaller subregions, leading to a high level Boolean Operation pseudo-algorithm (see Listing 3).

The performance effect of the spatial subdivision is limited as an over-subdivided sheet will increase the number of individual Boolean Operations, as any movement will span across multiple regions. The subdivision region is aimed to reduce the complexity of the Boolean Operation (by limiting the number of vertices) but avoiding to increase the mean number of Boolean Operations per movement.

Listing 3: Pseudo code for the Boolean Subtraction between the sheet and the sweep

```
OpBool (in-out Obj part, in Obj sweep)
{
  Set S = SelectRegions (part, sweep)
  for each region R in S }
    BoolOp res = PolyClip2D (R, sweep)
    part.SetRegion (R, res)
  end for
}
```

In our experiments (see Table 2), varying the number of subdivision with the same example, gives a performance peak using the 16×16 subdivision. The Fig. 9 shows the geometric complexity of the metal sheet after the simulation program is completely executed. Approximately, each full circle contains around 100 vertices, 50 for the internal circle and 50 for the external circle. The whole geometric sheet contains around 1 million vertices, and due to the spatial subdivision (16×16 in this case), the rendering is performed in real time using the OpenSceneGraph's internal polygon tesselator (OpenGL's GLUtesselator methods).

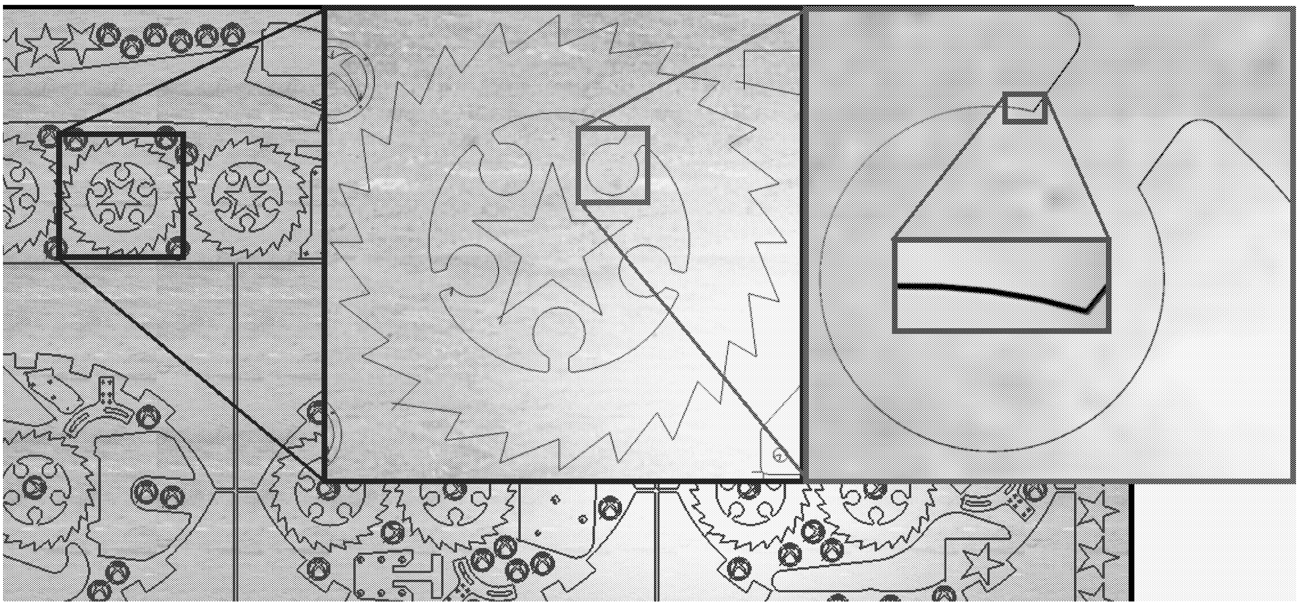


Figure 9: A complex cutting example with more than 12000 movements, combining linear and arc movements. The simulation time is about 90 seconds on an medium range PC.

Table 2: Spatial Subdivision Performance, with the direct sweep arc generation applied, varying the spatial subdivision. The number of low level Boolean Operation and the time are presented in the columns *OpBools* and *T*.

Model	Movs	Subdivision	OpBool	T (s.)
Simple	834	1×1	774	10
		4×4	1008	5
		16×16	2700	6
		32×32	6556	10
Complex	12371	1×1	11037	2100
		4×4	12481	195
		16×16	19533	87
		32×32	36910	130

Saving intermediate states

The VCR functionality enables users to select any desired instruction in the NC program. If the user wants to jump to a non executed instruction, the simulator must run the program till that point in the highest possible speed. But in the opposite way, the simulator should be able to load already calculated simulation points. Unfortunately, in programs with more than 8000 instructions, it is not convenient to save all the intermediate states into memory or even in temporary files in the hard drive, since it would take a lot of resources of the hosting machine.

So, the implemented solution saves a limited number of instructions, acting as keyframes of the simulation. Therefore, when users need to go a previous instruction or, generally speaking, an already calculated position in the CNC program, the simulator will load the previous saved state (the keyframe) and run a small and silent simulation from that point to the target instruction. Of course, the overall performance impact is a matter of balancing the number of keyframes, since a very high number of keyframes will reduce the mean waiting time (the silent simulation step) but will increase the resources usage. With a low number of keyframes, the simulator will use less resources, but the waiting time could interfere with the usability of the simulator, as this seek functionality is widely used to review certain parts of the machining process.

Hardware configuration

The tests and numerical analysis for this Section have been conducted using an Intel Quad Core Q9400 processor, 4GB of RAM and a GeForce GTX 285, Windows 7 PRO 64 bits (Service Pack 1) with the latest stable graphics drivers.

CONCLUSIONS AND FUTURE WORK

In this work we have presented a simulator for the sheet metal cutting and punching processes. Due to the characteristics of the sheet, a representation based on 2D complex polygons has been used to represent the metal sheet. All the operations of the programs have been transformed to internal boolean operations between the sheet and the sweep of moving cylinder (cutting machines) or a complex polygon (for punching machines). As the efficiency of the boolean operations decay with the number of points and polygons, we have introduced several mechanism to optimize the overall result of the simulation. To limit the number of points and polygons, the spatial partition system has been used, checking that it can not be increased to arbitrary numbers, since it will provoke a huge explosion in the number of boolean operations. A balanced spatial partition is required to get the best performance. In the future, a more detailed analysis of such conditions should be evaluated.

The optimizations for the generation of the sweeps in arc movements have provided a boost to the effiency, too. When applied, the number of boolean operations decreases, replacing several operations by just a single one with significant performance improvement. However, the sweep generation can be further improved as the current method samples the arcs independently of the radius of the arc movement and the tool. For example, an optimized version of the function would sample the inner arcs with fewer points than the exterior arcs.

As the sheet metal cutting simulator is intended to be utilised by experts in their field, a fully usable prototype has been developed, providing a GUI to interact with the simulation, displaying all the important information about the NC code, and other features explained in previous sections.

ACKNOWLEDGEMENTS

We thank the Basque Government Industry Department for the financial help received under the GAITEK research.

REFERENCES

Brunet P. and Navazo I., 1990. *Solid representation and operation using extended octrees.* In *ACM Transactions on Graphics 9, 2.* 170–197.

Cano P., 2002. *Representation of polyhedral objects using sp-octrees.* In *Journal of WSCG 10, 1.* 95–101.

Hook V., 1986. *Real Time shaded NC Milling Display.* In *SIGGraph86, Volume 20, Number 4.* 15–20.

Johnson A., 2012. *Clipper - an open source freeware polygon clipping library.* URL `http://www.angusj.com/delphi/clipper.php`.

Lantek, 2012. *Lantek Expert CAD / CAM system.* URL `http://www.lanteksms.com/`.

Leonov M.V., 1998. *Implementation of boolean operations on sets of polygons in the plane.*

LVDGroup, 2012. *LVD - Sheet Metalworking tools.* URL `http://www.lvdgroup.com/`.

OpenSceneGraph, 2012. *Open source 3D Graphics API over OpenGL.* URL `http://www.openscenegraph.org/`.

Preparata F.P. and Shamos M.I., 1985. *Geometry: An Introduction.* Springer-Verlag. ISBN 0-3879-6131-3.

Spence A.D. and Li Z., 2001. *Parallel processing for 2-1/2D machining simulation.* In *Proceedings of the sixth ACM symposium on Solid modeling and applications.* ACM, SMA '01. ISBN 1-58113-366-9, 140–148.

Stewart N.; Leach G.; and John S., 2003. *Improved CSG Rendering using Overlap Graph Subtraction Sequences.* In *International Conference on Computer Graphics and Interactive Techniques in Australalasia and South East Asia.* 47–53.

Vatti B.R., 1992. *A Generic Solution to Polygon Clipping. Communications of the ACM*, 35(7), 56–63.

Wallace E., 2012. *Constructive solid geometry on meshes using BSP trees in JavaScript.* URL `http://evanw.github.com/csg.js`.

Zhu W. and Lee Y., 2004. *Product prototyping and manufacturing planning with 5-DOF haptic sculpting and dexel volume updating.* In *Haptic Interfaces for Virtual Environment and Teleoperator Systems.* 98–105.

EXPERIENCE WITH FULL 3D NUCLEAR REACTOR MODEL

Janos Sebestyen JANOSY **Jozsef PALES**
Janos.S.Janosy@energia.mta.hu Jozsef.Pales@energia.mta.hu
Centre for Energy Research, Hungarian Academy of Sciences, P.O. Box 49, H-1525 Budapest, Hungary

KEYWORDS: Nuclear engineering, Thermodynamics, 3D simulation, Real-time simulation, Interactive

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 in between. 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 the Simulator Development Department (Hazi et al., 2001). KIKO3D - Neutron Kinetics 3D - has been developed in our Institute, too, in the Reactor Analysis Department (Kereszturi et al., 2003). Both of them should be integrated into our full-scope replica simulator, coupled, and stressed to operate parallel in real-time, using four high-power processors (Hazi, Kereszturi et al., 2002). Details of these models were presented on ISC 2010 Conference in Budapest (Janosy et al., 2010).

Verification and validation procedures of nuclear reactor models are extremely difficult because the lack of experimental data. Since the Chernobyl accident no experiments with nuclear power generating units are encouraged; practically data recorded during the normal operations – startup, normal operation, shutdown - are available only.

The simulation-specific details of different experiments with the model are discussed in the paper.

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 bars 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, most of the 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.

3D MODEL REQUIREMENTS

Earlier we could use simpler models with great success (Janosy 2003, Janosy 2007, Janosy 2008). Now we have 349 very different 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 required time step for the accurate numerical integration can be very different, too.

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.

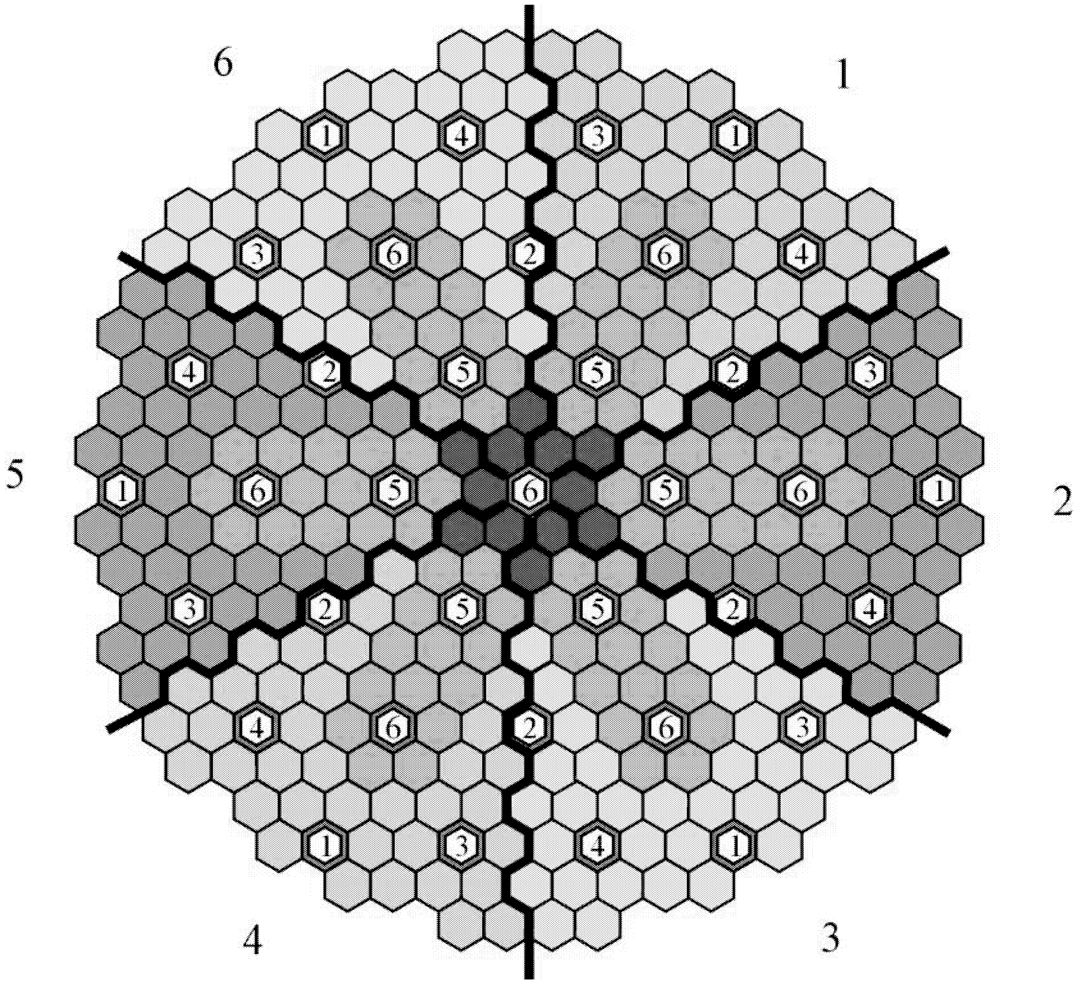


Fig 1. The map of the core with the 349 fuel assemblies, including the 37 control ones.

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 Pic. 1, 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 a 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 important 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. 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 particular assembly should be a separate node – we are going to have as much as 349 nodes horizontally.

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***.

NODALIZATION: THERMOHYDRAULICS

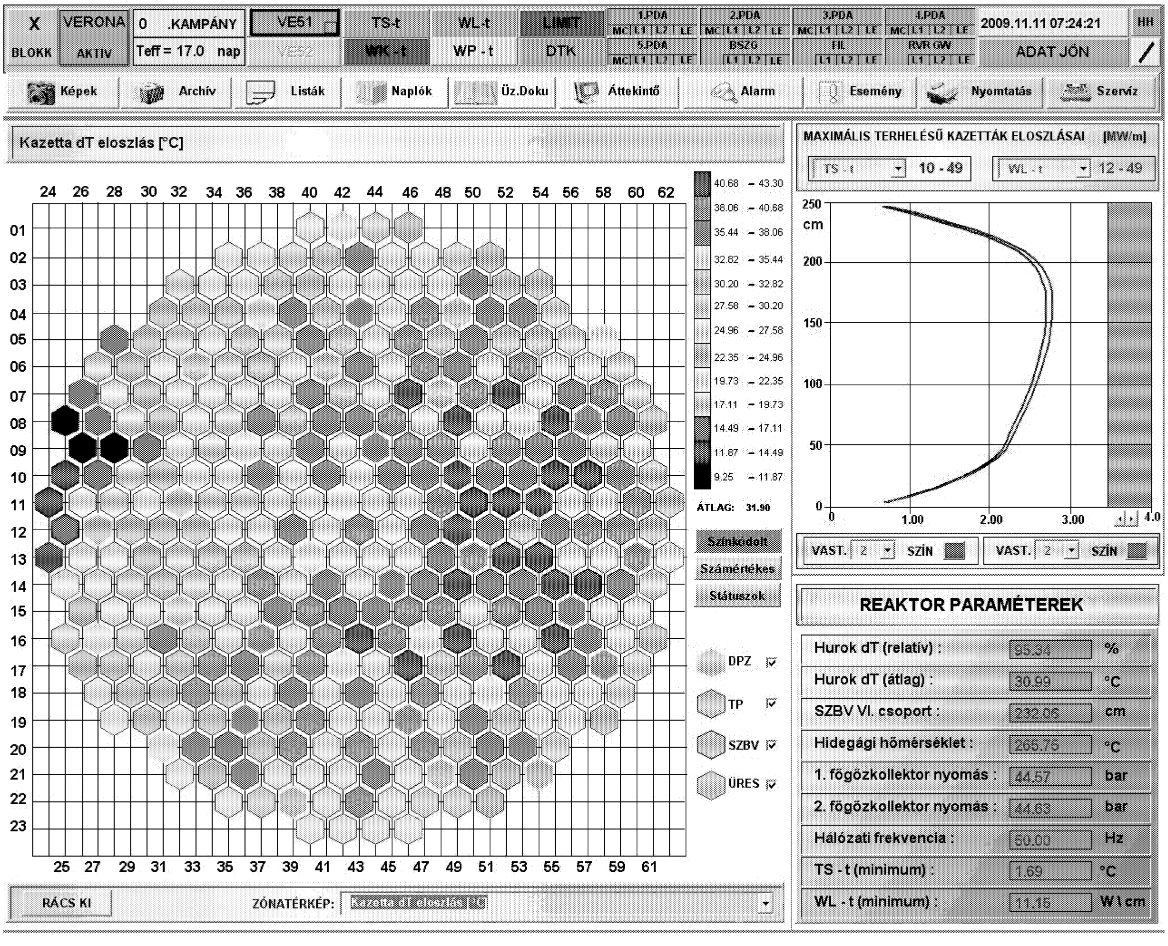
Thermo-hydraulic nodes should be much larger in space than the neutron-kinetic nodes. It is connected with the 0.2 sec. time step of the full scope replica simulator of the power plant. If we want to avoid large number of iterations, the amount of the steam/water leaving/entering the node each time step 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. Creating relatively large nodes we have to group fuel assemblies very carefully, to get as detailed simulation results as possible. The color coding of Pic. 1 shows our results.

The central (red) node on Pic. 1 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 each (one form Group 5 one form Group No. 6). The peripheral six large nodes, shown in different colors, contain 40 fuel assemblies each. Vertically we divide the core into 5 termo-hydraulical layers; it is easy to fit them with the 10 layers of the neutron-kinetical model.

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.

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.

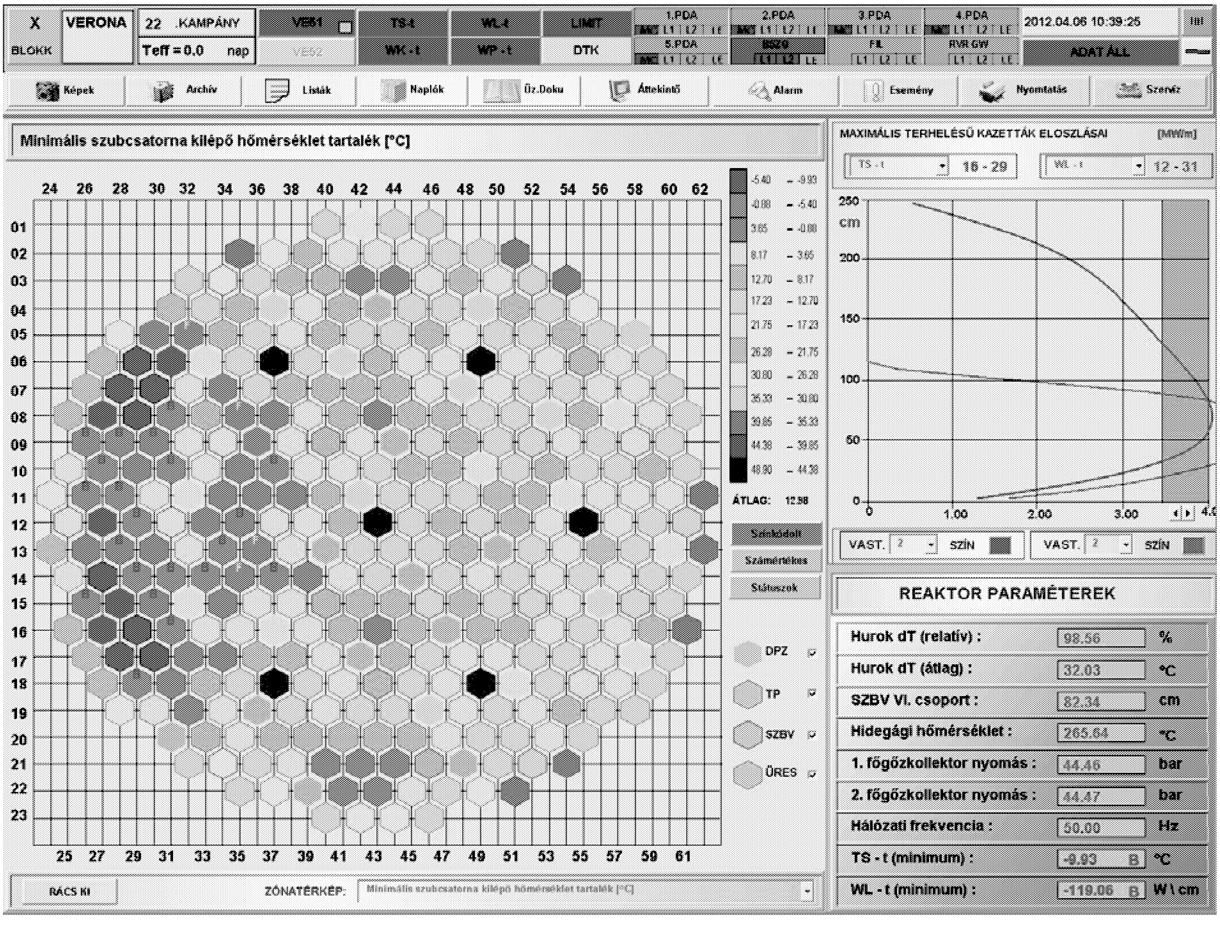


Pic. 2. Picture on the in-core surveillance system VERONA - driven by "rod drop" state data from the simulator

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: overheating causes negative reactivity thus decreases the heat power, and overcooling does the opposite - it leads to positive reactivity and the power increases. This compensates the negative reactivity introduced by the fallen rod and that's why the distortion of the power field - and the resulting temperature field – see Pic. 2. - is not so strongly distorted than it could have been expected. The details about out RETINA thermo-hydraulic model can be found in (Janosy et al. 2010, Hazi 2001, Hazi & Kereszturi, 2002)

Another experiment with the coupled spatial neutron kinetic and thermo-hydraulic models can be seen on Pic. 3. The controller keeps the power permanent moving the control rods. First we inserted some boron acid solvent to get all rods pulled out. (Boron acid is an absorber for the neutrons participating in the chain reaction). Next, we addressed the "control rod stuck" malfunction to the leftmost control rod. It will not move any more, remains in "pulled out" condition. Now extracting slowly the boron acid the controller has to move all the other control rods down, in order to keep the power constant. The restructuring neutron power distribution results in a serious overload at the left part of the core causing intensive boiling. The plant is not permitted to operate under such conditions: these situations can be studied on the simulator only.



Pic. 3. Six rods are "in", the leftmost 7th is "out". Note the overloaded and distorted axial distribution on the right side.

VERIFICATION AND VALIDATION PROBLEMS

The cornerstone of all simulation is the verification and validation method used for the constructed models. In our case the ***verification*** means that the computer program representing our model is error-free and calculates the numerical solution of that differential equation system exactly that has been described in the Technical Design document During ***validation,*** the exact value of the rather numerous free parameters of the model should be determined optimally in order to achieve the highest confidence of the model system - ensuring the best similarity to the modeled real object.

The validation is much more difficult than the verification process. The problem lies in the lack of proper experimental data. No experiments are allowed to perform on the real plant, and only the transient recordings of the **a**nticipated **o**perational **o**ccurrences (AOO) are available. This term stands for all events which are common: start-up, shutdown, operation on different power levels, control of the frequency of the electrical network, turbine start-up and trip, pump trips, synchronization of the electrical generators, physical measurements verifying the parameters of a new core after re-fueling, etc. Careful investigation of the archives provided by the plant surveillance computer systems after a given transient produces great amount of important information.

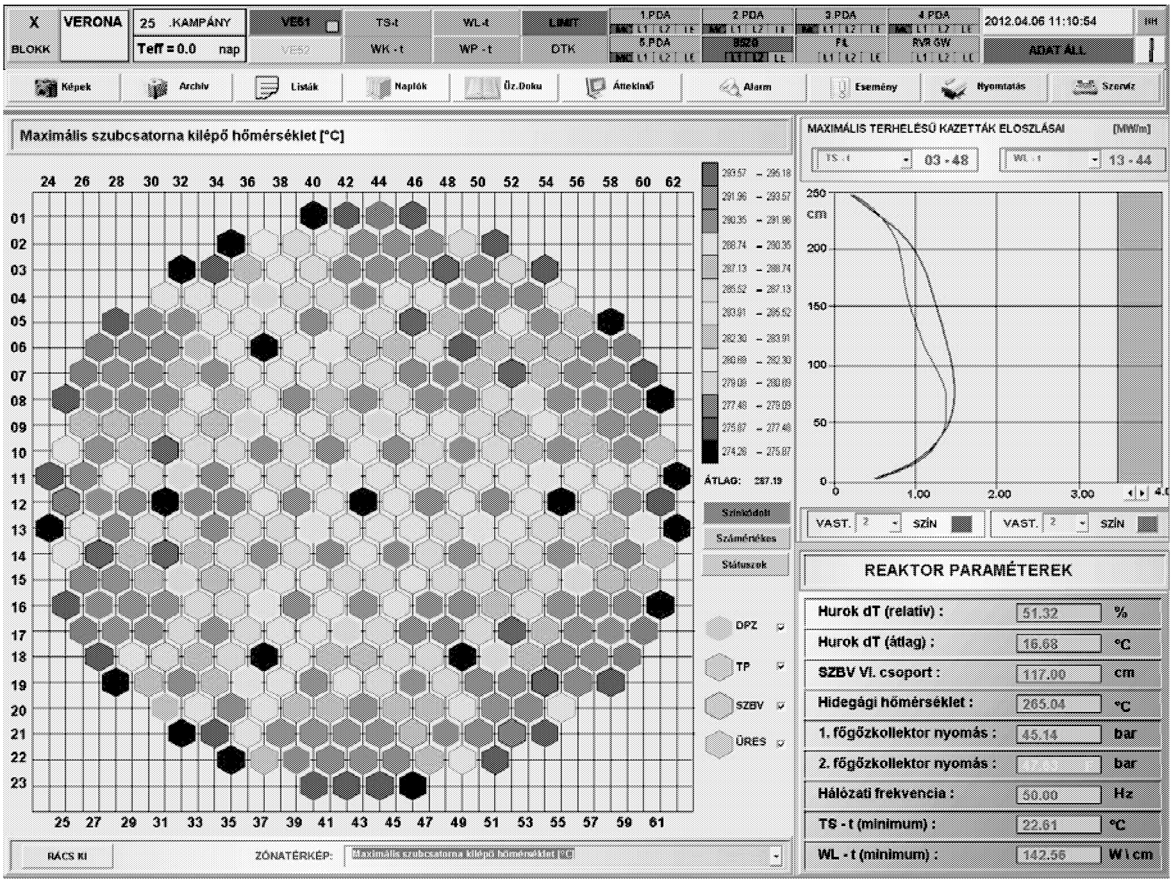
There are very well elaborated models and programs verified and validated by numerous and expensive experiments performed on dedicated experimental facilities. These models are used mostly to prepare the Safety Report of the given plant for the national authorities. The basic problem of the comparison of a training simulator models with them that the training simulator is based on the "best estimate" philosophy, while the programs mentioned above and used to evaluate the actual safety are based on the "worst case" methodology and scenarios. This difference becomes significant when the models have to handle uncertainties. "Best estimate" models usually take the estimated mean value, whereas the "worst case" models take the value which "hurts" most to the safety of the nuclear power plant.

Sometimes simple events help a lot. Turbine trips are not very frequent events, and it is not allowed to operate with only one turbine available. Sometimes, however, it is a simple electrical overload what happens and the protections disconnect the generator in question from the grid. If the operators are able to re-start the turbine and recover the full power soon, they do not shut down the plant. Meanwhile precious data can be collected.

The operation with one turbine is asymmetric. Three steam generators feeding the steam header of the operating turbine remain in a quite similar condition, but the other steam header to which the other three headers are connected remains without turbine. This steam header can feed the other one, but because of the connection losses the pressure of this header becomes higher by 2.5 bars than the other one. Correspondingly, the water temperature on the secondary side of them becomes higher by 2°C. Obviously, those parts of the core fed with coolant of higher temperature reduce somewhat the power (due to the negative temperature feed-back) therefore the output temperatures of these sectors are less warmer than the inlet difference of 2°C.

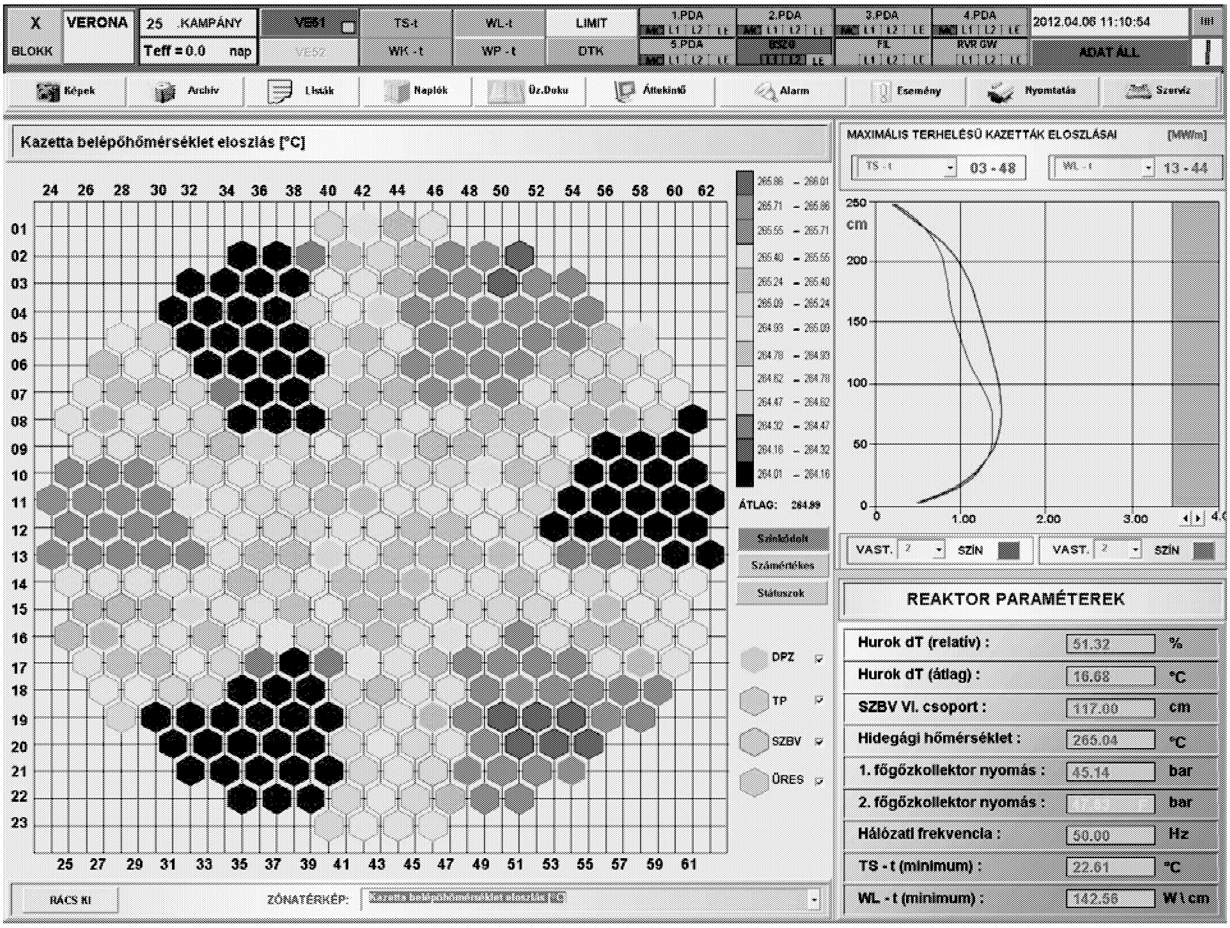
The reactor power controller had to reduce the thermal power of the core to about 50% (51.2% on the VERONA display below), and therefore all the seven control rods are lowered somewhat; the corresponding part of the fuel elements move out and down from the core and the outlet temperature of these channels decrease.

All these effects can be studied on the Pic. 4 below. The temperature difference of the cold legs can be studied on Pic. 5.



Pic. 4. Inlet fuel assembly temperatures calculated by the VERONA system on the basis of the simulator data

Three loops remain practically in the same condition as before (blue color) but the other three inlet temperatures are significantly higher (yellow and red), as it can be seen on the VERONA (Pic. 5):



Pic. 5. One steam turbine is out, and three from the six loops have higher cold loop (i.e. reactor inlet) temperatures.

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.

The real trouble we have always to encounter: thanks to the growing computer power, even if we are trying to formulate as detailed models as possible, it is very difficult to obtain useful measurement data from the real plant to compare our models with.

The standard instrumentation existing on the plant - being fully satisfactory to control the operation and ensure safety – is usually inadequate to compare with the results of 3D calculations made using elaborated model systems.

REFERENCES

J.S. Janosy, A. Kereszturi, Gabor Hazi, Jozsef Pales, Endre Vegh, **2010**: Spatial Power Distribution Simulation For Nuclear Reactor, Industrial Simulation Conference ISC 2010, June 7-9, Budapest, Hungary, ISBN 978-90-77381-5-57, pp. 239-244

J.S. Janosy, 2003: 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

Janos Sebestyen Janosy, **2007**: 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

Janos Sebestyen Janosy, **2008**: 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.

A. Kereszturi, Gy. Hegyi, Cs. Maraczy, I. Panka, M. Telbisz, I. Trosztel and Cs. Hegedus, **2003**: Development and validation of the three-dimensional dynamic code - KIKO3D, Annals of Nuclear Energy 30 (2003) pp. 93-120.

Gabor Hazi, Gusztav Mayer, Istvan Farkas, Peter Makovi and A. A. El-Kafas, **2001**: "Simulation of a small loss of coolant accident by using RETINA V1.0D code", Volume 28, Issue 16, November 2001, Pages 1583-1594

Gabor Hazi, Andras Kereszturi, Istvan Farkas, Gusztav Mayer, Gyorgy Hegyi and Istvan Panka, **2002**: "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

ELECTRONICS SIMULATION

VERIFICATION OF INITIALIZATION SEQUENCES FOR SEQUENTIAL CIRCUITS BY USING DEPENDENCY MATRIXES

Kestutis Morkunas, Rimantas Seinauskas
Software Engineering Department, Kaunas University of Technology, Lithuania
Email: kestutis.morkunas@ktu.lt
ITPI, Kaunas University of Technology, Lithuania
Email: rimantas.seinauskas@ktu.lt

KEYWORDS

Sequential circuits, initialization sequences, partial reset and initialization, dependency matrixes.

ABSTRACT

This article suggests an approach to the verification of initializing sequences for sequential circuits. A software prototype emulating an actual circuit is used. Using a software prototype allows for an early test case generation. Initialization sequences search and multiple verification methods are explained. Approaches like using binary and ternary logics, matrixes, and random search are discussed, results are compared and differences are explained. Other researchers experimental results based on different methods yet same circuits are included as well. Results are based on ISCAS'89 benchmarks. Experimental results show, as-good-as, or better results for same benchmark circuits compared to other methods.

INTRODUCTION

Testing is the largest expense in circuit manufacturing. Bad tests, unspotted defects lead to releasing bad quality chips. Bad chips lead to annual repair, retesting, field failures and returns. Therefore, testing is one of most important steps in manufacturing process. In this article we discuss verification of initialization sequences using dependency matrixes. A set of inputs, called an initializing sequence, switches system from any unknown state into a single fixed state. This is a starting point for testing. Once a system is switched into a fixed state, other inputs might be used to check transitions between states. Initializing sequences can be difficult to find or non-existent [3]. Two methods may be used for setting flip-flops in sequential circuits. First is using input patterns to switch flip-flop values from unfixed to fixed ones. There is another method, using DFT, or design for testability method, directly pushing required values into flip-flops. Mentioned methods have their advantages and disadvantages [9]. Such initializing sequence may be too long, not available at all or switch the system into an undesired state [1], [4]. DFT and scan techniques allow easy flip-flop access, yet it increases chip size. With it grows power consumption, heat, signal noise and price. Extra connection lines are introduced and breaks in them may result in failed tests.

Search for initializing sequence is a tricky task [14]. Full state and transition vectors are needed to find fully reliable sequence. Large circuits produce huge state-transition trees and difficult to make with many flip-flops present [2]. A circuit having 3 inputs and 3 triggers has a state space size of 8 and 8 inputs. 64 computation cycles are needed to test state-to-state transitions for all input/state combinations. A circuit having 14 inputs and 6 triggers would result in 1'048'576 computation cycles. A circuit from ISCAS'89 numbered s35932 would require 5,20e+530 computation cycles for full test. Therefore, this approach is not practical. It required huge computational resources and might not be possible with large circuits.

PREVIOUS WORK

Random search algorithm was used to find initializing sequences. These sequences are used in verification process after they are found. Software emulator was written to mimic the actual circuit's operation. No actual hardware was used. Emulating prototypes, based on ISCAS'89 circuits, were translated from Verilog source files.

Starting test generation early in the manufacturing process had many benefits: early design flaws discovery, some tests being ready upon circuit-on-chip manufacturing stage is done. Testing stage becomes partially concurrent with other processes. This saves time, as testing normally would start after the chip is made and ready to be tested.

Conditions are more difficult when using emulating software prototypes, as there is no knowledge of the inner circuit wiring and elements [10]. Lines, elements, gates, and flip-flops cannot be analysed, joined into groups or removed as suggested by other researchers. Experimental results in this paper show emulating-software-prototype based random search method to be as good as or better compared to similar tests [6], [7], [8].

At the beginning of our experiments, verification of initializing sequences was based on heuristics. Main idea behind it was: searching for an initializing sequence. Input patterns switch a system to a fixed state from small number of starting states (20 states for speed of computing); if such a sequence is found, a large set of random starting states is generated (50,000 states for most circuits, less for smaller ones). In the verification process this found sequences is used against large set of starting states. If it switches the system from all 50,000 random starting states into a single fixed state, we state, that this might be truly a correct and reliable initializing sequence. There is a good chance it will do so with larger/any number of states as well.

RANDOM SEARCH FOR AN INITIALIZING SEQUENCE.

Random search algorithm for an initializing sequence candidate consists of sets generation. First, small set of random starting states (20 in experiments) is generated. Then (100 in experiments) random input patterns are generated. These numbers are based on other experiments on how many states/inputs are sufficient to produce viable initialization candidates and reduce false-positives. It also shows how many calculations are needed for input/states space transition to be calculated. To check 100 inputs against 20 starting states, 2000 calculation cycles are needed.

State-to-state transitions are calculated. After all states are used and transitions calculated, the analysis part starts. Each flip-flop's value is checked for each state. If there is the same value for same flip-flop on different states, then there is a fixing of that flip-flop.

After checking all input patterns, the one that fixed most flip-flops is added to the end of initializing sequence, which may consist of 1 or more patterns. In our experiment the maximum length was limited to 50 input patterns.

The process is then repeated. New input patterns set is generated, but old state set is used. Old space state consists of partially initialized flip-flops by previous input patterns.

If the number of fixed flip-flops does not increase with new input patterns or the maximum length of initializing sequence is reached, the process is stopped [12].

Once initializing sequence candidate is found, verification starts. Only one set of inputs are used – those from the candidate. A large set of starting random states is generated, which was 50,000 in this experiment. This number is a small amount of all possible states for larger circuits. Therefore, there is no fully guaranteed result using this approach. This might be corrected by using dependency matrixes.

DEPENDENCY MATRIXES AND VERIFICATION OF INITIALIZING SEQUENCES

Dependency matrixes define circuit's inputs, outputs and flip-flops relationships. Analyzing such matrixes allows identification and ranking of flip-flops that have great influence for values of other flip-flops.

Main task for initialization is switching system's flip-flops into fixed state. Flip- flop values depend on input patterns and values of self or other flip-flops. If a flip-flop depends on many other flip-flops for its value, this means that all "parent" flip- flops must switch into a fixed state before a "child" flip-flop becomes fixed.

Information obtainable from dependency matrixes: list of inputs affecting each flip-flop, list of flip-flops affecting each flip-flop, list of inputs affecting each output, list of flip-flops affecting each output, list of inputs and/or flip-flops affecting a set of outputs/flip-flops. Ranking of flip-flops, putting least dependent on other flip-flops (and easiest to initialize) on top of flip-flop dependency tree.

S27 is a circuit from ISCAS'89 set of benchmarks. It consists of 4 inputs, 1 output, and 3 flip-flops. Table 1 contains its dependency matrix.

Table 1. Dependency matrix of s27 circuit

Flip-flop number	TO0	TO1	TO2
TI0	0	0	
TI1		1	
TI2	2	2	2

Table 1 displays flip-flop space and dependencies between them. Flip-flop value depends on its own value and other input's and other flip-flops' values in most cases. Some flip-flops may only depend on self and input values, not other flip-flops, as is the case with flip-flop 1. TI marking means starting state of flip-flop, where TO means resulting state of slip-flop.

Flip-flop 1 depends on itself and inputs, while flip-flop 0 depends on itself and flip-flop 1. Flip-flop 2 depends on itself and two other flip-flops {0;1}

This matrix knowledge was used to produce a flip-flop dependency tree, displayed in Table 2.

Table 2. Circuit's s386 dependency tree

Flip-flop number	Dependency set
T0	{0;1}
T1	{1}
T2	{0;1;2}

Such trees may be produced for most of the ISCAS'89 circuits. It basically show which flip-flops should be tested first in the verification process. Once these most influential flip-flops are set, others should be easier to fix.

If the influence group is too big, where all/most of flip-flops influence many other flip-flops at the same time, such circuit may be un-initialisable or hard-to initialize.

Such dependency matrixes are often expanded to include input/output pins. Analysing those dependencies allows identify most influential input pins, which might help reduce the set size of inputs or craft those that will have biggest impact on required flip-flops.

Generation of a full set of states may not be possible for larger circuits [13]. It might be possible to generate such sets for selected "problematic" flip-flops. If an initializing set is not found for a full set for these "problematic" flip-flops, there is a chance such a sequence does not exist. Circuit s953 was tested under such circumstances. It has 16 inputs, 23 outputs and 29 flip-flops. Flip-flops numbered 0..5 affects all others. The rest affect only outputs. Next states of mentioned 6 flip-flops are calculated from input pattern and their previous states. All of them tightly influence each-other.

B06 circuit has 4 flip-flops influencing the other 5.

This method will not work for circuits where flip-flops are tightly interconnected. Identifying "leading" ones is difficult, as for circuit s386. It's all 6 flip-flops depend on each other.

Once a dependency tree is available, as displayed in Table 2, it is possible to start small-sets verification:

1. Check if all “parent” or influencing flip-flops on a given flip-flop is already switched into a fixed state.

2. If so, generate all possible states for that tested flip-flop.

3. If a pattern switches flip-flop into a fixed state from all possible starting states and all of its “parent” flip-flops are fixed it is guaranteed, that this input pattern is fully reliable.

4. Check all flip-flops going from most “influential” ones to least “influential” ones.

Failed verification is not always a bad outcome. I.e. initializing sequence claims to have set 980/1000 flip-flops into a fixed state. Verification approaches confirm 970/1000 fixes. It might still be an acceptable result.

RESULTS OF INITIALIZATION SEQUENCES VERIFICATION PROCESS.

Experimental results were compiled using binary logic, simulated ternary logic and hardware ternary logic from other research papers. Results provided in Table 3.

Table 3. Number of initialized flip-flops using various verification approaches

Circuit	Num	Fixed usin	Set using	Other researche rs	Seq
s13207	638	477	432	314	15
s1423	74	74	74	74	3
s15850	534	458	458	306	18
s298	14	14	14	14	2
s38584	1426	1423	1423	1398	36
s386	6	6	6	-	2
s526	21	21	21	21	2
s5378	179	167	163	179	9
s9234	211	154	154	152	6
s953	29	25	10	13	1

Tests on manufactured hardware circuits provide most reliable results. Basic verification using ternary logic provides good results, yet it uses logic that is too strict (X values instead of lessened state space). Basic binary testing is based on heuristics and does not provide guaranteed answer. Binary logic approach coupled with matrixes and dependency-trees tend to provide most reliable results while experimenting using software circuit-emulators instead of actual manufactured circuits. This approach use full sets of states for selected flip-flops and can guarantee result correctness.

CONCLUSIONS

Dependency matrixes and trees does improve verification results for initialization sequences of sequential circuits. As generation of full sets of starting states for all flip-flops is not practical for large circuits, other approaches are used. Inner circuit layout and structure information is not available; therefore search and verification algorithms operate under more difficult conditions. Analysing the dependencies allows identifying “influential” flip-flops, that greatly affects other flip-flop values and targeting them specifically. By analyzing and splitting flip-flops into “parent” and “child” groups allow easier generation of states spaces, easier testing and verification of found initializing sequences. It also enables guaranteed results instead of those based only on heuristics.

Experimental results show as-good-as or better results compared to other researchers’ results based on same ISCAS’89 circuits.

REFERENCES

1. Pomeranz I., Reddy S. M.
On the Detection of Reset Faults in Synchronous Sequential Circuits. VLSI Design 1997. 1997, pp. 470-474.
2. Cheng K., Agrawal V.
Initializability consideration in sequential machine synthesis. IEEE Trans. Comput.1992, volume 41, pp. 374-379.
3. Wehbeh J. A., Saab D. G.
On the Initializationof Sequential Circuits. Intl. Test Conf. 1994, pp. 233-239.
4. Pomeranz I., Reddy S. M. On Removing Redundancies from Synchronous Sequential
Circuits with Synchronizing Sequences. IEEE Trans. Jan. 1996, pp. 20-32.
5. Keim M., Becker B.
On the (Non-)Resetability of Synchronous Sequential Circuits. IEEE VLSI test symposium. 1996, pp. 240-245.
6. Xiaojing H., Zhengxiang S. Ant Colony Optimizations for Initalization of synchronous ssequential circuits, IEEE Circuits and Systems International Conf. 2009, pp. 5-18.
7. Corno F., Prinetto P.
Initializability analysis of synchronous sequential circuits. ACM Trans. on Design Automation of Electronic Systems. 2002, vol. 7, no 2, pp. 249-264
8. Lu Y., Pomeranz I.
Synchronization of Large Sequential Circuits by Partial Reset*. IEEE VLSI Test Symp. 1996, pp. 93-98
9. Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.
Functional delay test generation based on software prototype. Microelectronics Reliability 49(12), 2009, pp. 1578-1585
10. Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.
The Use of a Software Prototype for Verification Test Generation. Information Technology And Control, Kaunas, Technologija, 2008, Vol. 37, No. 4, 265 - 274
11. Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.
On the Enrichment of Functional Delay Fault Tests. Information Technology And Control, Kaunas, Technologija, 2009, Vol. 38, No. 3, 208 – 21
12. Morkūnas K., Šeinauskas R.
Circuit Reset Sequences based on Software Prototypes. Electronics and electrical engineering, Kaunas, Technologija, 2010, No. 7(103).
13. Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.
Functional fault models for non-scan sequential circuits. Microelectronics Reliability 51(12): 2402-2411 (2011)
14. Karim Mohammadi, Hadi Jahanirad, Pejman Attarsharghi
Fast Reliability Analysis Method for Sequential Logic Circuits. ICSEng, 2011, p. 352-356

USAGE OF SIMULATORS IN TESTING SYSTEM

Ludek Dolihal
Tomas Hruska
Karel Masarik

Faculty of Information Technology,
Brno University of Technology,
Božetěchova 1/2, 612 66 Brno,
Czech Republic
E-mail: idolihal@fit.vutbr.cz

KEYWORDS
HW/SW codesign, testing system, embedded systems, Codal language, simulation, compiler testing, library porting.

ABSTRACT

The Lissom project (Lissom) (commercialized under the registrated mark Codasip - www.codasip.com) is focused on the development of tools for hardware/software (HW/SW) codesign. Amongst the tools that are automatically generated is a C compiler. The C language is still one of the most often used languages for programming of the embedded systems. For testing of these C compilers for various platforms we use different simulators. This paper will describe the way of generation of the compiler and the simulator and also the process of testing that is carried out by the simulator, the phase of test selection and test evaluation. We will give an overall description of testing system build on the simulators.

INTRODUCTION

One of the goals in our research group is an automatic generation of C compilers for various architectures. In the past we worked for example on Microprocessor without Interlocked Pipeline Stages (MIPS). Currently we provide startup platforms such as Codea1 and Codea2. To minimize the number of errors in the automatically generated compilers, it is necessary to put the generated compilers under test. Because the whole process of the compiler generation is highly automatic and we do not have all the platforms, for which we develop, available for testing, we use simulators for compiler testing instead of the chips or development kits. In case of our startup processors it is not even possible to use development kits. If one wants to test the C compiler within any simulator, it is necessary to add the support for the C library functions into the simulator, which is used for the testing.

Hence the simulators play crucial part in the whole process of testing. Given that we have a sets of tests we want to compile and execute we need to customize the simulator in certain ways. One of them is for example modifications made in favor of receiving exit values of the running program. Another of the changes is forced by the syscall mechanism.

The paper is organized in the following way. Second section provides the position of the testing in the Lissom project. After that we give the short overview of related. Next section describes the levels of description we recognize in our project. Section six discusses the addition of the library support into the simulator. In section seven is the description of the process of testing. The final section concludes the paper.

POSITION IN THE LISSOM PROJECT

As mentioned before the Lissom project is focused mainly on hardware software codesign. In order to deliver the best possible services we want to provide the C compiler for a given platform as the C language is still one of the main development languages for embedded systems. The C compiler is automatically generated from the description file. Besides the C compiler there are a lot of tools that are also generated from the description file. The tools include mainly:

- simulators,
- assembler,
- disassembler,
- profiler,
- hardware description.

The simulators can be generated either from a cycle accurate or an instruction accurate model. The profiler was thoroughly described in the article (Prikryl et al. 2010).

The model is written in ISAC (Masarik et al. 2006) language. The ISAC language is an architecture description language (ADL). It falls into the category of mixed ADL.

We would like to produce the whole integrated development environment (IDE) for hardware software co-design. This IDE called Codasip studio, should provide all the necessary tools for developers when designing embedded systems from the scratch. The simulator is part of the IDE and C library support is part of the simulators.

The tool for generating compilers is called *backendgen* and is also embedded in the IDE. The quality of a compiler is crucial for the quality of software that is compiled by compiler. Hence it is very important to test the compiler that is generated by the *backendgen*. Via locating errors in the compiler itself we can afterwards identify and fix problems in the generation tools and in the whole development process.

The *backendgen* closely cooperates with the semantic extractor. The semantic extractor as the title suggests, extracts the semantics of the instructions specified in the ISAC file and after that the *backendgen* creates backend of the compiler that recognizes given instructions. The primary role of the C library is to enlarge the range of constructions that can be used during the process of testing. It is without all doubts important to test the basic constructions such as if-statement, loops, function calls etc. On the other hand it is highly desirable to have a possibility of printing outputs or exiting program with different exit values and this can not be done without a C library support. The exit values are the basic notification of program evaluation and debugging dumps are also one of the core methods of debugging. Note that all the tests are designed for the given embedded system, and the tests are run on the simulator.

Secondary role of the library in the whole process of development is providing additional functions for writing programs. One of the most used functions is a group of functions used for allocating memory, string comparison and parsing, input/output methods etc.

As it is possible to generate several types of simulators in the Lissom project, it will be necessary to add the library support into all types of simulators. It should not include any substantial changes to the process of generation.

RELATED WORK

As the core of the paper is dedicated to the testing of the compiler in the simulator we will focus mainly on related work in this area.

Simulators in general are one of the most popular solutions as far as embedded systems development is concerned. They are very often used for testing. We tried to pick up several examples that are connected to embedded systems development, and were published in a form of article. The Unisim project that will be mentioned in this section is not aimed at embedded systems but provides interesting idea.

In (Onder and Gupta 1998) is suggested a system that is very similar to the one that is developed within our project. It is called Upfast. The article describes system that generates different tools from a description file such as we do. The article mentions that C libraries were developed, but no closer information is given. It seems that in the simulator of the Unisim project the support for C language library have been right from the beginning. Unfortunately this is not our case. Porting of the library is critical for us, because without the support it is very difficult to test and evaluate the results of any tests.

Another interesting system including simulator is described in (Hughes et al. 2002). The project is called Rsim and is focused on simulation of shared memory multiprocessors. The Rsim project works under Solaris. The Rsim simulator can not use standard system libraries. Unfortunately it is not explained why. Instead the Rsim provides commonly used libraries and functions. The Rsim simulator was tested for support of C library. All system calls in the Rsim are only emulated, no simulation is performed. In our system we will simulate the calls when necessary. The Rsim does not support dynamically linked libraries and our system also does not consider dynamic linking at the current state. Unfortunately in this article is not mentioned how the support for C library functions was added into the simulator.

Unisim project (August et al. 2007) was developed as an open simulation environment which should deal with several crucial problems of today simulators. One of the problems is a lack of interoperability. This could be solved, according to the article, by a library of compatible modules and also by the ability to inter-operate with other simulators by wrapping them into modules. Though this may seem to be a little out of our concern the idea of the interface within the simulator that allows adding any library is quite interesting. In our case we will have the possibility to add or remove modules from the library in a simple way. But the idea from the Unisim project would make the import of any other library far easier than it is now.

The articles above are all related to simulations. The C programming language is not a new one and it is not possible to list all the articles that are in any way related to any library of C language. The simulator is either created in a way that it already contains the library or it has at least some interface which makes it easier to import the library in case it is wrapped in a module. So far our simulator does not contain such interface.

LEVELS OF THE DESCRIPTION

In our project we use different levels of description. There is a cycle accurate level and an instruction accurate level. These two descriptions are both created in the Codal language but differ largely. In the following paragraphs I will try to point out the main differences.

The instruction accurate model can be considered more high level. It describes the instructions and theirs behavior. From this model it is possible to extract the semantics of the instructions and afterwards build the compiler backend from the semantics. We are also able to generate other tools from this model such as assembler, simulator and disassembler. Unfortunately we are not currently able to extract hardware from this model.

On the other hand the cycle accurate model is on lower level. It deals more with the hardware side of the controller. From this model we can extract hardware description. This description can later be turned into the hardware realization. We are also able to create the register transfer level (RTL) simulator which is as close to hardware as possible. This simulator is very useful for testing and is vastly used.

These two levels of description should not be mistaken. While many of our architectures are described on instruction accurate as well as on cycle accurate level it is not yet possible to generate all tools just from one level of description. Figure 1 shows which tools are associated with each of the models. Note that assembler and common simulator can be obtained from both descriptions while compiler or hardware description is model dependent.

The simulators obtained from our descriptions can vary in speed according to the level of optimalization. Currently we support levels of optimalization from 0 up to 3. In case of zero the simulator is created fast, but the execution of programs is quite slow. If we choose the opposite, so go for the highest level of optimalization we have to wait for the creation of the simulator but afterwards the execution of programs is swift.

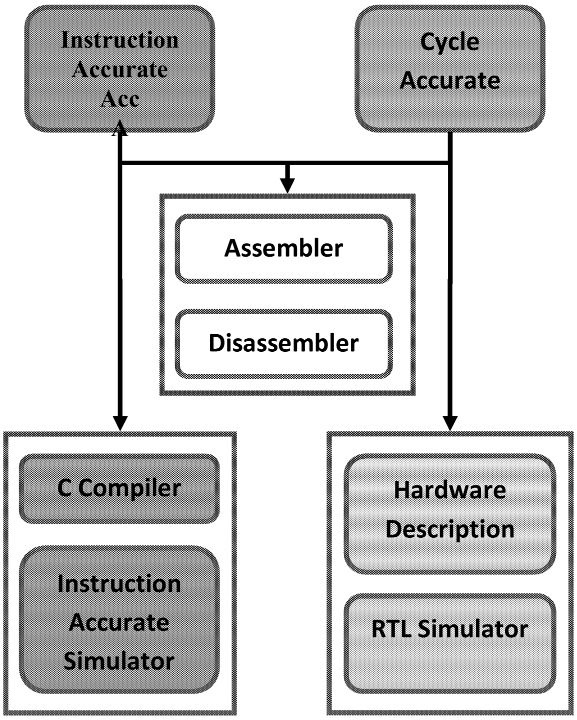


Figure 1: Models and Generated Tools

LIBRARY SUPPORT IN THE SIMULATOR

The main reason for porting the library into simulator is the fact that we need to add the support for C functions into the simulator itself. To be precise, we want to use the libc functions such as printf, malloc, free etc. in the programs that will be used for testing. And because we do not possess the development kits for all the platforms we use simulators instead.

If one does not grant libc library support in the simulated environment, the number of constructions we can use and test is very limited.

Even very simple programs can not be executed, because they usually use functions such as strcmp that is part of the C library. This program can not be compiled unless the inclusion of string.h and possibly some other header files are included.

On the contrary the main aim of testing is to cover as wide area as possible and also try as many different combinations of functions as we can. However, this goes against the idea of embedded solutions. And because we focus especially on embedded systems, we do not even try to cover all the functions provided by glibc or in our case Newlib (Newlib). In fact we will use and hence test only functions that can run under the simulated environment and are useful for the programs that will be executed on the given platform. Moreover embedded systems are not designed for use of vast number of constructions that programming languages offer. Typically there is just one task, usually quite complicated, that is launched repeatedly. As we will see, the functions that we will use form just small part of Newlib. The functions that are not important to us can be easily removed via configuration interface or it is possible to remove them manually.

The following parts were not removed from the library. Stdio, this was the main reason for porting the library; to get any human readable output from the simulator. Module for working with strings and memory, in our applications we would like to use functions such as memcpy, strcpy, strcat etc. Memory functions, for example malloc, free, realloc. Support for wchar, but without support of different encodings. The position of library in the whole process of testing is shown in the Figure 2.

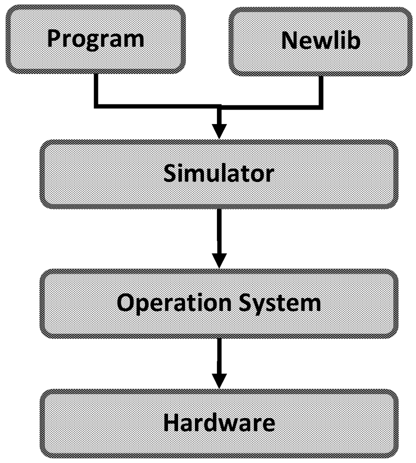


Figure 2: Position of Newlib in the Applications

Now return to the functions that remain in the library. They can be divided into two groups. First group consists of functions that are completely serviced within the simulated environment. For example function strcmp falls into this category. This function and its declaration remains unchanged within the simulator if it written in portable C. These functions are not tied with kernel header files so there is no need to change them.

The second group of functions consists of functions that are translated to the call of system function. Function printf can be used as an example of this group of functions.

In the beginning the call of printf function is translated on the call of system function, with the highest probability it is going to be the call of function Write. Write, being the POSIX function, is offered by the operation system. But as we want to use the simulator on UNIX platform as well as on Windows systems we have to get rid of these dependencies. To do so we will use the special instruction principle.

Special instruction principle

The special instruction principle means, that we will use instruction with the opcode that is not used within the instruction set for the special purpose. So far all architectures that were modeled within the Codasip project had several free opcodes. It is typical that the instruction sets do not use all operation codes that are provided. But in case of no free opcode this method can not be used. The special instruction principle will be used for ousting the dependencies on kernel header files.

Functions provided by operation system are called by the syscall mechanism. The system calls can be quite easily detected. Each library should have defined the syscall mechanism in special source file. This syscall mechanism differs, as they usually are platform dependent. So i386 architecture will have different syscall mechanism than arm.

We wish to preserve the mechanism. The syscalls will remain in the library, but with different meaning. The file containing syscall will be changed in the following way: in the beginning the parameters of the syscall will be placed at the given addresses in the memory and we will also define where the syscall return value will be placed. Afterwards the call of the chosen instruction will be performed. It is also possible to put the parameters into registers, but some platforms have limited number of registers, hence this method could cause problems. Figure 3 illustrates the idea of special instruction call.

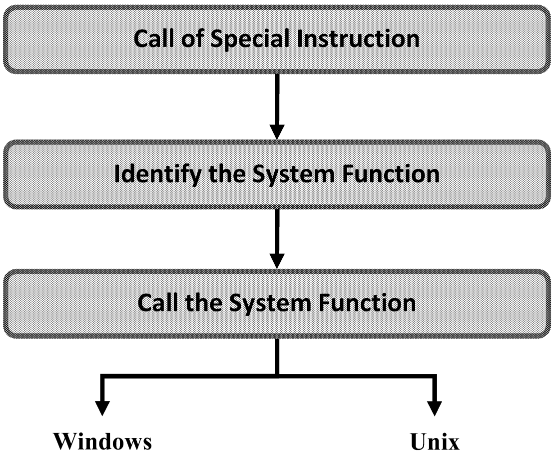


Figure 3: Calling Sequence of Special Instruction

In the future we would like to automatize this process. All the platforms we describe in Codal language can have the support for the syscall. On the side of the simulator we can ensure, that it will provide certain interface that is needed for the syscalls to be performed. The most difficult part seems to be the library itself. We need to provide for every new platform we model the actual implementation of the syscalls in the assembly language of the given platform. Moreover we need to write in assembly language routines for jumps, runtime (crt0.s) and some other files.

PROCESS OF TESTING

Now when we have thoroughly gone through the library porting, we can have a look at the test selection problematic.

Test selection phase

As we have a large amount of tests from the different sources, we need a universal approach that will define which tests are suitable for the compilation and execution on a given platform.

We have created a system of files that restricts the number of the tests that can be compiled on a given platform according to the libraries that are available. The libraries are just one of the test selection criteria; also other characteristics are taken into account for example the size of the registers or the size of stack.

The naming convention for these files is very simple. The file bear same name as the test does but have suffix .x instead of .c. The system is hierarchical. We can have the hierarchy because we support a nesting of the directories and we keep .x files not just for the tests, but also for the directories. In case of the directory the .x file has the same name as the directory with the .x suffix.

These files have minimal functionality. We try to keep their size as small as possible. Their typical functionality is that according to some state of the flags the test is excluded from testing, because implicitly all the directories and all the tests are selected for the testing. So, if we want to exclude the tests or whole directories from testing we have to indicate this.

As the size of the files is kept minimal the functionality and flag settings must be done elsewhere. This is performed centrally in the main testing module. The functions that check the current state of the flags and control what libraries are accessible for linking to the given platform are declared here. The centralization in this case has purely practical base. The typical usage of the .x files is that we disable testing of the whole directories according to the libraries that are accessible. The .x files can also bear other functionality. We can for example set different variables. We can specify flags that should be added to the compilation or add some files to the linker as in the following example.

```
if [ "$C_LIB" == "0" ]; then
    FILE_DEPS+=crt0.o
fi
```

On the level of files we most often use the .x files for the filtering the test that depend on the compiler-rt library for a given platform. As usually only few tests of any directory depend on the compiler-rt and the dependence does not have to be same for all platforms, the best solution is to condition the test execution by the platform and the compiler-rt presence. This is demonstrated in the following example.

```
is_arch "mips_basic" $1
  if [ "$?" == "0" ]; then
    if [ "$RUNTIME_LIB" == "0" ]; then
  RUN_TEST=0
    fi
  fi
```

The biggest advantage of this approach and also the main reason for introduction of this system is its universality. We employ the tests from the LLVM testsuite, the gcc testsuite, the Mibench (Guthaus et al. 2001) set of tests and we also have tests that were created within our project. The system of the .x files can be used for all these sources as long as we use just the tests without the testing infrastructure that is provided in several cases.

The only set of tests that we use together with the infrastructure that is provided together with the tests is the Perennial testsuite (Perennial).

Test execution and evaluation

The tests are currently executed on simulators obtained from cycle accurate as well as instruction accurate description. If the results vary it is an important signal for us that the models should be revised for compatibility.

The test evaluation is kept decentralized. As we deploy tests from different sources we need to keep the test evaluation together with the tests. For some tests we evaluate on the basis of exit code, but there are the tests that produce for example text output and we have to compare the output with the referential values (this is where the library comes to use).

As in case of test compilation we keep detailed logging information. We keep the output of the simulator and after the test evaluation we list it into the list of passed tests or failed tests according to the result of evaluation. The logs are created for every tested platform and can bear time reference.

For having a comparison with commercial compilers we tested our automatically generated compiler with commercial Perennial testsuite. The results described here were gained from the generated MIPS compiler. The testing was performed on our complete toolchain. The tests were compiled by our generated compiler and afterwards execute the tests on our simulator that was also automatically generated. We have only part of the Perennial testsuite. We used only the tests that examine the core of the compiler. We excluded some of the tests that can not be compiled due to the header files dependencies we do not support. The tests in the testsuite are divided into groups according to the chapter of the standard that is tested. We use tests for the clauses 5 and 6. We have mainly tests for the standard C90 and several tests for C99 standard. The final number of tests that we execute is 796. From 796 tests are 733 tests compiled and executed correctly. 63 tests fail either during the compilation or return incorrect value. The results are summarized in the following table.

Table 1: Results of Perennial Testsuite

Compiler	All tests	Pass tests	Fail tests	Not compiled	Not executed
Codasip	796	733	63	9	54
Gcc	796	796	0	0	0

As the table shows, just 63 tests do not succeed. Out of this 63 test only 9 do not succeed due to compilation error. One of these tests fails because of problems with linking. 54 tests fail, because they return wrong value. This can be caused either by wrong compilation or by incorrect simulation.

We tested gcc compiler with the Perennial testsuite to get comparison with our own compiler. All of the tests were compiled and executed correctly for x86 platform.

Some of the tests that failed during the compilation and execution with our toolchain included trigraphs that are not supported by the toolchain. The rest of the tests that failed are currently worked on.

CONCLUSION

In this paper we sketched the idea of adding the support for the C library into the simulator and described how the simulators are obtained and used in our project. The motivation is quite clear: to be able to use the library functions in the tests that are run on the simulator of the given microcontroller. The special instruction principle was proposed which enables us to forward the call of system function. It also allows us to identify which system function is called. This principle is quite universal and can be used for the majority of platforms. After implementation of this method, we are able to run all the functions that are commonly used such as I/O functions, memory management and string functions, etc. Moreover, we can adjust the library according to our needs. Thanks to the modularity we can enable or disable any module. This may turn to be an advantage, because the complete library occupies tens of megabytes and the compilation and linking such a library can be time consuming.

We also tested our generated compilers with the commercial Perennial testsuite. The compiler was tested against the C90 and C99 standard with good results when we take into account the fact, that the compiler is generated automatically. The fact that we can easily compose new testing systems and sets of tests into our own together with the results we gained is encouraging.

ACKNOWLEDGEMENTS

This research was supported by doctoral grant GA CR 102/09/H045, by the grants of MPO Czech Republic FR-TI1/038, by the grant FIT-S-10-2 and by the research plans MSMT no. MSM0021630528. This work was also supported by the IT4Innovations Centre of Excellence Project CZ.1.05/1.1.00/02.0070 and by the Artemis EU Project SMECY.

REFERENCES

August, D.; J. Chang; S. Girbal; D. Gracia-Perez; G. Mouchard; D. Penry; O. Temam; and N. Vachharajani. 2007. "UNISIM: An Open Simulation Environment and Library for Complex Architecture Design and Collaborative Development," *Computer Architecture Letters* , vol.6, no.2, pp.45-48, Feb. 2007

Guthaus, M.R.; J.S. Ringenberg; D. Ernst; T.M. Austin; T. Mudge; and R.B. Brown. 2001. "MiBench: A free, commercially representative embedded benchmark suite", *Workload Characterization*, Dec. 2001, pp.3-14, doi:10.1109/WWC.2001.990739

Hughes, C.J.; V.S. Pai; P. Ranganathan; and S.V. Adve. 2002. "Rsim: simulating shared-memory multiprocessors with ILP processors ," *Computer* , vol.35, no.2, pp.40-49, Feb 2002,

Masarik, K.; T. Hruska; and D. Kolar. 2006. "Language and development environment for microprocessor design of embedded systems", *In proceedings of IFAC workshop of programmable devices and embedded systems PDeS 2006*, pp. 120-125, Faculty of electrical engineering and communication BUT, 2006

Onder, S. and R. Gupta. 1998. "Automatic generation of microarchitecture simulators", *Proceedings of 1998 International Conference on Computer Languages*, pp.80-89, 14-16 May 1998

Přikryl, Z.; K. Masařík; T. Hruška; and A. Husár. 2010. "Generated cycle-accurate profiler for C language", *13th EUROMICRO Conference on Digital System Design, DSD'2010,* Lille, France, pp. 263—268.

WEB REFERENCES

Lissom Project. http://www.fit.vutbr.cz/research/groups/lissom.

Newlib. http://sourceware.org/newlib/

Perennial testsuite, http://www.peren.com/

SIMULATION OF AESTHETIC RATING OF CAD LAYOUTS

Daniel Skiera
Mark Hoenig
Juergen Hoetzel
Pawel Dabrowski*
Bosch Thermotechnik GmbH
Thermotechnology
Werk Lollar
Postfach 11 61
35453 Lollar, GERMANY
E-mail: Daniel.Skiera@de.bosch.com
E-mail: Mark.Hoenig@de.bosch.com
E-mail: Juergen.Hoetzel@de.bosch.com
E-mail: Pawel.Dabrowski@de.bosch.com

Slawomir Nikiel
Institute of Control and
Computation Engineering
University of Zielona Gora
Podgorna 50
65-246 Zielona Gora, POLAND
E-mail: S.Nikiel@issi.uz.zgora.pl

*Institute of Computer
Engineering and Electronics
University of Zielona Gora
Podgorna 50
65-246 Zielona Gora, POLAND
E-mail: P.Dabrowski@weit.uz.zgora.pl

KEYWORDS

Simulation, Computer Aided Design (CAD), Graphical User Interface (GUI), Information Design, Algorithm

ABSTRACT

Increasing complexity of Information Systems results in employment of various automated or semi-automated methods for data visualization. Good information design is very important for human-computer interfaces since it improves productivity and enhances human understanding. This paper proposes two novel algorithms for estimation of deployment of the visual elements of the CAD layout. GUI is treated as the output of simulation locating uniform rectangular blocks on the layout. Those blocks eventually are replaced by the system dependent object representations, e.g. visual metaphors of the heating system. The proposed tools and metrics can be used in simulation of aesthetic rating of various models, as long as they consist of logically separable elements. The results returned by the described algorithms were compared with results obtained by a survey. The paper discusses the theoretical background, the properties of proposed algorithms together with the sample prototype application.

INTRODUCTION

The information design is mostly related to visual media (Harrison 2012). Graphical User Interface (GUI) details may influence the overall user experience. Efficient design of complex systems requires an understanding of many things. Included are the perception of people: how we see, understand, and think (Winograd and Flores 1986). It also includes how information must be displayed to enhance human acceptance and comprehension (Miller 1955, Bodker 1991). Good design is a crucial part of any CAD system and must also consider the capabilities and limitations of the hardware and software of the human-computer interface (Wood 1997, Teo et al. 2000). Especially the software part, when equipped with some automated tools used in creation processes, needs special attention. Information design, crucial for human-computer interaction (HCI) should blend the results of visual design research, knowledge concerning people, knowledge about the hardware and software capabilities of the interfaces and artificial intelligence algorithms. Visual information may be an output of numerical simulations. During the design process visual metaphors are assigned to the individual elements. A logical flow of information should also to be determined. The next step is to organize and lay out individual elements of CAD clearly and meaningfully. Proper screen layout presentation and its background structure will encourage quick and correct information comprehension, the fastest possible execution of tasks and functions, and enhanced user acceptance (Gromke 2007, Hoffmann et al. 2011). However, it is much harder to estimate numerically the information related to aesthetics that is actually understood by user. There are some attempts to provide useful metrics (Gamberini et al. 2011). This paper will present new metrics for the redesigned CAD layouts, and some rationale and reasoning that explains why they are useful in numerical rating of deployment of visual elements.

LAYOUT ALIGNEMENT

Many researchers addressed interface and screen design from the users perspective (Hartswood and Procter 2000, Keppel and Wickens 2004, Galitz 2007). Generally fewer alignments of screen elements make it more visually appealing, it is known as the rule of minimum design (Sirlin 2009). Aligning elements will also make eye movement through the screen much more obvious and reduce the distance it must travel. Screen organization will also be more consistent and predictable. Alignment is achieved by creating vertical columns and horizontal rows of screen fields. Screen balance should be attained as much as possible. But how can we check whether the pattern created is consistent, predictable, and distinct? All symmetries are closely related to human sense of aesthetics, so one of the solutions is the visual complexity measure related to the even deployment layout described in the following sections.

RATING OF EVEN DEPLOYMENT

The CAD layout can be treated as an area (e.g. a display) with GUI elements, which are represented as uniform rectangles. An example of CAD layout is shown in Figure 1, the GUI elements are shown as the gray rectangles.

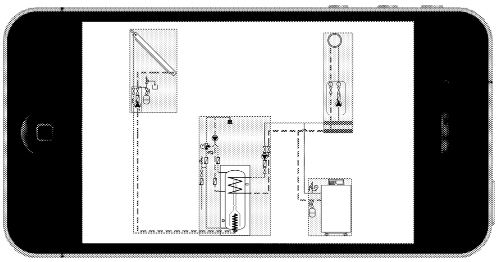


Figure 1: An example of GUI elements on the display of a smartphone.

The described algorithm simulates aesthetic (visual) rating of CAD layouts with GUI elements. This can be useful for construction of a cost (objective) function for a placement algorithm (Nikiel and Dabrowski 2011). The algorithms rate the even placement of the GUI elements in the given area of the CAD layout. The paper will not consider the connections between the GUI elements.

Preconditions

In this paper the presented algorithms need some preconditions:

- the GUI elements on the CAD layout can not overlap,
- the dimension of CAD layout affects its rating,
- the uniform rectangles of GUI elements may be positioned vertical or horizontal in the given area of the CAD layout,
- the GUI elements are treated as black rectangles,
- it is possible to compare two different CAD layouts if they include the same GUI elements and the dimension of CAD layouts is the same.

The Block Filter Algorithm

The algorithm partitions the CAD layout into non-overlapping blocks and returns the standard deviation of their density. If the standard deviation of the density of blocks is smaller, the distribution of associated GUI elements will be rated as better one. The Block Filter Algorithm requires proper determination of the number of blocks. An example of partitioning is shown in Figure 2.

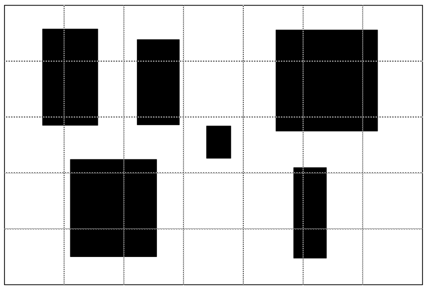


Figure 2: An example of the CAD layout with partition of blocks.

If the CAD layout is partitioned into too small or too big blocks the returned result will not determine the even distribution of the GUI elements correctly. This situation happens, because the results are comparable to each other for various distributions of the GUI elements. It has been observed that the best results are obtained if the area of the block was equal to 50% of the mean area of all GUI elements. It means, that 1,5 blocks have to be reserved for each GUI element. That holds for cases where the sizes of the GUI elements are similar. The algorithm does not work correctly in this case, because the much bigger GUI elements when compared to much smaller GUI elements affect the returned result strongly. The number of blocks is calculated from Equation 1.

$$a = nkA_l(\sum_{i=0}^{n} A_i)^{-1} \tag{1}$$

where:
a - the number of blocks
n - the number of GUI elements
$i \in N \setminus \{0\}$
A_l - the area of the layout
k - the number of blocks for a GUI element. Experiments determined that $k = 1,5$
A_i - the area of i GUI element

All blocks may have the same dimensions.

The second issue with the Block Filter Algorithm is the determination of the horizontal and vertical number of blocks. The blocks should be squares but for some cases it is not possible. The horizontal number of blocks is calculated from Equation 2.

$$a_h = \sqrt{at} \tag{2}$$

where:
a_h - the horizontal number of blocks
t - the proportional dimension of CAD layout (see Equation 4)

The vertical number of blocks is calculated from the Equation 3.

$$a_v = \sqrt{a\frac{1}{t}} \tag{3}$$

where:
a_v - the vertical number of blocks

Proportional dimensions of CAD layout is calculated from the Equation 4

$$t = \frac{w}{h} \tag{4}$$

where:
w - the width of CAD layout
h - the height of CAD layout

The dimension of the blocks can be calculated from the width and height of the underlying CAD layout. The width of a block is calculated from the Formula 5.

$$w_b = \frac{w}{a_h} \tag{5}$$

where:
w_b - the width of block

The height of a block is calculated from the Formula 6.

$$h_b = \frac{h}{a_v} \tag{6}$$

where:
h_b - the height of block

The Block Filter Algorithm calculated the standard deviation of the density of blocks. The density of block is calculated from the Formula 7.

$$\rho = \frac{A_c}{A_b} \tag{7}$$

where:
ρ - the density of block, $\rho \in < 0, 1 >$
A_c - the black area of block
A_b - the area of block

Finally the standard deviation of the density of blocks is calculated from the Formula 8.

$$\sigma = \sqrt{\frac{1}{a}\sum_{i=1}^{a}(\rho_i - \bar{\rho})^2} \tag{8}$$

where:
σ - the standard deviation of the density of blocks
ρ_i - the i density of block
$\bar{\rho}$ - the mean of densities $\bar{\rho} = \frac{1}{a}\sum_{i=1}^{a}\rho_i$

Figure 3 shows the visual solution returned by the Block Filter Algorithm. The white color ($rgb(0, 0, 0)$) corresponds to the density $\rho = 0$ and the black color ($rgb(255, 255, 255)$) corresponds to the density $\rho = 1$. Intensity of the gray color ($rgb(c, c, c)$, $c \in [1, 2, \ldots, 254]$) depends on the density $\rho \in (0, 1)$. If density is higher, the gray color is darker.

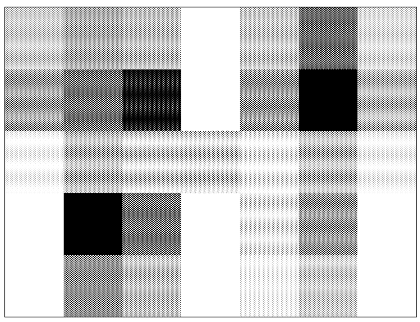


Figure 3: The visual solution for the example shown in the Figure 2 returned by the Block Filter Algorithm.

If colors are more uniform, the GUI elements will be more evenly distributed. For the Block Filter Algorithm, a situation can occur that for two different deployments of the GUI elements on the CAD layout the standard deviation of their density is the same. If the GUI element in the right bottom corner from the Figure 4 is moved to the gray position, then the standard deviation of the density of blocks will still be the same.

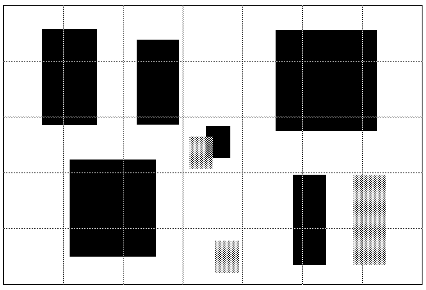


Figure 4: Some various positions of the GUI elements (the gray rectangle), which do not change the standard deviation of the density of blocks.

This happens because, the move does not change the density of blocks but only their order. The moves do not affect the standard deviation of the density of blocks. The similar situation is for the smallest GUI element depicted in Figure 4. The GUI element can be moved into its block or to another empty block (without black area) and this move will not change the return result of the Block Filter Algorithm.
The time complexity for the Block Filter Algorithm is dependent on the number of blocks and is equal to $\Theta(N)$.

The Zoom Algorithm

The Zoom Algorithm zooms in the GUI elements and calculates the overlapping and the left (outside) area. The GUI elements are zoomed in evenly with a fixed value. An example is shown in the Figure 5.

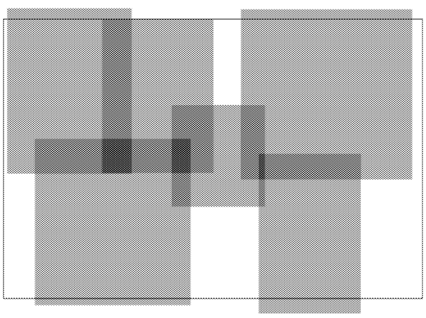


Figure 5: The GUI element from Figure 2 are zoomed in. The darker fields represent the overlapping and the fields outside the frame represent the outside area.

The algorithm can be divided into two phases:

1. calculate the overlapping of the GUI elements,
2. calculate the outside area.

The GUI elements are checked for overlapping with another GUI elements. The overlapping area is calculated from the Equation 9. For this equation the areas are treated as a set of boxes (pixels).

$$A_{ol} = \sum_{0<i<j}^{n} \{A_i \cap A_j\} \tag{9}$$

where:
A_{ol} - the sum of overlap areas of the GUI elements
n - the number of GUI elements
$i, j \in N \setminus \{0\} \cap 0 < i < j$
A_i, A_j - the i, j area of GUI element

The outside area is calculated from the Equation 10. For this equation the areas are treated as a set of boxes (pixels).

$$A_{os} = \sum_{i=1}^{n} \{A_i \setminus A_l\} \tag{10}$$

where:
A_{os} - the sum of outside areas of the GUI elements
$i \in N \setminus \{0\}$
A_i - the i area GUI element
A_l - the area of CAD layout

The Zoom Algorithm returns the value calculated from the Equation 11.

$$Z = A_{ol} + A_{os} \tag{11}$$

where:
Z - the value returned by the Zoom Algorithm

If the value returned by the Zoom Algorithm is smaller, the distribution of GUI elements on the CAD layout are rated as better. The problem with this algorithm is caused by the zoom margin (the margin will be added to all sides of the GUI elements). The question is, how much should the GUI elements be zoomed in? Experiments determined, that the GUI elements have to be zoomed in, in such a way, that the sum of area of zoomed GUI elements is equal to the area of CAD layout multiplied by 75%. The parameter $p = 0.75$ was experimentally determined. To determine the zoom margin the Equation 12 should be solved.

$$kA_l = \sum_{i=1}^{n} [(d_{1i} + z)(d_{2i} + z)] \tag{12}$$

where:
p - the parameter is equal to 0.75
z - the margin for the Zoom Algorithm
d_{1i}, d_{2i} - the length of sides of the i GUI element

The two CAD layouts can be compared, if the zoom margins are the same. The algorithm returns good results, but has also a small drawback. Figure 6 shows the various configurations of GUI elements, where the returned result of the Zoom Algorithm is the same.

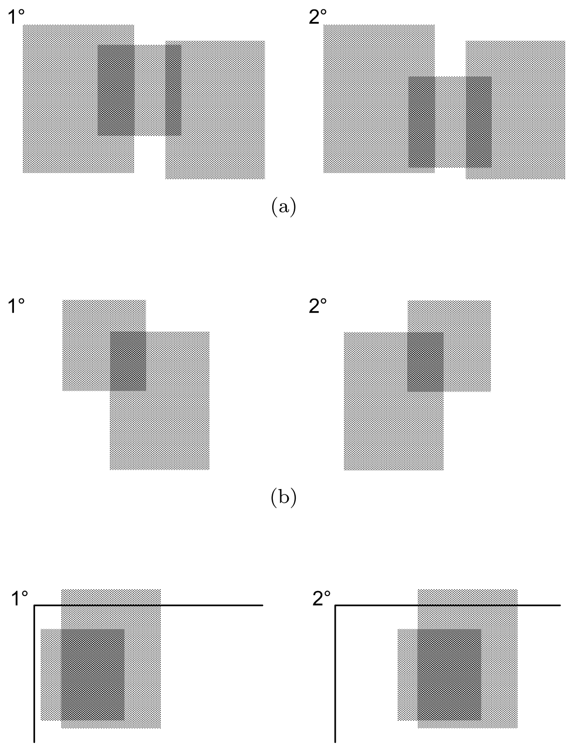


Figure 6: For the cases 1° and 2° the Zoom Algorithm returns the same results.

The time complexity for the Zoom Algorithm is dependent on the number of GUI elements and is equal to $\Theta(N^2)$.

Discussion

The Block Filter Algorithm works correct for the GUI elements with similar dimensions and its time complexity is equal to $\Theta(N)$. The Zoom Algorithm returns better results (see Figure 7) than the Block Filter Algorithm but its time complexity is equal to $\Theta(N^2)$.

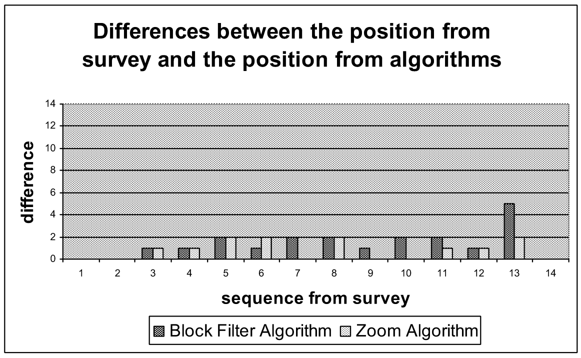


Figure 7: Results from the survey to the results returned by the Block Filter and Zoom algorithm.

The survey was conducted on a group of fourteen people (users in the CAD related fields), who received fourteen examples of GUI elements deployment. The examples were ordered by the respondents from the visually most to the least appealing. For example, in the chart in Figure 7, the survey classified a deployment (from fourteen deployments) on the position seventh (the x axis). The Zoom Algorithm classified the same example of deployment also on position seven, because the difference (the y axis) is equal to zero. The Block Filter Algorithm classified the same example of deployment on the position five or nine, because the difference is equal to two. Figure 7 shows, that the two algorithms can be used for a simulation of aesthetic rating of CAD layouts, because the differences between the order from the conducted survey and the order returned by the algorithms are similar.

CONCLUSION

Information design is very important in the field of CAD user interfaces. The paper proposes two novel algorithms for estimation of deployment of the visual elements in the given CAD layout. GUI is obtained as the output of simulation placing rectangular blocks on a given layout area. The paper discusses alternative metrics of visual aesthetics, based on the Block Filter and the Zoom algorithms. The researches showed that the Zoom Algorithm returns better results than the Block Filter Algorithm, but its time complexity is bigger. The authors plan to develop further the methods in order to support more complex systems.

ACKNOWLEDGEMENT

This study is supported by Bosch Thermotechnik GmbH, and the authors would like to thank the company for the cooperation and assistance rendered.

REFERENCES

Bodker S., 1991. *Through the interface: a human activity approach to user interface design.* Lawrence Erlbaum Associates, Hillsdale.

Galitz W., 2007. *The Essential Guide to User Interface Design.* Wiley.

Gamberini L.; Spagnolli A.; Prontu L.; Furlan S.; Martino F.; Solaz B.; Alcaniz M.; and Lozano J., 2011. *How natural is a natural interface? An evaluation procedure based on action breakdowns. Personal and Ubiquitous Computing*, 5.

Gromke G., 2007. *Digital Asset Management - der effektive Umgang mit Mediendaten. Proceedings, Intl Conf EVA 2007 Berlin*, 161–166.

Harrison C., 2012. *InfoDesign Understanding by Design, TapSense: Enhancing Finger Interaction on Touch Surfaces.* URL `http://www.informationdesign.org`.

Hartswood M. and Procter R., 2000. *Design guidelines for dealing with breakdowns and repairs in collaborative work settings. International Journal on Human Computer Studies*, 53, 91–120.

Hoffmann P.; Lawo M.; and Kalkbrenner G., 2011. *Zur Aesthetik interaktiver Medien - Hypervideo im Spannungsfeld zwischen Usability und Design. Proceedings, Intl Conf EVA 2007 Berlin*, 117–123.

Keppel G. and Wickens T., 2004. *Design and analysis: a researchers handbook.* Pearson/Prentice Hall, Upper Saddle River.

Miller G., 1955. *The Magical Number Seven, Plus or Minus Two - Some Limits on Our Capacity for Processing Information. In: Psychological Review*, 101, no. 2, 343–352.

Nikiel S. and Dabrowski P., 2011. *Deployment algorithm using simulated annealing.* In *Methods and Models in Automation and Robotics - MMAR 2011 : 16th international conference; ISBN: 978-1-4577-0913-5.* Miedzyzdroje, Poland, 111–115.

Sirlin D., 2009. *Subtractive design. Game Developer*, 23–28.

Teo L.; Byrne J.; and Ngo D., 2000. *A Method for Determining the Properties of Multi-Screen Interfaces. International Journal of Applied Mathematics and Computer Science*, 10, no. 2, 413–427.

Winograd T. and Flores F., 1986. *Understanding computers and cognition.* Ablex Publishing, Norwood.

Wood L., 1997. *User Interface Design: Bridging the Gap from User Requirements to Design.* CRC Press.

BIOGRAPHY

DANIEL SKIERA is currently working as a Software Architect at Bosch Thermotechnik GmbH, Lollar, Germany. He obtained a PhD in Physics at the University of Giessen, Germany. His scope of work included data analysis, image processing and system simulation. His e-mail is `Daniel.Skiera@de.bosch.com`.

MARK HOENIG is currently working as a group manager at Bosch Thermotechnik GmbH, Lollar, Germany. He graduated in Physics at the University of Goettingen and received a PhD from the University of Cologne. His work included data analysis and nonlinear optimization. His e-mail is `Mark.Hoenig@de.bosch.com`.

JUERGEN HOETZEL is currently working as manager at Bosch Thermotechnik GmbH, Lollar, Germany. He graduated in Electronics at the University of Darmstadt and received a PhD from the Technical University Berlin. His work included system architecture and Internet connectivity. His e-mail is `Juergen.Hoetzel@de.bosch.com`.

SLAWOMIR NIKIEL is currently the Professor at the Institute of Control and Computation Engineering, Department of Electrical Technology, Computer Science and Telecommunication, University Of Zielona Gora, Poland. His research interest include HCI, game programming and multimedia systems. His e-mail is `S.Nikiel@issi.uz.zgora.pl`.

PAWEL DABROWSKI is currently the PhD student at the Institute of Computer Engineering and Electronics, Faculty of Electrical Engineering, Computer Science and Telecommunications, University Of Zielona Gora, Poland. His dissertation is developing in cooperation with Bosch Thermotechnik GmbH, Lollar, Germany. His research interest include automatic deployment of GUI elements. His e-mail is `P.Dabrowski@weit.uz.zgora.pl` or `Pawel.Dabrowski@de.bosch.com`.

THERMAL IMPULSE SENSOR WITH FOUR POINTS ACQUISITION CARD FOR TEMPERATURE MEASUREMENT

Vlad Jinga
Cornel Samoilă
Doru Ursuţiu
Petrică Cotfas
Transilvania University of Braşov
B-ul Eroilor 29, 500035, Braşov,
Romania
E-mail: jingavlad@yahoo.com

KEYWORDS
Automatic control, electronics, industrial control, industrial processes, signal processors

ABSTRACT

The authors of this paper are trying to develop a new type of gas flow meter that works using a thermal impulse. For determining the real gas flow, some temperature measurements are required, and from the obtained data – through post processing – the flow can be calculated. The temperature measurements will be done in four points along the sensor and for this kind of Data Acquisition (DAQ), a special circuit was build. During this paper, the new circuit manufacture will be described and some preliminary tests will be presented with the relevant results obtained. Afterwards, the implementation of the new board regarding the development of the new thermal impulse gas flow sensor will be explained thoroughly.

INTRODUCTION

Because of the latest development concerning the hardware equipments including DAQ boards and the software data processing methods, the authors of this paper came back to an older idea regarding a gas flow sensor.
In the 80s there was a concept on how a special flow meter could work, measuring the flow by not disturbing the flow section. The method was a simple one, concerning a heat exchange from a thermal source to the gas volume and by measuring the temperature evolution of the gas in time to calculate the real flow. A lot of attempts were made by the specialists from those times to implement such a high-tech sensor, but no one accomplished that without disturbing with an active element the flow section, thus causing the appearance of turbulences. Because of that, the measurement flow was a local one and for determining the global flow, a software correction was needed. Implementing the method in such a way, using the soft correction, the measured results regarding the gas flow weren`t so accurate.
Nowadays, because the technique evolved so much comparing it with the one available 30 years ago, the authors of this paper believe that the method of the thermal impulse can be implemented in such a special device, resulting a new type of flow meter on the market.
The working principle of this new gas flow sensor is a simple one: a repeating cycle having a specific frequency is used. The repeating cycle consists of first generating a thermal impulse, heat exchange taking place between the heat source and a specific gas volume and then, by measuring the evolution of the gas temperature for a short period of time with the help of a DAQ board connected to a PC, to calculate the exact volume of gas that passed inside the sensor.
In other words, the working principle of the new gas flow meter can be translated like this: if the volume of heated gas is large, the heat dissipation will be greater than in the case when the volume of gas is smaller. So, supposing each time we generate the same amount of heat for the gas to collect, if the gas volume is larger, after the heat exchange takes place and the gas temperature reaches the peak, the temperature will decrease more quickly than in the case where the volume of gas would have been smaller. So the new gas flow meter will work having as a basic rule the direct relation between the temperature evolution and the gas volume (that is the speed of the gas through the sensor multiplied by the inner diameter of the sensor and multiplied by time).
As a result of implementing the above described principle, this new gas flow meter will be able to measure the actual flow in a transitory working regime.
Two essential hypothesis are specified from the beginning of the work:

- first, the new sensor will be used only for measuring low flows; this comes as a limitation of the measuring method, but it will be cleared up in an experimental way, and the upper limit regarding the precise measured gas volume will be determined;

- second, the sensor won`t affect in any way the gas flow by introducing a perturbation element, in this way eliminating the caused perturbations from other implementations regarding the working principle.

These two mentioned hypothesis are very important, from these described situations resulting all the major advantages regarding this new gas flow sensor.

ACTUAL DEVELOPMENT OF THE GAS SENSOR

Regarding the actual development of the thermal impulse gas flow meter, two main parts were developed during the last two years:

- first, a theoretical model regarding the physical phenomena taking place inside the flow sensor, concerning the thermal impulse generation, afterwards the heat transfer and then the gas temperature evolution was written; in this way, some simulations are still being made for theoretical results/predictions to be obtained from this mathematical model;
- second, a practical and experimental part was developed, involving the purchase of different equipments, implementing some available hardware solutions and thus resulting an experimental booth, presented in Figure 1.

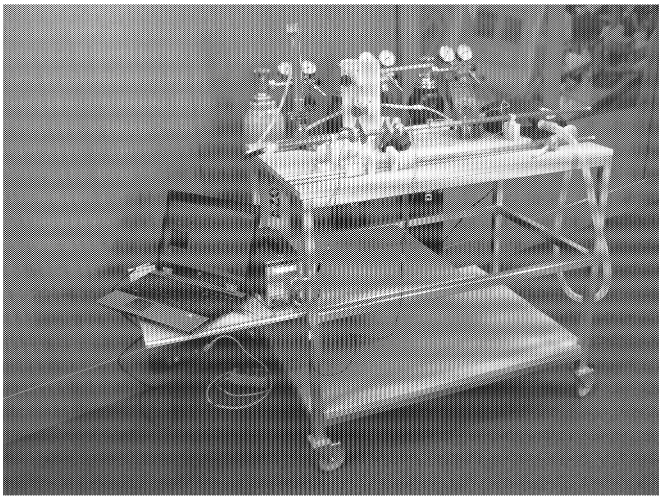


Figure 1: Experimental booth for testing and calibrating the thermal impulse gas flow sensor

The experimental booth consists of the gas source, three tanks of 10 liters, each one having its own reducer for Oxygen, Nitrogen and CO_2. With the help of some special, very accurate needle taps, the gas flow can be controlled and the feedback regarding the gas volume going into the sensor put to the tests is assured by a classical ball flow meter.

Afterwards, the thermal impulse generator – a resistive coil – is operating, heating the volume of gas for a specified period of time – controlled through a PC that actually controls the voltage source that gives power to the coil.

Then, the measuring part comes, where the evolution of the gas temperature is monitored in four distinct points, each point being very precise located along the gas flow.

The temperature measurements are done by using a DAQ board, BNC 2120 from National Instruments company from the United States of America and special K thermocouples that are placed along the sensor.

The used DAQ board has a dedicated channel for the usage regarding thermocouples and at the beginning of the experimental work, the gas temperature was monitored using only one K thermocouple.

Because the results obtained from one measuring point were not enough for determining through soft post processing algorithms (established in the mathematical model) the real gas flow, the temperature measurements needed to be extended to more than one point.

This decision caused a small impediment concerning the experimental part: the used DAQ board (BNC 2120) had only one dedicated channel for thermocouples and the application needed four this type channels.

The solution was a simple one: because the BNC 2120 DAQ board has another eight normal channels, the four temperature measurements had to be done by using these regular channels.

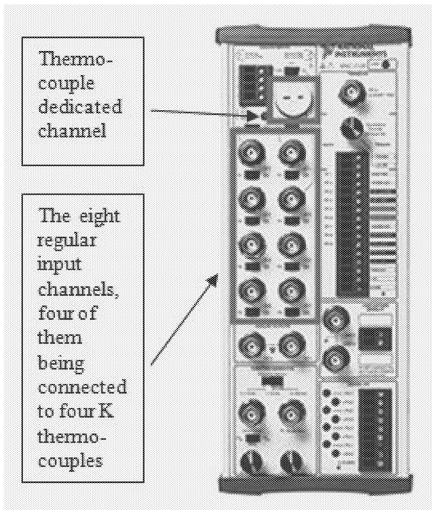


Figure 2: The National Instruments BNC 2120 DAQ board

In this situation, the question was how to properly connect the four K thermocouples that have a special connector to the four DAQ board channels that have a BNC connector each?

After proper documentation, the solution was to use for each channel/thermocouple another special circuit consisting of a thermocouple amplifier. This circuit had to be manufactured in such a way, for the board to have as an input a special K type thermocouple junction and as an output a BNC connector. In this way, the thermocouple would be connected to the thermocouple amplifier board using the special K thermocouple connection and the thermocouple board would be connected to one of the regular input channels of the DAQ BNC 2120 board with the help of a BNC – BNC cable.

CIRCUIT BOARD DEVELOPMENT & TESTING

The used circuit was a simple one, the main part of it being a precision thermocouple amplifier with cold junction

compensation, like in Figure 3, where an example of such a basic connection using a J or a K thermocouple is shown.
After proper documentation, the AD 8495 AMRZ was used as the precision thermocouple amplifier with cold junction for the described circuit.
The three main reasons for using the AD 8459 AMRZ in this specific application were:

- ambient temperature (reference junction): 0 – 50 ^{0}C;
- measurement junction: full K type range;
- measurement range: - 25 to + 400 ^{0} C.

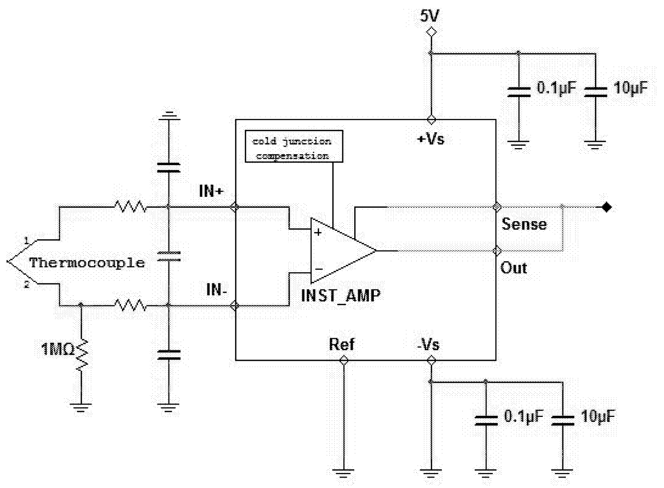


Figure 3: Basic connection for the AD 8495 AMRZ with a K type thermocouple

The AD 8495 AMRZ is a precision instrumentation amplifier with thermocouple cold junction compensators on an integrated circuit. It produces a high level (5 mV/ ^{0} C) output directly from a thermocouple signal by combining an ice point reference with a pre – calibrated amplifier. It can be used as a standalone thermometer or as switched output set point controller using either a fixed or a remote set point control.
The 5 mV/ ^{0} C output is achieved by the AD 8495 AMRZ circuit with an error of less than ± 2 ^{0} C, thus resulting in good temperature measurements, proper for the development of the new gas flow sensor.
Another advantage of this circuit is the fact that it works using a 5V single supply and the BNC 2120 DAQ board has an output of 5V, so there is no use for another power supply.
The next figure will show the development of the desired circuit in Multisim software, first the 3D version of the circuit and afterwards the one layer circuit.
Before manufacturing the final board, that consists of four such circuits, one for each channel/thermocouple, the circuit was pretested with the help of a rapid prototyping equipment NI Elvis, also from National Instruments company, as shown in Figure 5.
The testing was done by reading the temperature from a K thermocouple using Measurement & Automation Explorer software especially developed by National Instruments company for the NI Elvis equipment.

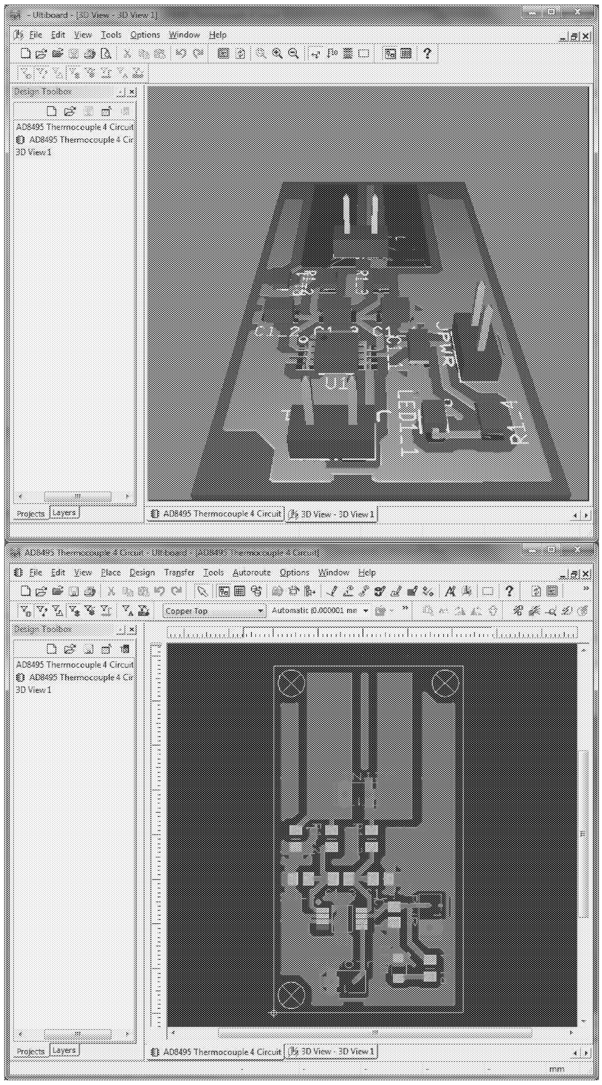


Figure 4: Multisim circuit development

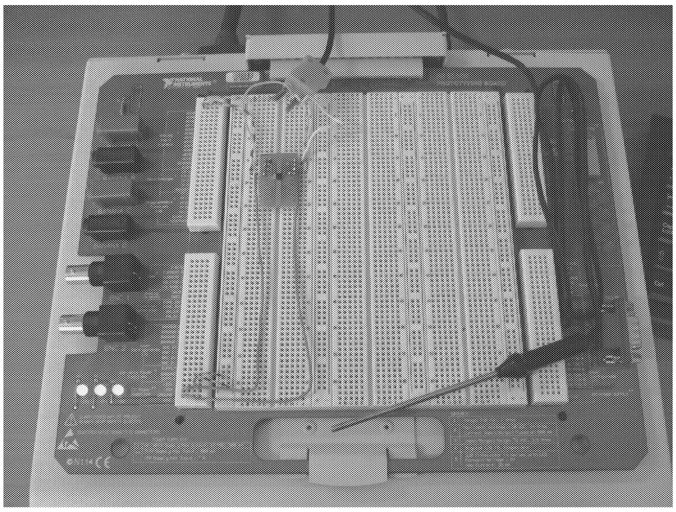


Figure 5: Pretesting of the circuit to validate its functioning

The testing had good results, the read temperature using the developed circuit was correct. The only problems were the interferences as specified in the documentation regarding the AD 8495 AMRZ thermocouple amplifier.
Theoretically, according to the documentation – the official data sheet – for the AD 8495 AMRZ having 0.025 ^{0} C / ^{0} C ambient temperature rejection, a 20 ^{0} C change in the reference junction temperature adds less than 0.5 ^{0} C error to the temperature measurement. In reality, even a rather small change in the reference junction temperature makes the circuit go wild, the measured temperature starts oscillating quite a lot.
Another disadvantage of this circuit, that was observed in practice, was the fact that the temperature measurements are very vulnerable to interferences, for example the presence of a mobile phone in the proximity of the AD 8495 AMRZ.
Because of these inconveniences, the authors of this paper decided that the final board, consisting of four such circuits – one for each channel/thermocouple as described earlier, should be placed in a special metal box, for the ambient temperature to be constant and for stopping all the interferences.
The above described solution was implemented, and a special box having enough space for putting inside the four circuits board was manufactured. Holes were drilled in the box for both the K thermocouples wires (all the connectors are inside the box) at the input and also for the BNC connectors as outputs. The 5V power supply is the same for all four circuits.
For connecting the four thermocouples to the board, a special piece was manufactured out of Plexiglas using the lathe. By screw – fastening this Plexiglas piece to the board, the K thermocouple junctions make perfect electrical contact with the board; the Plexiglas piece has milled on one side the tracks of the K thermocouple junction.
The connections between the new developed board and the BNC 2120 DAQ board are as shown in Figure 6: one BNC – BNC cable for each channel/thermocouple (four in total) and a power supply cable.

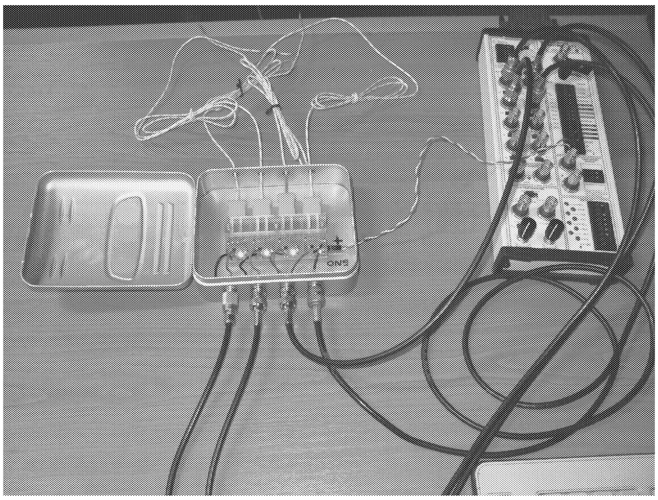


Figure 6: The connections between the four circuits board and the BNC 2120 DAQ board

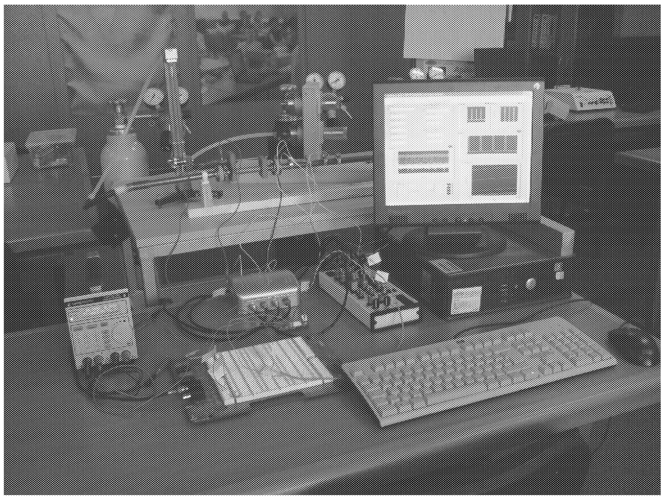


Figure 7: Testing the final four circuits board

The testing of the new four circuits board was not able by using the Measurement & Automation Explorer software as for the previous case when only one channel/thermocouple was involved. A dedicated software program had to be developed in LabVIEW – graphical programming – for testing the final version of the board as shown in Figure 8.

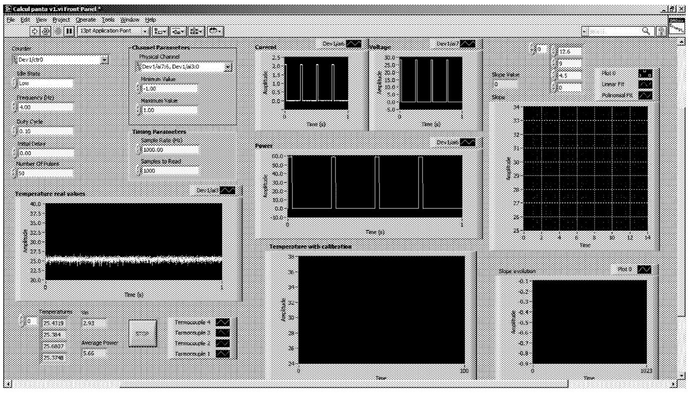


Figure 8: Front Panel of the dedicated software

At first, the results were not so satisfactory, because each thermocouple registered another temperature value. For this matter a software calibration was made as shown in Figure 9: from each measured value it was subtracted the difference between the mean value of all the measured temperatures on a channel and the mean value of all the measured temperatures on all four channels. After this software implementation regarding the automatic calibration, the results were more than satisfactory: the values of the measured temperatures for one channel to another fluctuate only at the second decimal.
Also, all the retrieved data from the four points temperature measurements will be saved in the PC`s memory for being able to be post processed.

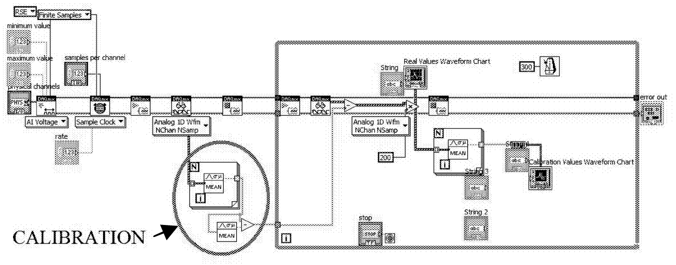


Figure 9: The Back Panel of the dedicated software

WORK IN PROGRESS & FUTURE WORK

For the future, the authors of this paper proposed to really use this new implementation regarding the temperature measurement done in four points in the developing of the new thermal pulse gas flow sensor.
From the practical point of view, the sensor part where the temperature measurements will take place is consisting of a copper pipe having 15 mm the inner diameter. For increasing the thermal conductibility of this material, the copper pipe is on the verge of being anodized, using different scenarios concerning the electrolytic solutions – pH, applied voltage and time exposure.
Also, a new version of the box is being under development: the new box will contain the same four circuits board, plus the DAQ hardware regarding the thermal impulse generation (for visualizing the thermal pulse in real time, a DAQ is needed that consists of mounting a low capacitance resistance in series with the existing coil and to measure the voltage/current applied to that resistance); the only difference will be that as outputs, the metallic box will have six BNC connectors and six channels will be used from the total of eight from the BNC 2120 DAQ board. As a consequence, the LabVIEW software will be upgraded according to the new necessities.
From the software point of view, this new developed software for reading the temperature values from four points along the flow will be updated with the voltage/current DAQ part regarding the thermal pulse visualization and then it will be integrated in another LabVIEW software that already contains the part for controlling the power supply for the thermal pulses generator. Afterwards the data post processing part will be also included in the same software, thus resulting the final software version for the new gas flow meter.

CONCLUSIONS

For the temperature measurements, the new developed board is very handy especially if the DAQ equipment has BNC connectors. Also, as shown before, the user can add to the board as many circuits as needed, for measuring simultaneously the temperatures in more than four points: for example in our case, by using the BNC 2120 DAQ board, the maximum is eight points.
The only disadvantage regarding this new hardware development is the fact that the board is sensitive to thermal/electrical influences and needs to be mounted inside a metallic box, for it to function properly. But, when speaking about an industrial environent, this approach is one of the best because the electrical circuit is more than protected from the above mentioned thermal/electrical influences and also from the unfriendly working conditions, like for example humidity or corrosion.
Coming back to the new gas flow meter development, the authors of this paper consider that the thermal impulse gas flow sensor will be a real succes on the market mainly because of its advantages caused by the simple working principle:

- the flow sensor will work properly in a transitory regime;
- the flow sensor will be very accurate because there is no intrusion in the flow section causing the elimination of the software correction module (that is used by classical flow sensors and that causes measuring errors), thus resulting the usage for calibrating the other types of flow meters;
- because the flow sensor will measure low flows with high accuracy, it will be appropriate for usage both in medical domain and also in the industrial environment.

ACKNOWLEDGEMENT

This paper is supported by the Sectoral Operational Program "Human Resources Development" (SOP HRD), financed from the European Social Fund and by the Romanian Government under the contract number POSDRU/88/1.5/S/59321.

REFERENCES

Jinga, V., Samoila, C. (2010): *Gas flow measurement methods using pulse sensor* in Creativity and Inventics Journal, Ed. Transilvania University of Brasov

Jinga, V., Oros, R., G., Samoila, C., Ursutiu, D. (2011): *Global flow rate modeling with local heating regime* in Proceedings of AFASES 2011 Conference p. 795 – 799

Jinga, V., Samoila, C., Ursutiu, D. (2011): *Experimental booth for determining gas flow using local heating transitory regime* in Proceedings of UgalMat 2011 Conference p. 77 – 81, Galati University Press

Jinga, V., Oros, R., G., Samoila, C., Ursutiu, D. (2011): *Mathematical model of heat transfer in the impulse sensor* in Proceedings of DAAAM 2011 Conference p. 1259 - 1260

Oros, R., G., Jinga, V., Samoila, C., Ursutiu, D.,Cotfas, P., Cotfas, D.: *Industrial monitoring and control of temperature using TAG4M* in Proceedings of REV 2011 Conference

Oros, R., G., Jinga, V., Samoila, C., Ursutiu, D. (2010): *New technologies – WITAG for temperature and pressure monitoring in critical areas of chemical equipments* in Proceedings of METAL 2010 Conference

Petru A. COTFAS, Daniel T. COTFAS, Doru URSUTIU, and Cornel SAMOILA "NI ELVIS Computer-Based Instrumentation", National Technology & Science Press – USA, ISBN 978-1-934891-11-7, 2011

Petru Cotfas Adrian "Prelucrarea semnalelor. Aplicatii in LabVIEW", Ed. Lux Libris, ISBN 978-973-131-071-8, Brasov, 2010

WEB REFERENCES

http://ro.farnell.com/analog-devices/ad8495armz
http://sine.ni.com/nips/cds/view/p/lang/en/nid/10712

SIMULATION IN ENGINEERING

WINTUS – A SIMULATOR FOR USUAL WIND TURBINES

Dan Stefanoiu, Florin Sebastian Tudor, Janetta Culita

„Politehnica" University of Bucharest, ROMANIA
Faculty of Automatic Control and Computer Science (www.acs.pub.ro)
E-mails: danny@indinf.pub.ro, tudor.s.florin@gmail.com, jculita@yahoo.com,

KEYWORDS
Horizontal axis wind turbine, analytical modeling, SIMULINK models.

ABSTRACT
The aim of this article is to introduce WINTUS – a simulator that replicates the dynamics of wind turbines from horizontal axis class. Two main purposes of WINTUS are outlined within the article. The first one is practical: allowing the operator to test various identification and control strategies, before implementation on the plant. The second one is didactical: analyzing the plant dynamics even in case of weak wind activity (or in absence of such activity). Behind the simulator, analytical non linear models of turbine components are integrated. The description is completed by a set of simulation results, after testing WINTUS according to several scenarios.

1. INTRODUCTION

Horizontal Axis Wind Turbines (HAWTs) are nowadays spread all over the Earth, as there is an increasing need to exploit green energy sources, in order to save the conventional resources, as much as possible. From modeling point of view, HAWT is a multi-variable system with intrinsic nonlinear dynamics, heavily influenced by a random process – the wind activity (Burton et al. 2001), (Bianchi et al. 2007), (Jain 2011). Numerous wind farms were installed so far in both off shore and on shore environments, but mostly in zones with strong wind activity. Nevertheless, extending the wind farms in zones with weak-medium wind activity became a necessity. Therefore, increasing the efficiency of wind power capture in such conditions is an obvious challenge.

It has been proven that HAWT efficiency cannot increase beyond the Betz limit of 59.26% (Ackerman 2005), especially because of propeller discontinuity. This is an ideal limit, as some other losses have to be accounted. For example 6% are wasted because of inappropriate blades/nacelle orientation, and/or tower height. Joule effect, small local oscillations and sub-optimal generator control take another 8-12%. Thus, the actual level of efficiency (of 40-45%) is quite far below the Betz limit, which means that improvements still are possible, by constructive approaches, as well as by more efficient automatic control techniques. A comprehensive state of the art on HAWT constructive issues and control techniques was very recently introduced in (Pao and Johnson 2011). As this tutorial article outlines, the linear modeling and control of HAWT is dominating the scientific literature, whilst the nonlinear identification and control seemingly is in the beginning of its era. Consequently, tools that can realistically simulate the HAWT dynamics are necessary.

Various mathematical models of wind conversion plants were proposed in the literature so far. For example, (Bianchi et al. 2007), (Hooft et al. 2007), and (Munteanu et al. 2008) are among the best references in this aim, since wind turbines are extensively described, from the mathematical modeling point of view. Based upon such descriptions, simple to complex simulators can be designed and implemented. WINTUS, shortly described next, is a simulator of medium complexity that tries to reach for a good trade-off between operability and accuracy.

The article is structured as follows. The next section is devoted to a short general description of HAWTs. Analytical models of each HAWT component are presented in section 3. They constitute the foundation of WINTUS simulator introduced in section 4. Some simulation results obtained after testing WINTUS are revealed in section 5. The concluding remarks complete the article.

2. GENERAL HAWT MODEL

The basic components of a HAWT are shown in Figure 1.

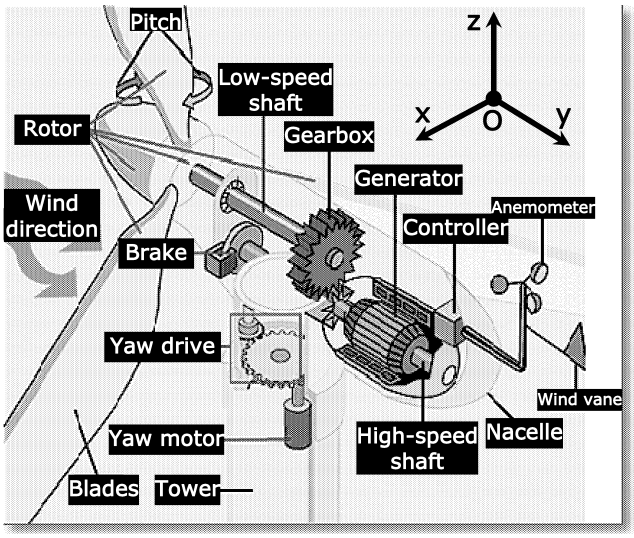


Figure 1. Dissection on a HAWT.

The propeller rotates when the average wind speed is at least 4-5 m/s and should be stopped in case of strong wind (of more than 25 m/s speed). Figure 2 displays the 3 basic operating modes of HAWT. The nominal power (P_N) is provided only in case the wind average speed (V_m) is large enough and varies in range III (between V_N and $V_{\max}$). This is the *nominal* mode. When V_m varies in range I (between $V_{\min}$ and V_{Ω_N}), the generated power is smaller than the nominal one, but increases/decreases nonlinearly. This is the *sub-nominal* mode. Finally, the *nearly-nominal* mode is defined by the nonlinear variation of V_m in range II (between V_{Ω_N} and V_N), being the shortest.

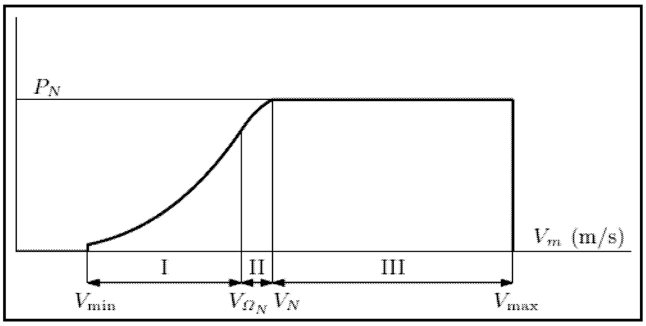


Figure 2. Operating modes of a HAWT.

From the modeling point of view, any HAWT can be expressed by a multi-variable system with several nonlinear blocks, as displayed in Figure 3. Usually, the wind speed is composed by 2 vectors: the average/mean speed ($\vec{V}_m$) and the stochastic speed ($\vec{v}_s$). Their amplitudes and directions can vary, but $\vec{V}_m$ is usually constant on short durations, while $\vec{v}_s$ is more like a colored noise.

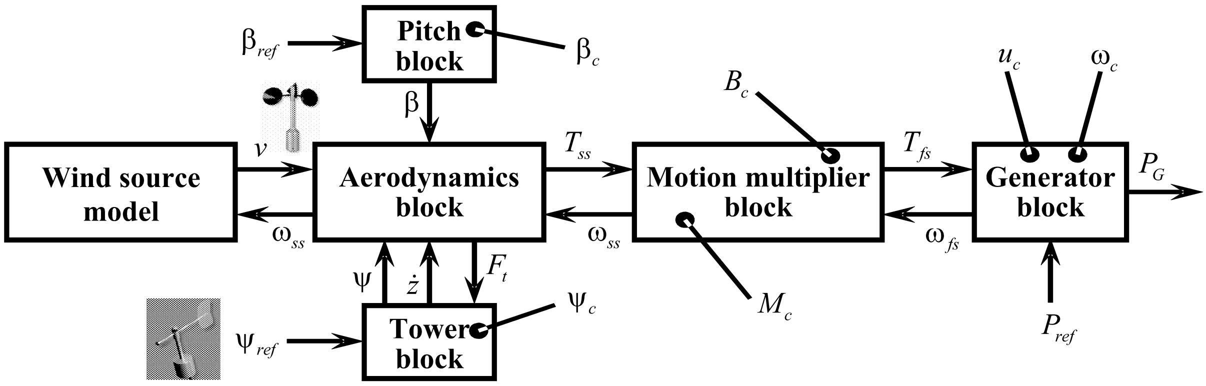


Figure 3. HAWT functional blocks and signals.

Although one cannot know how two wind vectors mix to each other, one relies on the superposition principle. The wind source is actually an exogenous phenomenon that provides the input signal v, i.e. the speed of the wind component pushing on the propeller blades to move. The anemometer can measure v. The aerodynamics block includes the propeller, is quite complex and relies on the tower block. It receives signals like: β (the *pitch angle*), ψ (the *azimuth*), $\dot{z}$ (the tower displacement speed), ω_{ss} (the rotation speed of slow shaft). (Depending on the aerodynamics model, the first and perhaps the second derivative of pitch are also required.) On its turn, it sends signals like: F_t (the thrust force acting on tower, due to impact between the propeller and the wind), T_{ss} (the torque of slow shaft). The pitch block is responsible for blades orientation about the hub (with respect to some reference, β_{ref}), whereas the tower block is in charge with gyration (*yaw*) movement and damping of traction force effects. Gyration is set according to the reference azimuth ψ_{ref} coming from the vane. Normally, towers made of hard concrete, are practically insensitive to wind pushing force. However, as the HAWT height increases, the effect of this force becomes more and more visible. Next, the motion multiplier block, of mechanical nature, includes the coupling, gear box and brake. It establishes relationships between the torque / rotation speed of slow shaft (T_{ss} / ω_{ss}), on one side and the torque / rotation speed of fast shaft (T_{fs} / ω_{fs}), on the other side, when taking into account the friction force generated by the brake. The generator block has electro-magnetic nature and operates with signals like: torque and rotation speed of fast shaft, electro-magnetic torque T_G (internal), generated power (P_G) as main output, reference power from the electrical power grid (P_{ref}). Beside the aforementioned signals, the scheme in Figure 3 illustrates some other interesting signals that could be used to actuate the system: the pitch control signal (β_c), the azimuth control signal (ψ_c), the gear box (coupling) control (M_c), the brake control (B_c) and the generator control signals (u_c, ω_c). Other signals and internal constructive parameters can be mentioned as well.

3. ANALYTICAL EQUATIONS OF USUAL HAWTs

The WINTUS simulator relies on a set of analytical equations expressing interdependencies between the aforementioned signals. Usually, HAWTs are small, medium or large size. The simulator adapts itself to any of such classes. On the whole, 4 main modules are designed: (a) wind source; (b) aerodynamics; (c) mechanical subsystem; (d) electro-magnetic subsystem. Each module includes one or more blocks from Figure 3.

The **wind source modeling** is concerned with stochastic time-frequency descriptions of wind generator and filtering applied by the propeller. Figure 4 illustrates the SIMULINK schemata of wind source model, including the impact with the HAWT propeller. The first wind component is the stepwise profile of wind average, V_m. After analyzing many data sets provided by various anemometers, it can be noticed that wind keeps constant its average for some time interval, before jumping to another constant average for the next time interval. The durations of such steps, as well as their heights, are generated at random, with uniform or Gaussian distributions respectively.

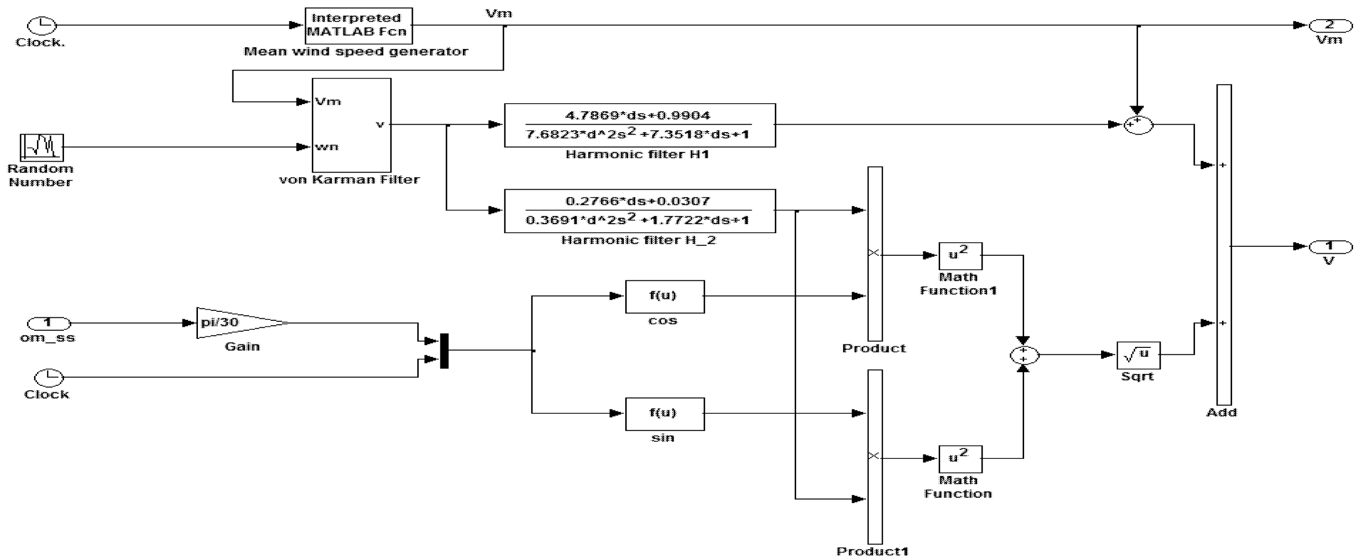


Figure 4. WINTUS: the wind generating source model.

The steps are however grouped around V_{m0} – a constant value of V_m, set by the tower height z and soil asperity type. Any of the two equations below can be employed to define V_{m0}:

$$V_{m0}(z) = V_{m0}\left(z_{ref}\right)\frac{\ln\left(z/z_0\right)}{\ln\left(z_{ref}/z_0\right)} \text{ or} \tag{1}$$

$$V_{m0}(z) = V_{m0}\left(z_{ref}\right)\left(\frac{z}{z_{ref}}\right)^{\alpha}, \tag{2}$$

where the reference height z_{ref} is set to 10 m, and the coefficients z_0 or α are given in Table 1.

Table 1. Soil asperity constants.

Soil type	z_0 [mm]	α
Sand	$\in[0.2, 0.3]$	0.10
Mown grass	$\in[1, 10]$	0.13
High grass	$\in[40, 100]$	0.19
Suburbia	$\in[1000, 2000]$	0.32

Also, $V_{m0}\left(z_{ref}\right)$ has to be set experimentally, depending on the intensity of wind activity.

Stochastic disturbances are added to stepwise profile. Several wind disturbance models were proposed in the literature so far. Perhaps, the most realistic of them is described in (Bianchi et al. 2007). The model in Figure 4 is however improved in several respects. As one can easily notice from the figure, the stochastic speed v_s has 3 components: the filtered Gaussian white noise $v_{s,e}$, the horizontal blade disturbance $v_{s,e,c}$ and the vertical blade disturbance $v_{s,e,s}$. Two phenomena are considered when generating the disturbances. First of all, the main component is a colored noise, obtained after adaptively filtering some Gaussian white noise e_G. A realistic filter is von Karman's one, with the non rational transfer function:

$$\mathrm{H}_{vK}(s) = \frac{K_{vK}}{\left(1 + T_{vK}s\right)^{5/6}}, \tag{3}$$

where both the gain K_{vK} and the time constant T_{vK} are depending on the average speed V_m. A rational approximation of filter (3) is the following:

$$\mathrm{H}_N(s) = \frac{K_{vK}\left(1 + bT_{vK}s\right)}{\left(1 + T_{vK}s\right)\left(1 + aT_{vK}s\right)}, \tag{4}$$

with $a = 0.25$, $b = 0.4$ (Nichita et al. 2002) (also known as *Nichita's filter*). The filter adapts itself to wind speed average, by means of its variable parameters:

$$T_{vK}(t) = \frac{L_v}{V_m(t)}, \quad K_{vK}(t) = \sqrt{\frac{2aT_{vK}(t)\left(1 - a^2\right)}{b^2 - a^2 + a\left(1 - b^2\right)}}, \quad \forall t \in \mathbb{R}_+, \tag{5}$$

where L_v is a constant length set according to HAWT location. Usually, $L_v \in [100, 330]$ m.

The second phenomenon that generates disturbances is the impact between the propeller blades and the air flow. Two aspects were considered here. On one hand, the propeller discontinuity induces the periodical effect of *wind cutting*, expressed by the harmonic filter below:

$$\mathrm{H}_{h,1}(s) = \frac{b_{1,0} + b_{1,1}\phi s}{1 + a_{1,1}\phi s + a_{1,2}\phi^2 s^2}, \tag{6}$$

where ϕ is the propeller diameter (for example, $\phi = 34$ m), while $b_{1,0} = 0.9904$, $b_{1,1} = 4.7869$, $a_{1,1} = 7.3518$ and $a_{1,2} = 7.6823$. On the other hand, the impact between blades and the wind enforces some of the air flow to suddenly change direction, becoming parallel with the propeller plane. This component tremendously depends on the slow shaft rotation speed, ω_{ss} and inserts an additional perturbation as a local turbulence. A second harmonic filter is employed to estimate the part of the air flow that changes direction along the propeller plane:

$$\mathrm{H}_{h,2}(s) = \frac{b_{2,0} + b_{2,1}\phi s}{1 + a_{2,1}\phi s + a_{2,2}\phi^2 s^2}, \quad (7)$$

where $b_{2,0} = 0.0307$, $b_{2,1} = 0.2766$, $a_{2,1} = 1.7722$ and $a_{2,2} = 0.3691$. The influence of rotation speed is then expressed by adding the two instantaneous projections along the central axes of propeller plane (see Figure 4).

In conclusion, the 3 components of stochastic speed v_s are given by the following equations:

$$\begin{cases} V_{s,e,1}(s) = V_{s,e}(s) = \mathrm{H}_{h,1}(s)\mathrm{H}_N(s)E_G(s) \\ V_{s,e,2}(s) = \mathrm{H}_{h,2}(s)\mathrm{H}_N(s)E_G(s) \end{cases}, \quad (8)$$

$$\begin{cases} v_{s,e,c}(t) = v_{s,e,2}(t)\cos\left(\omega_{ss}(t)\cdot t\right) \\ v_{s,e,s}(t) = v_{s,e,2}(t)\sin\left(\omega_{ss}(t)\cdot t\right) \end{cases}, \quad \forall t \in \mathbb{R}_+, \quad (9)$$

where F is the Laplace Transform of f, as usual. It has to be noted that the resulted wind speed, v, is what HAWT *perceives* when being pushed by the air flow. The wind available power can also be estimated:

$$P_w(v) = \frac{\pi\phi^2}{8}\rho_a v^3, \quad (10)$$

where ρ_a is the air flow density.

Modeling the **aerodynamics** means writing all equations that describe the conversion of wind motion into the slow shaft motion. In subsidiary, the influence of variable pitch (and, eventually, of blades geometry) has to be emphasized. Several aerodynamic models can be found in the literature. One of the most interesting and realistic models is based on *Blade Element Momentum* (BEM) Theory, introduced, for example, in (Bianchi et al. 2007), (Manwell et al. 2009) or, more recently, in (Ingram 2011). The BEM Theory relies on two hypotheses: (a) blades are far enough to each other, so that the turbulent interferences can be neglected; (b) each blade can be decomposed into a number of elements (between 10 and 20) with small length and approximately constant chord size, like suggested by Figure 5(a). After computing the local forces applied by the wind on each element, the overall forces are obtained by integration along each blade. The first hypothesis is comfortably verified by the usual 3-blade propeller that can be seen in almost every wind farm. The second hypothesis allowed one to design and implement the WINTUS aerodynamic module of Figure 6.

The computations are mainly driven by the image of instantaneous forces, speeds and angles of Figure 5(b). Two elementary forces are of interest, in order to produce the outputs of aerodynamics module (of Figure 6): *the thrust force* F_t (that pushes the blade and the tower to bend) and *the tract force* F_{ss} (that leads to the slow shaft torque T_{ss}). In order to capture as much as possible of the wind power, F_t should be minimized, whereas F_{ss} should be maximized. As it can be derived from Figure 5(b),

$$\begin{cases} F_t \equiv F_L \sin\varphi - F_D \cos\varphi \\ F_{ss} \equiv F_L \cos\varphi + F_D \sin\varphi \end{cases}, \quad (11)$$

where F_L is *the lift(ing) force*, F_D is *the drag(ging) force* and φ is the *inflow angle* between the tangential speed of current element ($\omega_{ss}r$) and the relative speed (v_r) resulted by summing the tangential speed with the wind speed (v). The lifting is realized along the relative speed, while the blade is dragged on a perpendicular direction. Both lift and drag forces depend on *the attack angle* α, defined by:

$$\alpha \equiv \frac{\pi}{2} - \varphi - \beta. \quad (12)$$

As one can easily notice, the pitch is one of the actors that can lead to the efficient capture of wind power. Moreover, the model adopted here allows setting one pitch for each blade element (if the HAWT is designed in this aim). Obviously, the relative speed is:

$$v_r(r) = \sqrt{v^2 + \omega_{ss}^2 r^2}, \quad (13)$$

and the inflow angle can be estimated by:

$$\varphi(r) = \arctan\left(\frac{v}{\omega_{ss}r}\right). \quad (14)$$

The lift and drag forces are then:

$$\begin{cases} F_L(r) = \dfrac{\rho_a v_r^2(r)}{2} C_L\left(\alpha(r)\right) l_c(r) \\ F_D(r) = \dfrac{\rho_a v_r^2(r)}{2} C_D\left(\alpha(r)\right) l_c(r) \end{cases}, \quad (15)$$

where l_c is the chord length, whilst C_L and C_D stand for the corresponding coefficients depending on the attack angle and Reynolds criterion (Re). Tables of C_L and C_D variations are already known and have been integrated into the model of Figure 6. It is quite important to note that equations (12)–(15) are expressed in terms of current radius r. However, the forces (11) and (15) are considered constant for the whole element length (Δr). If the blade is divided into $N_e \in \overline{10,20}$ elements, the thrust force, as well as the slow-shaft torque:

$$T_{ss}(r) = F_{ss}(r) \cdot r, \quad (16)$$

have to be integrated along the blade (for example, by using the N_e-point trapezium method).

Apparently, the thrust force is useless. Nevertheless, the bending of both the blades and the tower are thrust force effects that have to be accounted. Recall that v is the wind speed with respect to the ground. Or, in the rationale above, v is understood as the speed with respect to the blade element. Let $\zeta(r)$ be the current bending angle of the blade. Then $\dot{\zeta}(r)\cdot r$ is the current bending speed. Similarly, $\dot{z}$ is the speed of tower top (corresponding to the top tower displacement aside the Oz axis – see Figure 1). Since both speeds have the same direction as the wind speed, the following correction has to be applied:

$$v \leftarrow v(r) = v - \dot{\zeta}(r)\cdot r - \dot{z}. \quad (17)$$

The correction (17) is represented within the upper left corner of model in Figure 6.

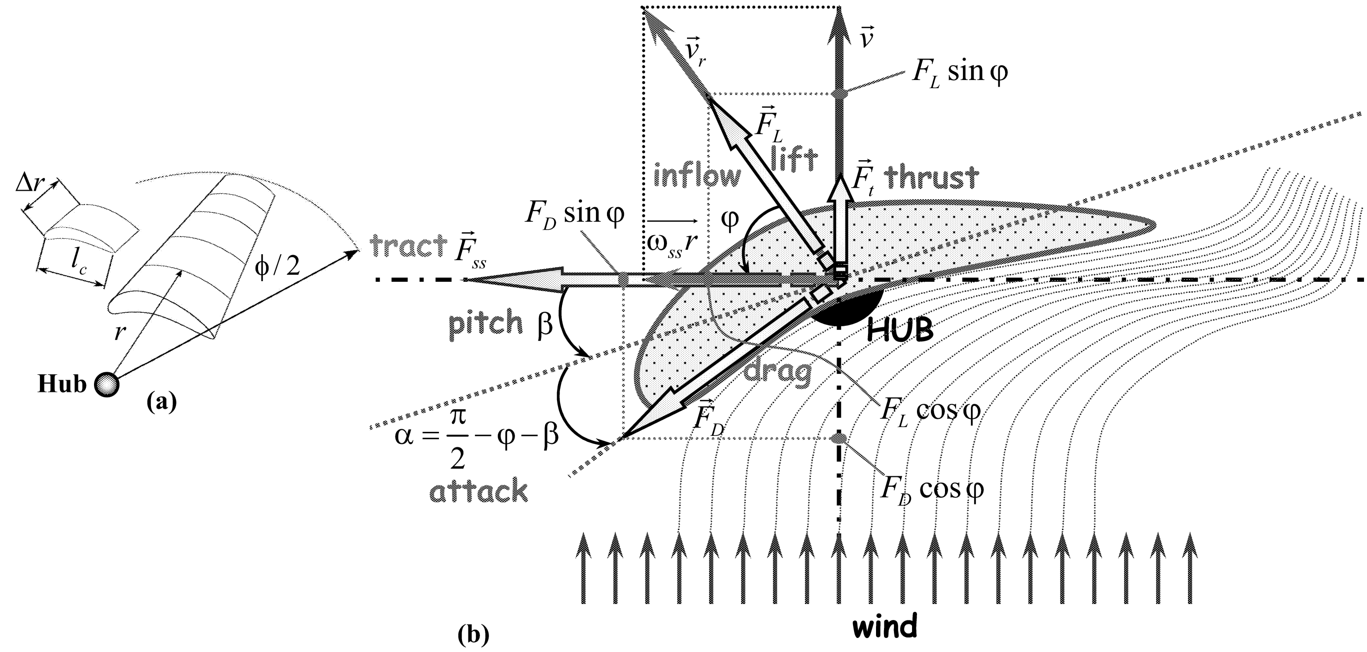


Figure 5. (a) Principle of BEM Theory. (b) Forces and angles on blade element transversal section.

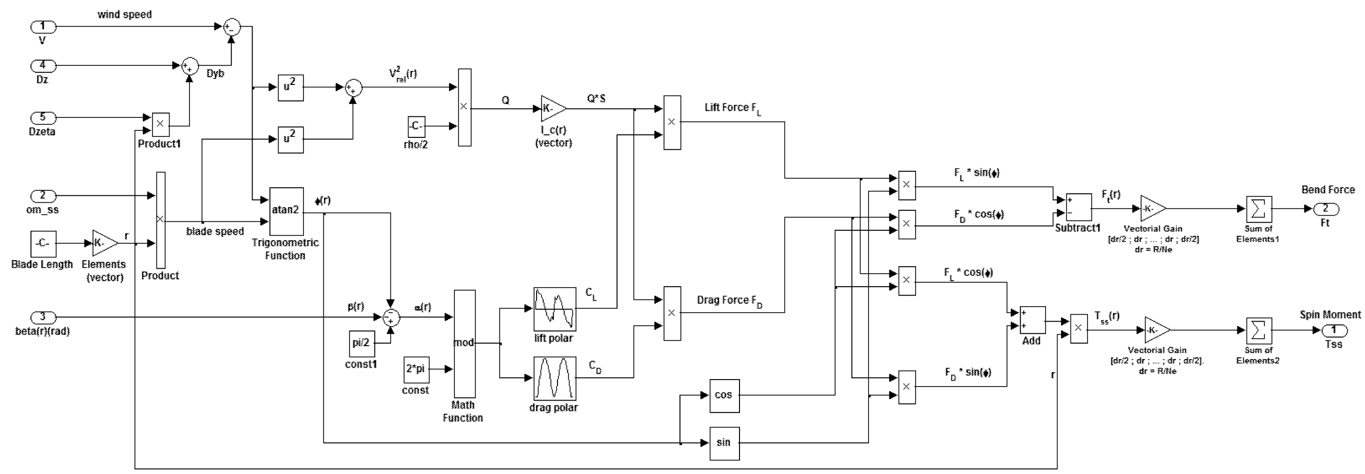


Figure 6. WINTUS: the aerodynamics model (based on BEM Theory).

This version of WINTUS simulator is not concerned with the azimuth ψ. One assumes here that the nacelle is always oriented according to wind direction (either manually or automatically).

The **mechanical subsystem** includes: the gear box, the (hydraulic) brake, the tower engine and its coupling. Additionally, the pitch actuators can join this configuration. A linear analytical model of Euler-Lagrange type can be assigned to this subsystem (Lescher 2006), given that the nonlinear dynamic behavior is only a negligible side effect. The Euler-Lagrange approach leads to the couple of independent state space representations below:

$$\begin{cases} \dot{\mathbf{x}}_i(t) = \mathbf{A}_i \mathbf{x}_i(t) + \mathbf{B}_i \mathbf{u}_i(t) \\ \mathbf{y}_i(t) = \mathbf{C}_i \mathbf{x}_i(t) \end{cases}, \ \forall t \in \mathbb{R}_+, \forall i \in \{1,2\}, \quad (18)$$

where the states and the signals are defined by:

$$\mathbf{x}_1 \equiv \begin{bmatrix} z & \zeta & \dot{z} & \dot{\zeta} \end{bmatrix}^T;\ \mathbf{u}_1 \equiv F_t;\ \mathbf{y}_1 \equiv \begin{bmatrix} \dot{z} & \dot{\zeta} \end{bmatrix}^T;$$

$$\mathbf{x}_2 \equiv \begin{bmatrix} \theta_s & \omega_{ss} \end{bmatrix}^T;\ \mathbf{u}_2 \equiv \begin{bmatrix} T_{ss} & \omega_{fs} \end{bmatrix}^T;\ \mathbf{y}_2 \equiv \begin{bmatrix} T_{fs} & \omega_{ss} \end{bmatrix}. \quad (19)$$

Beside the aforementioned notations, definitions (19) make use of the *torsion angle* between the two shafts (as internal variable), namely $\theta_s = \theta_{ss} - \theta_{fs}$. The dynamical systems (18) are configured by the matrices below:

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{I}_2 \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix},\ \mathbf{B}_1 = \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{M}^{-1}\mathbf{Q} \end{bmatrix},$$

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{I}_2 \end{bmatrix}; \quad (20)$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ -K_s/J_R & -D_s/J_R \end{bmatrix},\ \mathbf{B}_2 = \begin{bmatrix} 0 & -1/N_t \\ 1 & D_s/(N_t J_R) \end{bmatrix},$$

$$\mathbf{C}_2 = \begin{bmatrix} K_s/N_t & 0 \\ 0 & 1 \end{bmatrix}. \quad (21)$$

In definitions (20), the following notations are employed:

$$\mathbf{M} = \begin{bmatrix} m_t + N_b m_b & N_b m_b \phi/2 \\ N_b m_b \phi/2 & N_b m_b \phi^2/4 \end{bmatrix};\ \mathbf{K} = \begin{bmatrix} K_t & 0 \\ 0 & K_b \phi^2/4 \end{bmatrix};$$

$$\mathbf{D} = \begin{bmatrix} D_t & 0 \\ 0 & D_b \phi^2/4 \end{bmatrix};\ \mathbf{Q} = \begin{bmatrix} N_b \\ N_b \phi/2 \end{bmatrix}. \quad (22)$$

Matrices (21) and (22) are defined according to the constructive parameters of Table 2.

Table 2. Constructive parameters of mechanical subsystem.

Symbol	Parameter	Typical values
N_b	Number of blades	3
m_b	Blade mass	3 t
m_t	Tower & nacelle mass	35 t
J_R	Rotor inertial momentum	214 t.m^2
K_b	Blade elasticity coefficient	1 000 t.m^2/s^2
K_t	Tower elasticity coefficient	8 500 t.m^2/s^2
K_s	Transmission elasticity coefficient	11 000 t.m^2/s^2
N_t	Gear transmission ratio	40
D_b	Blade dumping coefficient	10 t.m^2/s
D_t	Tower dumping coefficient	50 t.m^2/s
D_s	Shaft dumping coefficient	60 t.m^2/s

The **pitch actuators** are modeled by second order linear systems with well known transfer functions, such as:

$$\mathrm{H}_p(s) = \frac{\omega_p^2}{s^2 + 2\zeta_p\omega_p s + \omega_p^2}, \quad (23)$$

where ζ_p and ω_p are specific damping factor and resonance pulsation, respectively. For example, $\zeta_p = 0.9$ and $\omega_p = 8.88$ rad/s. The overall pitch model is however nonlinear, because of two saturations. Firstly, the blade can be turned around its axis with a bounded pitch only, like e.g. $\beta \in \left[-2^\circ, 30^\circ\right]$. Secondly, the turning around speed is also bounded. Usually, $\dot{\beta} \in [-10, 10]$ °/s.

Finally, the **electro-magnetic subsystem** refers to the generator and its controller. Basically, 2 types of generators have to be accounted (Munteanu et al. 2008): directly connected to the grid (with ON/OFF controller); indirectly connected to the grid through synchronizers and frequency controllers (with one or several regulation loops). Since the ON/OFF-like shocks are not suitable for standard grids, the second type generators are often preferred. Here, again, two constructive solutions are frequently employed within usual HAWTs (among many others): *Squirrel-Cage Induction Generator* (SCIG) and *Doubly Fed Induction Generator* (DFIG). A configurable DFIG has been integrated in WINTUS. However, after slight modifications of DFIG model, the SCIG model can be obtained, if necessary. Figure 7 depicts the schemata of a DFIG model. The general equation that describes the forces conversion from mechanical to electrical is the following:

$$J_G\dot{\omega}_{fs} \equiv T_{fs} - T_G, \quad (24)$$

where T_G is the electro-magnetic torque opposing to the mechanical torque T_{fs}.

Note that the rotation speed ω_{fs} tremendously depends on the load T_G. In case of DFIG,

$$T_G \equiv 3N_pL_{sr}\left(i_{s,q}i_{r,d} - i_{r,q}i_{s,d}\right)/2, \quad (25)$$

where N_p is the number of pole pairs (usually 2), L_{sr} is the stator-rotor mutual (magnetic) inductance, $\{i_{s,d}, i_{s,q}\}$ are the stator currents and $\{i_{r,d}, i_{r,q}\}$ are the rotor currents (specific to *doubly fed* machines). Also, (d,q) denotes the orthogonal generalized coordinates (*direct* and *quadrature*). Normally, DFIG is a 3-phase machine (with 3 alternative currents for each the rotor and the stator). The conversion to/from (d,q) coordinates can be realized by means of DQO or Park transforms (Blackburn 1993). The (d,q) approach is preferred, in order to reduce the model dimensionality. The 4 currents can be assigned to internal states of a dynamical system, whilst the 4 corresponding voltages $\{u_{s,d}, u_{s,q}, u_{r,d}, u_{r,q}\}$ to its inputs and T_G to the output. Then, the general equations that describe the DFIG dynamics are (24), together with:

$$\begin{cases} \dot{\mathbf{x}} \equiv \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ y \equiv 3N_pL_{sr}\left(x_2x_3 - x_1x_4\right)/2 \end{cases}. \quad (26)$$

The matrices $\mathbf{A}$ (depending on ω_{fs}) and $\mathbf{B}$ are defined by means of DFIG constructive parameters:

$$\mathbf{A} = -\frac{1}{\sigma}\begin{bmatrix} \alpha_s & -\sigma\omega_c - (1-\sigma)\omega_{fsp} & -\lambda_s\alpha_r & -\lambda_s\omega_{fsp} \\ \sigma\omega_c + (1-\sigma)\omega_{fsp} & \alpha_s & \lambda_s\omega_{fsp} & -\lambda_s\alpha_r \\ -\lambda_r\alpha_s & \lambda_r\omega_{fsp} & \alpha_r & \omega_{fsp} - \sigma\omega_c \\ -\lambda_r\omega_{fsp} & -\lambda_r\alpha_s & \sigma\omega_c - \omega_{fsp} & \alpha_r \end{bmatrix};$$

$$\mathbf{B} = \frac{1}{\sigma L_s L_r}\begin{bmatrix} L_r & 0 & -L_{sr} & 0 \\ 0 & L_r & 0 & -L_{sr} \\ -L_{sr} & 0 & L_s & 0 \\ 0 & -L_{sr} & 0 & L_s \end{bmatrix}. \quad (27)$$

The following notations are employed in definitions (27):

$$\alpha_s = R_s / L_s; \quad \alpha_r = R_r / L_r; \quad \omega_{fsp} = N_p\omega_{fs};$$
$$\lambda_s = L_{sr} / L_s; \quad \lambda_r = L_{sr} / L_r; \quad \sigma = 1 - \lambda_s\lambda_r. \quad (28)$$

The 's' and 'r' indices point to generator stator and rotor, respectively. In definitions (28), one can recognize stator/rotor attenuations, as well as inductance ratios. The constructive parameters of DFIG, together with some usual values are listed in Table 3. Normally, DFIG is operating at 380 V(AC) and, if the load is missing, the rated maximum generated power is of 660 kW. The torque is bounded to a maximum amplitude $T_G^{\max}$ – a constructive parameter as well. The generator modeling block is illustrated in Figure 7. The reference power P_{ref} is collected from the grid in order to control the generator. However, the controller is not included into WINTUS, so that the user can make her/his own choice.

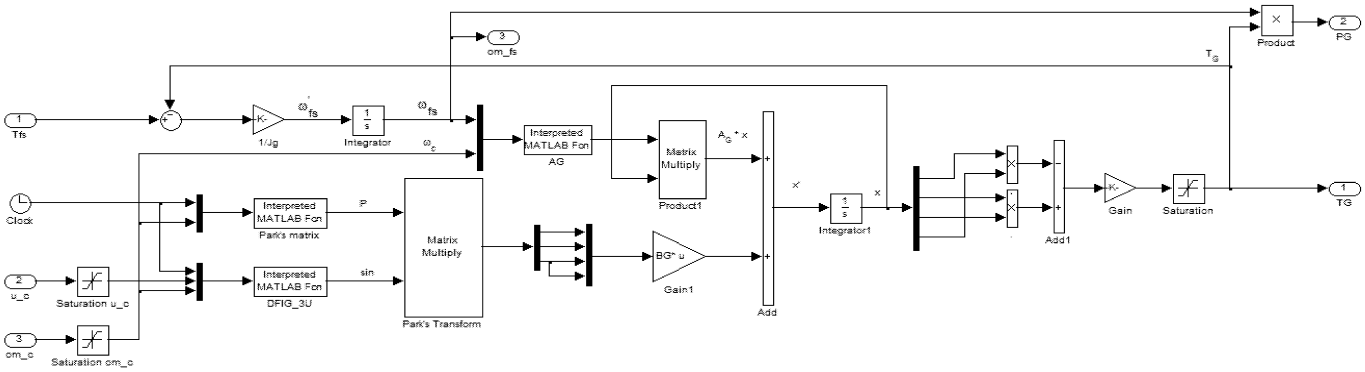


Figure 7. WINTUS: the electro-magnetic model (DFIG).

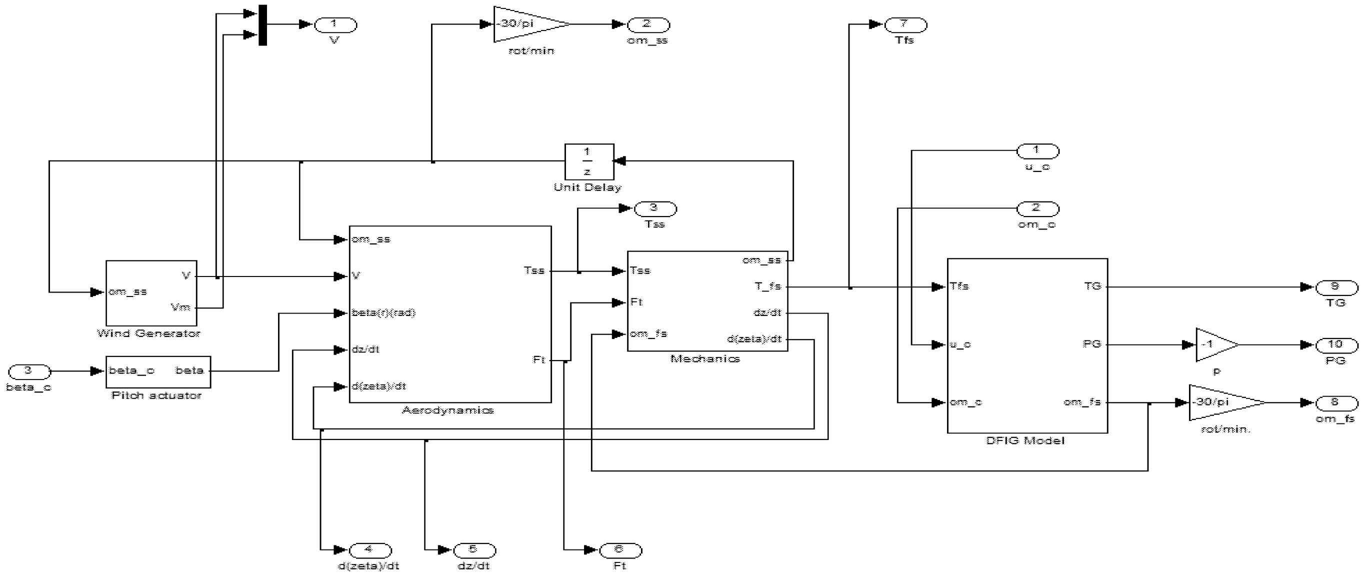


Figure 8. WINTUS: the whole picture.

Table 3. Constructive parameters of DFIG.

Symbol	Parameter	Typical values
J_G	Generator inertial momentum	250 kg.m^2
ω_G	Maximum angular frequency	100π rad/s
R_s	Stator resistance per phase	6.7 mΩ
L_s	Stator inductance per phase	7.5 mH
R_r	Rotor resistance per phase	39.9 mΩ
L_r	Rotor inductance per phase	52 mH
L_{sr}	Stator-rotor mutual inductance	19.4 mH
N_p	Number of pole pairs	2
$T_G^{\max}$	Torque maximum amplitude	4970 N.m

Normally, the DFIG controller is based on a frequency converter that provides the angular frequency ω_c. Moreover, a second controller can provide the voltage amplitude u_c. However, the generated power $P_G \equiv T_G\omega_{fs}$ is more sensitive to ω_c than to u_c. The two control signals are employed to compose the 3-phase stator voltages before entering the Park transform. Thus:

$$u_i(t) = u_c(t)\sin\left(\omega_c(t)t + \theta_i\right),\ \forall t \in \mathbb{R}_+,\ \forall i \in \overline{1,3}, \quad (29)$$

where $\theta_1 = 0$, $\theta_2 = 2\pi/3$, $\theta_3 = -2\pi/3$.

4. WINTUS AT A GLANCE

After all modules being assembled together, resulted the WINTUS simulator in Figure 8. Beside the wind signals v and V_m (uncontrollable), the inputs are the control signals β_c, u_c and ω_c only (also see Figure 3 again). No other controls are possible, as the corresponding mechanisms are not yet modeled. The 9 outputs are: ω_{ss}, T_{ss}, $\dot{z}$, $\dot{\zeta}$, F_t, T_{fs}, ω_{fs}, T_G, P_G. Together with reference signals, the outputs signals are crucial when designing optimal controllers (linear or not).

5. SIMULATION RESULTS

The simulator was tested for stability and robustness first. Steps signals were set for all three inputs, at the levels: $\beta_{c0} = 1^\circ$, $u_{c0} = 380$ V, $\omega_{c0} = 29\pi$ rad/s. The wind speed variations v and V_m are depicted in Figure 9. One can notice the random variation of average speed V_m steps. As Figures 10 clearly reveal, in response to such stimuli, WINTUS acts like a stable and quite robust system, without any controller, which is not surprising at all. In real world, usual HAWTs can work very well (though not optimally) in absence of automatic control loops.

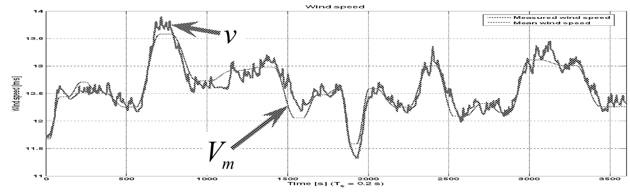


Figure 9. Wind speed profiles.

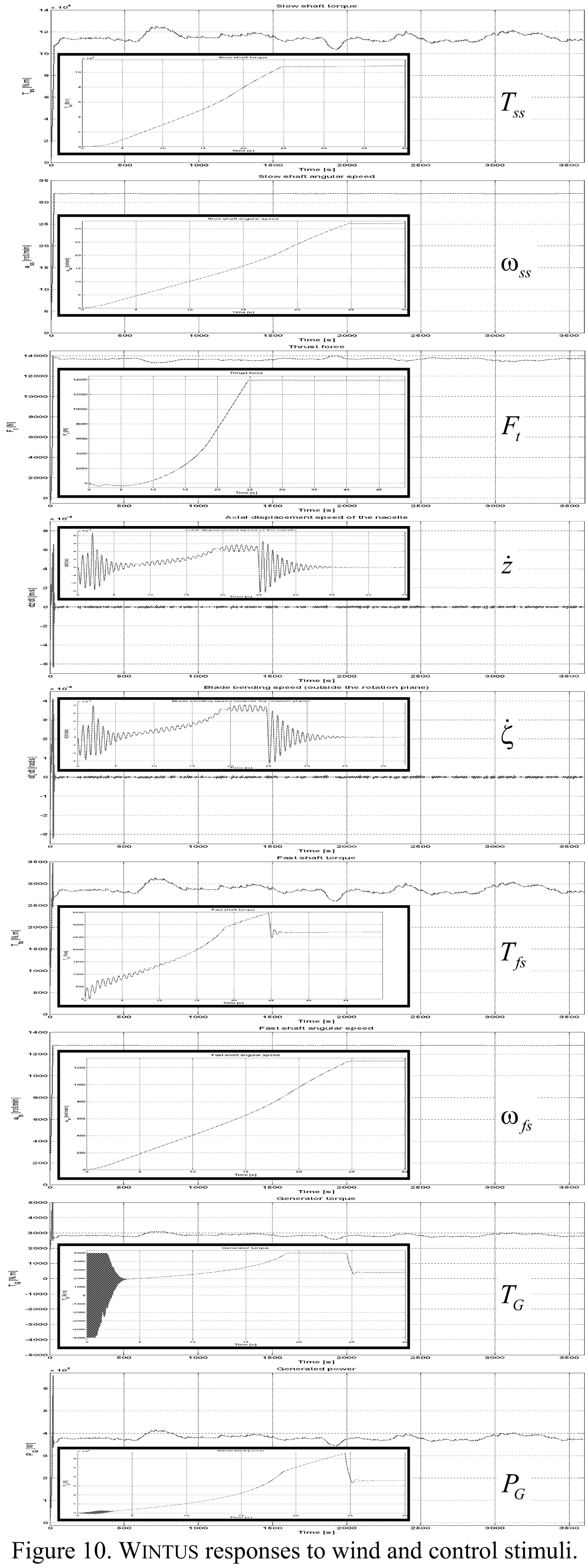


Figure 10. WINTUS responses to wind and control stimuli.

Two types of dynamic modules are combined within usual HAWTs: slow (with mechanical nature) and fast (with electrical nature). The insets inside Figures 10 zoom on the transitory part of each response. One can see that, when aggregated together, the modules are adapting to each other and seem to return an overall time constant between 25 and 30 seconds. Conspicuously, the 3 torques and the generated power are enforced to follow the wind speed profile, while the angular speeds are almost constant. The approximate generated average power is of 380 kW, for an average speed roughly estimated at 12.5 m/s. Other simulations have shown that the maximum rated power (of 660 kW) is obtained for average speeds superior to 22 m/s.

6. CONCLUSION

This article introduced WINTUS – a flexible and yet realistic simulator of usual HAWTs. The design is modular and the analytical equations flow is based on the causality principle. The BEM Theory was successfully employed. Although some functional features were not modeled (like the azimuth orientation, the braking system or the gearbox), the simulator is sufficiently complex to allow connections to other systems, such as automatic controllers.

REFERENCES

Ackerman T., 2005. – *Wind Power in Power Systems*, John Wiley & Sons Ltd., U.K.

Bianchi F.D., de Battista H., Mantz R.J., 2007. *Wind Turbine Control Systems: Principles, Modeling and Gain Scheduling Design*, Springer, London, U.K.

Blackburn J.L., 1993. *Symmetrical Components for Power Systems Engineering*, Marcel Dekker, N.Y., U.S.A.

Burton T., Sharpe D., Jenkins N., 2001. *Wind Energy Handbook*, John Wiley & Sons, U.K.

Hooft E.J., Engelen T.G., Pierik J.T.G., Schaak P., 2007. *Real Time Process Simulator for Evaluation of Wind Turbine Control Systems – Modeling and Implementation*, Research report #ECN-E-07-046.

Ingram G., 2011. *Wind Turbine Blade Analysis using the Blade Element Momentum Method*, Research report, Durham University, U.K.

Jain P., 2011. *Wind Energy Engineering*, McGraw Hill, U.S.A.

Lescher F., 2006. *Commande LPV d'une éolienne à vitesse variable, pour l'optimisation énergétique et la réduction de la fatigue mécanique*, Ph.D. Thesis, Central School of Lille & University of Science and Technology of Lille, France.

Manwell J., McGowan J., Rogers A., 2009. *Wind Energy Explained. Theory, Design and Application.* John Wiley and Sons Ltd., Chichester, U.K.

Munteanu I., Bratcu A.I., Cutululis N.A., Ceanga E., 2008. *Optimal Control of Energy Systems – Towards a Global Approach*, Springer Verlag, London, U.K.

Nichita C., Luca D., Dakyo B., Ceanga E. 2002. Large band simulation of the wind speed for real time wind turbine simulators, IEEE Transactions on Energy Conversion, Vol. 17, No. 4, pp. 523–529.

Pao L.Y., Johnson K.E., 2011. Control of Wind Turbines – Approaches, Challenges and Recent Developments, IEEE Control Systems Magazine, No. 4, pp. 44–62.

CRYOGENIC HELIUM INSTALLATION SIMULATION MODEL: APPROACHES TO IMPLEMENTATION AND USAGE

Vladimir Shpakov
Saint-Petersburg Institute for Informatics and Automation RAS
199178, 14-th Line, 39, St.-Petersburg
Russia
E-mail: vlad@iias.spb.su

KEYWORDS
Production, Hybrid simulation, Rule based Approach.

ABSTRACT

The structure of an installation for liquid helium production and its constituent interactions are described. A conclusion is drawn that an approach for interacting hybrid process specification is most applicable to the installation simulation model implementation. A short description of this process specification approach is presented and possible ways of simulation models implementation are discussed. The approach is based on using of transitive model of processes. The process current state is presented with sets of logic and real variables. The state evolution is assigned by transitive rules. It is shown that the approach is convenient to specify and realize interactions between processes of different dynamics. Possible methods of using the simulation model for teaching students and training operators are considered.

INTRODUCTION

A modern manufacturing system is a collection of automated modules (technological, transport, store, robotic and others) works of which are controlled and coordinated with shop floor control system to realize a production process assigned. Some principal production process properties may be pointed out. First, the whole process is a set of concurrent sub processes related to the individual modules. Some of these concurrent processes are discrete-event and others are continuous, some of them are independent others are interdependent and mutually influential. So most of modern manufacturing systems may be represented with a collection of pure discrete-event, pure continuous and hybrid processes. Most distinct representatives of hybrid systems are automated manufacturing installations, used, for example, in chemical and food industries. Helium liquefier is a typical representative of such system. It contains many different apparatus and devices. Some of them (automatic valves, for example) can be currently at one state of some finite set of states and discretely change it in accordance with definite events. Others evolve their states continuously for most time but can also change them by leaps under some situations or discretely change their mode of work after what they go on to change their state continuously but a character of the change (dynamic) differs from previous one. Some of the devices are interacting, so that current state of one may depend on the state of the other.

Analytical investigations of such complex systems are very difficult so their modelling and simulation are as far main means for analysis and development of production processes. There are many techniques for discrete event and hybrid systems modelling and simulation. Typical modelling techniques are finite state machines, hybrid automata, predicate calculus, real-time temporal logic, timed automata, timed Petri nets and others. There were proposed a variety of models constructed by unification of different computational models. Associations are done to supply the model with desirable properties. Main demands of process specifications are the following. The process specifications must be executable and procedures, based on them must be effective. The last demand is comparatively easy to satisfy thanks to high effectiveness of modern processors. Expressiveness and mistake proofness of the specifications must be as high as possible. Unfortunately, the last two demands are as a rule mutually contradictory. In this paper we describe cryogenic helium installation as an example of hybrid systems and consider a rule based approach to specification and implementation of the system simulation model. The approach has increased mistake proofness and expressiveness, which is sufficient for the purpose of real manufacturing plants simulating. It is shown that the approach is suitable for implementation both of automatic and hand modes of systems control.

THE SIMULATION MODEL DESCRIPTION

Helium liquefier is a heat engineering system consisting of heterogeneous equipments and having a complicated scheme of technology connections. The system behavior is determined by the collection of parallel interacting processes of different dynamics (continuous, discrete-event, hybrid). Simulation model of such a system is a convenient way of presentation of scientific and heuristic expert knowledge of physics and helium liquefying production processes, and of the processes control strategy. A mnemonic circuit of the installation model (a copy of the screen form) is shown in Figure 1. The essential equipments and fittings (turbo-expanders, oil pump, electric heater, compressors, receiver, gasholder, heat exchangers, containers, throttles, faucets and valves) are presented in the circuit. Positions of pressures, temperatures and levels measuring are indicated. At that, a table of symbols is just the same as one used in real installation ОГ-100. In the left part of the scheme control organs of the installation are situated. There are buttons for starting ("Start") and halt ("Stop") of the liquefier. Under the buttons, there are panels for working mode choosing, for servo loops controlling and for emergency conditions indicating. At the last, the emergency conditions are shown with gray color of the font. When originating one of the emergency conditions the color of the font is changed for red.

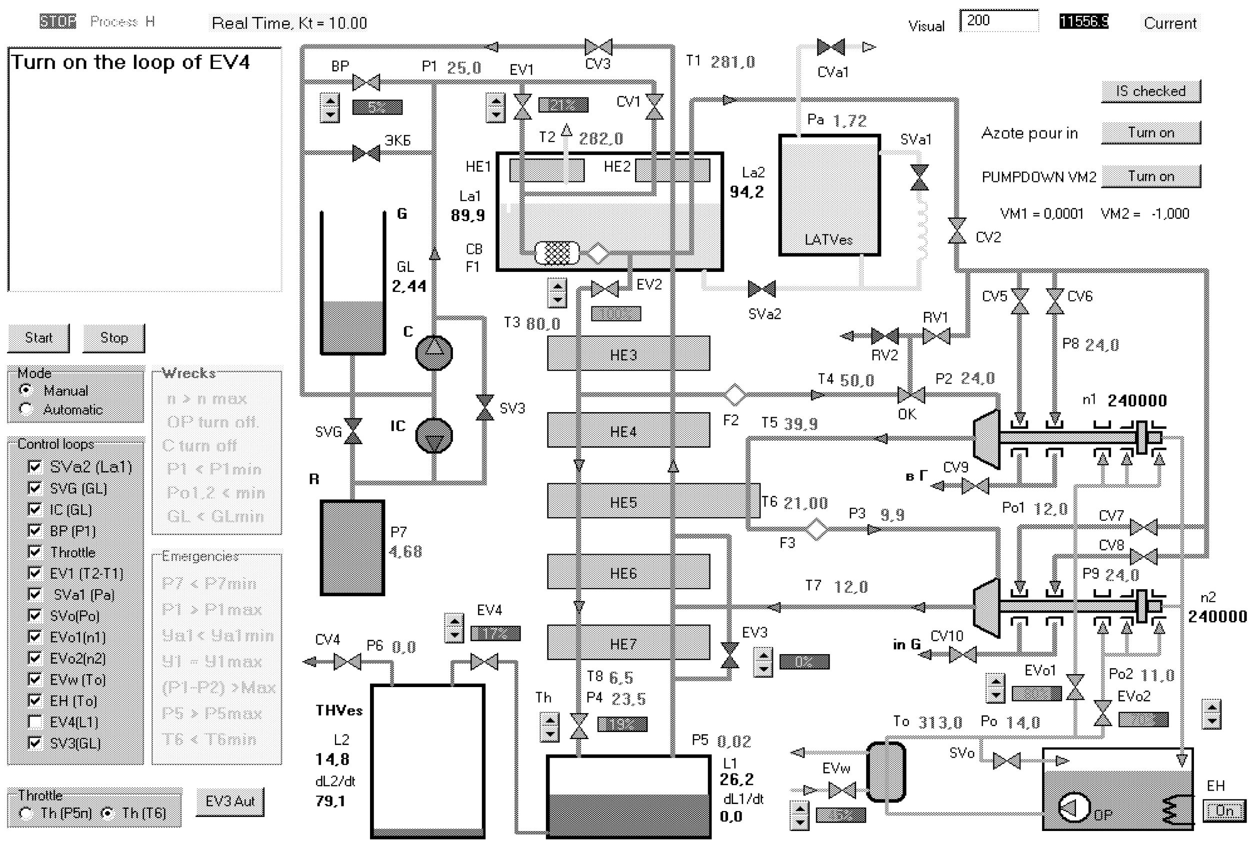
The liquefier simulation model in real or model time determines and shows changes of all the installation parameters (pressures, temperatures, cryogenic liquid levels). It reproduces states of fittings and machines in all stages of an operational cycle: launch, cooling, liquefaction, liquid helium pouring into Dewar vessels and the installation stopping. The model makes it possible to imitate different deflection from a main mode of operation and malfunctions, to reproduce real protections including machine equipment shutdown.

A mode of initial state setting, hand-operated and automatic modes of production process control are realized in the model. When automatic mode is chosen the installation control carries out completely automatically: control loops and necessary equipment switch on and out, faucets and valves open and close. In hand mode control actions are performed by the mouse clicks on corresponding equipments or on their control organs. In such a way machines starts and stops are executed, valve drives switch on, reverse and switch out. An information support of operator acting is provided in hand mode. If the support is turned on then when a situation demanding the operator action is occurred a text message about the necessary action is appeared in the instructions window. The window is situated in the left upper part of the scheme.

A capability of simulating supernumerary situations originating and processing is realized in the model. The possible supernumerary situations are divided into two categories: wrecks and emergencies. It was done because algorithms of processing the situations of various categories differ from each other essentially. Lists of the supernumerary situations are represented in windows which named "Wrecks" and "Emergencies". When an emergency originates an algorithm of scheduled stop is performed. In automatic mode this algorithm fulfills automatically. In hand mode instructions for the operator are appeared and the algorithm is realized by his commands. When wreck occurs a wrecking stop algorithm always triggers.

A RULE BASED APPROACH TO SPECIFICATION AND IMPLEMENTATION OF THE SYSTEM SIMULATION MODEL

Real production interacting processes almost always are hybrid ones that is they have discrete and continuous components. A base of the approach is an abstract model of hybrid automaton (*Henzinger T.A.*, 1996), in which discrete states are represented by logical variables. A set of logical variables contains also a subset of predicates of the continuous states. The predicates are used to simulate interactions between continuous and discrete-event components of the processes. The logic variables can determine both discrete changes of continuous component and changes of their evolution dynamic that is modes of hybrid processes.



Figures 1: A Mnemonic Circuit of the Installation Model.

Thus a state of the processes model may be represented by a set of real variables X for the continuous components and a set of logic variables W for the discrete components. The last must includes a subset Q for the discrete states and a subset G of predicates. It is convenient to separate subsets of independent variables for representation of external impacts and inputs: Σ – for logic and X_i – for continuous ones. So to represent a collection of interacting hybrid processes we must have $W = \Sigma \bigcup Q \bigcup G$ and $X = X_i + X_s$, where X_s is a real subset for continuous states.

To specify processes it is necessary to define transition functions of the following types:

$\sigma : W \to Q \times \{False, True\}$ — function of discrete state transitions;

$\delta : W \times X \to X_s$ — function of continuous state transitions;

$\gamma : X \to G \times \{False, True\}$ — predicate value dependence of continuous process states.

The approach expressiveness and effectiveness of the model implementation essentially depend upon specific forms of these functions. A definitional domain of the transition function σ may be represented with help of logic formulae that makes the representation rather obvious. In the general case, the model specification may need whatever logic formulae. To our mind as the condition it is convenient to use an elementary conjunction of the logic variables. Such conjunctions may be understandably interpreted as logic-dynamic situations [Shpakov, 2004]. The situation S_j may be defined as following:

$$S_j = s_{j_1}, \ldots s_{j_i}, \ldots, s_{j_n}, \text{ where } s_{j_i} = w_{j_i}, \text{ or } s_{j_i} = \neg w_{j_i},$$
$$w_{j_i} \in W,\ n = 1 \ldots N_w, N_w = |W|.$$

Designating the set of situations S the type of the transition function of discrete states may now be defined as following: $\sigma : S \to Q \times \{False, True\}$. This function may be assigned with help of a collection of production rules "*condition* $\to$ *action*". In the rules as conditions are used the logic-dynamic situations and as actions are used procedures which assign necessary values to discrete states defined. At that the rule takes the following form:

$$S_j \to r'_{j_1}, \ldots, r'_{j_i}, \ldots, r'_{j_m}, \qquad (1)$$

where $r'_{j_i} = q'_{j_i}$ or $r'_{j_i} = \neg q'_{j_i}$, $q'_{j_i} \in Q$.

In (1) a convention is used which is accepted when processes are specified. In accordance with the convention state identifiers with strokes designate the states which immediately follow the states designated with the same identifiers but without strokes. It means that at first the situation defined at the conditional part of the rule originates and then the states from the right hand executive part of the rule take the specified values. To implement an arbitrary logic formula it is necessary to use several rules with different conditions and the same executive parts. It is a realization of a normal disjunction form which enables to realize any logic formula. In the rules (1) as the action may be used a procedure which displays some text information corresponding to the situation of conditional part of the rule. It may be a hint for an operator or useful prompt for student.

The function δ must for every mode determine new values of the process states which correspond with a current process state, the system dynamic and also with current states of some processes interacting with this process. The hybrid process mode may be naturally represented by a logic situation considered above. It was proposed to find the new values of the process states by calculation of their transitive relations with the current states (Alur R., 2000). Algorithms of the transitive relations calculations for some elementary processes (integration, differentiation, smoothing and others) are presented in (Shpakov, 2006). Complicated processes are implemented by different connections of the elementary ones and using arithmetic operations and functional procedures. In this case the transitive function δ may be represented with help of a collection of following rules:

$$S_j \to x'_k = \tau_k(x_k, x_i) \qquad (2)$$

where $S_j \in S$ — the situation, corresponding to the mode, τ_k — transitive relation, $x_k \in X_s, x_i \in X$ — the continuous variables.

Since the hybrid process modes very often depend on the process states being at some assigned ranges it is convenient to represent the function γ by a collection of following rules:

$$((x_{j_1} \geq a_k + x_{j_2}) \wedge (x_{j_3} \leq b_k + x_{j_4})) \to g_k \qquad (3)$$

where $x_{j_1}, x_{j_2}, x_{j_3}, x_{j_4} \in X$, $g_k \in G$, a_k and b_k - constants, corresponding to the range. More complicated predicates may be received by using simple ones and rules (1).

A computer implementation of the processes specified by rules (1, 2, 3) is executed with help of an interpreter of these rules. The interpreter architecture and algorithms of functioning are rather simple. The sets of process states variables (X and W) are represented in the interpreter with arrays of records. Each record contains a variable name and value. All rules are implemented with help of condition operators "`if…then…`". Both parts of the operator contain lists of records. Each record contains the array identifier, the variable index and name, a current or assigned value and a pointer to the value. The interpreter base is an executing procedure which scanning lists of rules in a cycle. The processing algorithm calculates the value of the rule condition part. If this value is equal to `True` then a procedure of the executive part of the rule is triggered. In case of the rule (1) specified logic values assign to the corresponding variables from the executive part of the rule. This rule can also be used for realization of instructions giving out when hand mode simulating. In this case the procedure shows a text of instruction corresponding to the current situation. In case of the rule (2) the procedure calculates a new value of the process state and assigns it to the corresponding variable. In case of the rule (3) values of the inequalities, pointed out in conditional part, are calculated and a value of their conjunction assigns to the predicate from the executive part. When using the interpreter the process specifications in the form of rules (1, 2, 3) can be directly executed.

For implementation of this helium liquefier simulation model there were used 335 rules (1), 225 rules (2) and 121 rules (3). Duration of scanning the whole of the rule base is not more than 300 microseconds when using modern personal computer.

A programming of dynamic system simulation models based on use of transition rules (1, 2, 3) has a number of advantages in comparison with using of universal programming languages. First of all a transmission of knowledge from domain and problem experts to programmers is essentially facilitated. Then a usage of the rules enables to shorten the time of the model development. The program turns out more obvious. The model modification, extension and validation are simplified.

A USAGE OF THE MODEL FOR CRYOGENIC TECHNOLOGY LEARNING AND TEACHING

Cryogenic specialty students and operating personnel teaching and training directly using the operating facilities are practically impossible. It is because of high cost of installation startup and of impossibility to reproduce emergency situations, without danger to hurt the equipment. Another problem consists in continuity of the processes, which cannot be stopped or paused. All it involves severe difficulties and embarrasses the teaching and learning processes.

The difficulties pointed out may be overcame when teaching thanks to using liquefier simulation model as a trainer-simulator. A simulation model of industry helium liquefier with turbo expander (ОГ-100) was worked out in Kapitsa Institute in common with SPIIRAS. The model is successfully used for students teaching in corresponding departments of a number of Russian universities. It is also used as a supplement to the instruction for the installation operators.

There provides for three the simulator working modes: demonstrational, teaching and check. In demonstrational, automatic mode the simulation enables to introduce students to helium liquefying different stages, and to purpose and function of separate apparatus and machines. At the concluding stage of the teaching, the automatic mode may be used to introduce students to cryogenic installation automation main principles and works of servo loops.

In teaching, supervisor mode, a student can peculiarly work as a cryogenic installation operator. At that, the mode practically eliminates the possibility of making faults because before any action the proper hint appears in the instruction window. Another version of the mode has no hints but in this case, the process is suspended before every necessary action and resumed only if there was done a proper action.

To simplify student task at the first stages of learning some servo loops may be switch on. It enables the students to attract attention to basic regularities of cryogenic installation work. At the next stages number of working servo loops may be cut down, that compels the student to increase his attention and to conceive the working algorithm as a whole.

It is advisable to use the hand mode as a control one (including self-control). When running it there is a real danger to create an emergency situation in case of wrong student actions since in this mode in contrast to supervisor one the processes do not suspend but respond to student actions sufficiently.

The teacher is able to hide the instruction window and show it again at any stage of the processes. The teacher can use this facility depending on an extent of the subject learning and the experience acquired. Furthermore the processes may be paused at any moment. It enables the teacher to give necessary explanations and the student to understand interdependences of changing parameters and to think over his next actions.

CONCLUSIONS

While developing the model described we came to conclusions that the rule and transitive model based approach is rather good for effective computer realization of big installation simulation model and has expressiveness quite sufficient for specifying interacting hybrid production processes. We also make certain that the logic-dynamic situation interpretations of the process state evolutions are intuitively understandable and very convenient for transmitting of knowledge from an expert to a programmer.

ACKNOWLEDGEMENTS

The present research was supported partly by project funded by grants 12-01-00015 of the Russian Foundation for Basic Research.

REFERENCES

Alur, R., T.A. Henzinger, G. Lafferriere, G. J. Pappas. 2000. Discrete Abstractions of Hybrid Systems, *Proceedings of the IEEE.* No. 88.

Henzinger, T.A. 1996. The Theory of Hybrid Automata, *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS 96).*

Shpakov, V. M. 2006. Dynamic knowledge specification on the base of continuous process transition model, *SPIIRAS Proceedings,* issue 3, vol. 1, 191-197 (in Russian).

Shpakov, V.M. 2004. Executable specifications of production processes transition models, *Mechatronics, automation, control,* No. 3, 183-89 (in Russian).

AUTHOR BIOGRAPHY

SHPAKOV VLADIMIR was born in St. Petersburg (Leningrad), Russia (USSR). He graduated from St. Petersburg National Research University of Information Technologies, Mechanics and Optics. Then he worked in a research institute where he was occupied with development of automatic control system. Since 1985 he is a senior researcher, laboratory of integrated systems for automation, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS).

IMPROVING THE PERFORMANCES OF AN ELECTRO-HYDRAULIC SERVO VALVE BY AN ADDITIONAL POSITION FEEDBACK

Bogdan Mihalescu
Constantin Călinoiu
Nicolae Vasiliu

University POLITECHNICA of Bucharest
313, Splaiul Independenţ ei, Bucharest, Romania, 060042
E-mail:{bogdan|calinoiu|vasiliu}@fluid-power.pub.ro

KEYWORDS
Electro hydraulic servo valve, hybrid feedback, numerical simulation, experimental identification

ABSTRACT. The paper presents the mathematical modeling, numerical simulation and experimental identification of a mechanical feedback electro hydraulic servo valve with an additional electric feedback. The spool position is measured by an LVDT and fed to an internal servo controller in order to improve the steady-state behavior around the null point. The dynamic behavior is also improved by this feedback. The contribution of the position feedback was studied first time by (Mare 1994) using a General Prediction Control algorithm implemented on a 386 PC with a Schaevitz data acquisition card. This paper discusses various aspects of the servo valve performances connecting them with the design. The main target of the work is to evaluate the real improvements which can be gained by an extra feedback loop using the modeling and simulation environment AMESim. The theoretical conclusions were validated by static and dynamic tests performed with a modern test bench at the University POLITECHNICA of Bucharest. The laboratory is certified by the National Accreditation Body (RENAR). The test is showing that the additional feedback is reducing the overall hysteresis of the steady state characteristics, and at the same time is reducing the time constant.

INTRODUCTION

The industrial design and test processes needed for developing a new product must be carried out as quickly as possible and the development cost has to be kept to a minimum, in order to successfully launching onto the market. The classical procedure of designing and building a prototype then carrying out tests is not very competitive. It allows for various design stages to be included unintentionally in the final product. This occurs because in the design stage not all the possible factors are taken into account, no feedback from the client is possible and little testing can be done on the emerging product. This can be corrected by an iterative approach where design, prototype building and testing stages are being repeated until the customer`s demands (both technical and economical) are met. In order to lower the costs of developing a new product, this iterative approach uses inputs not only from design engineers but also from manufacturing specialists, maintenance workers and the customers to which the product is ultimately addressed. In order to integrate all these concurrent aspects more easily, the design and prototyping stages must take place in a virtual environment. This shortens the development phase of a project and in some cases minimizes the costs.

For a hydraulic servo system, delivering a good steady state and dynamic performance is of the utmost importance, because such systems are being employed in domains such as aeronautics, automotive and heavy industry, were the performance demands are very strict.
The performances of any servo valve have a very strong influence over the general performances of the hydraulic system. Servo valves can be regarded as an interface between the hydraulic and the electrical part of an electro-hydraulic servo-system.
In the case of hydraulic servo-systems the modeling and simulation of the servo valve plays a major role in accurately simulating the system and obtaining meaningful results.
Out of the many designs of servo valve developed over the years by various manufacturers we have chosen the most common one, a nozzle flapper (Figure 1) design with a mechanical force feedback.

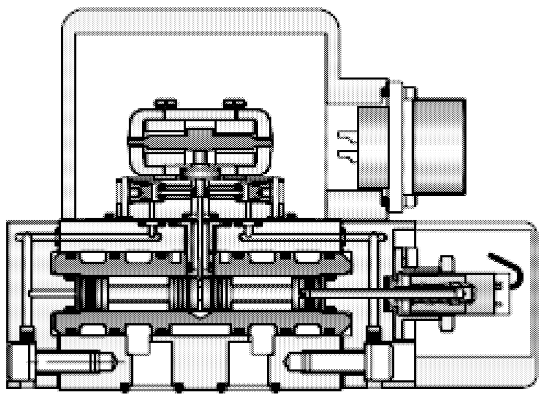


Figure 1 SE2E type servo valve, with additional position feedback from the spool
(by courtesy of PARKER HANNIFIN)

This design, although it has its shortcomings, especially regarding the fluid filtering conditions, is well studied and proven to deliver good performances. It is also technologically easier to manufacture than other designs (ex. jet pipe servo valve).

SERVO VALVE MODELING

Servo valves are complex devices and therefore a mathematical model for them is going to be complex and extremely difficult to integrate in order to conduct a numerical simulation. The modeling process for a servo valve has to be divided among its three main stages.
Each of these subsystems must be modeled by equations that are compatible with each other so that in the end the results can be summed up in one unifying equation and integrated. This task, regarded from a mathematical point of view, is a complicated one, but if individual

components can be modeled separately, then a high degree of accuracy can be obtained. So in a truly concurrent way we can have electrical engineers model electrical components like the torque motor and the electrical drive, mechanical engineers model mobile armature and flexible tube which couple the torque motor to the hydraulic pre-amplifier and hydraulic engineers model the flow of fluid through the pre-amplifier, and the sleeve and spool assembly.

This approach is integrated into the AMESim environment, and the model of the servo valve was developed in the AMESim environment.

The modeling of the torque motor is based on the electromagnetism theory. The two coils and the mobile and fixed armatures form a magnetic circuit. The equations governing its behavior are given in (1) and (2):

$$M_{em}(\theta, i) = K_i \cdot i + K_m \cdot \theta + M_h \qquad (1)$$

Here

M_{em}- electromagnetic torque of the motor (Nm);
K_i - motor`s torque constant (Nm/A);
i - command current (A);
θ - angular displacement of the mobile armature (the angle between the initial and final positions of the mobile armature axis) (rad);
K_m - motor`s electromagnetic constant (Nm/rad);
M_h - motor`s hysteresis couple.

$$u(t) = \frac{1}{2} \cdot R \cdot i + L \cdot \frac{di}{dt} + 2 \cdot K_i \cdot \frac{d\theta}{dt} \qquad (2)$$

Here

u(t) - coils voltage;
L - coil inductance;
R - coil resistance.

Converted to the AMESim environment, we have an electromagnetic circuit (figure 2) which expects an electrical current as input (ports 6 and 3) and as output it has mechanical displacement at ports 1, 2, 4 and 5.

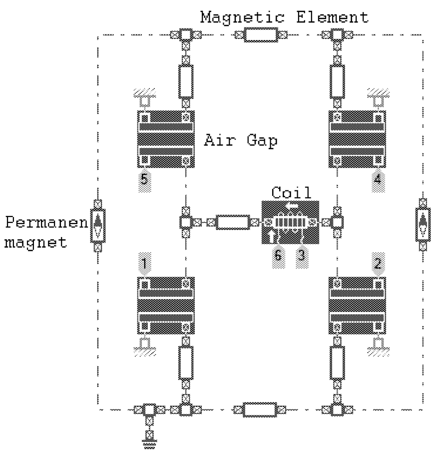


Figure 2 Torque motor super-component in AMESim

The motor`s torque is transmitted through the flexible tube from the torque motor`s mobile armature to the nozzle and flapper system:

$$M_{em}(t) = J \frac{d^2\theta}{dt^2} + \beta \frac{d\theta}{dt} + k_\theta \cdot \theta + M_s(t) \qquad (3)$$

Here

J – mobile armature`s inertial torque;
θ - rotation angle of the mobile armature;
k_θ - elastic tube`s rigidity coefficient;
β - viscous friction coefficient;
M_s - resistive torque.

In AMESim this was modeled as a separate custom component which transforms the torque of the motor into linear movement of the flapper.

$$\begin{cases} f = \dfrac{M \cdot l^2}{2E \cdot I} + \dfrac{F \cdot l^3}{3E \cdot I} \\ \theta = \dfrac{M \cdot l}{E \cdot I} + \dfrac{F \cdot l^2}{2E \cdot I} \end{cases} \qquad (4)$$

Here

f - the sag of the flexible tube`s free end;
θ - the angle of the tubes medium fiber
E - elasticity module of the tube`s material;
I - inertial moment of the tube`s cross section;
F - resultant force acting on the tube;
M - resultant moment action on the tube;

This linear movement drives two pistons which transform the mechanical input of the torque motor into hydraulic terms. The mobile armature movement equation (3) and the flexible tubes equilibrium equations (4) are used at this stage. Also at this level, the mechanical feedback of the servo valve has to be taken into account. The mechanical force feedback is implemented through the means of a steel rod connecting the flapper of the hydraulic pre-amplifier with the spool of the main valve. The value of this force is given by (5).

$$F_{ra} = K_{ra}\left\lfloor X_p + X_s + \theta_p (l_2 - l_1) \right\rfloor \qquad (5)$$

Here

F_{ra} - feedback force (N);
K_{ra} - rigidity of the elastic feedback rod (N/m);
l_2 - the distance between the end of the elastic feedback rod and the axis of the torque motor (m);
X_s - valve spool displacement.

The last factor which needs to be taken into account is the flow of hydraulic oil through the nozzles of the hydraulic preamplifier (6).

$$Q = C_d \cdot A \cdot \sqrt{\frac{2}{\rho}|\Delta P|} \cdot sign(\Delta P) \qquad (6)$$

Here

Q - the flow through the orifice (m^3/s);
C_d - flow coefficient;
A - flow area of the orifice (m^2);
ΔP - pressure drop on the orifice (N/m^2);

The hydro-mechanical model built in AMESim incorporates the equations stated above and models the mobile ensemble (the mobile armature, flexible tube and feedback rod) as a super-component with the input in the form of a torque motor and displacement of the valve spool and an output in the form of displacement of the flapper. The hydraulic component is modeled using small diameter pistons which are used to change the volume of some hydraulic volumes which connect simple fixed orifices, thus a variable pressure drop proportional to the travel of the flapper is obtained.

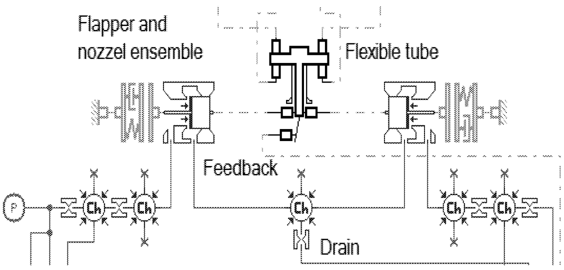


Figure 3 Hydraulic nozzle flapper preamplifier modeled in AMESim

The whole servo valve model is shown in figure 4.

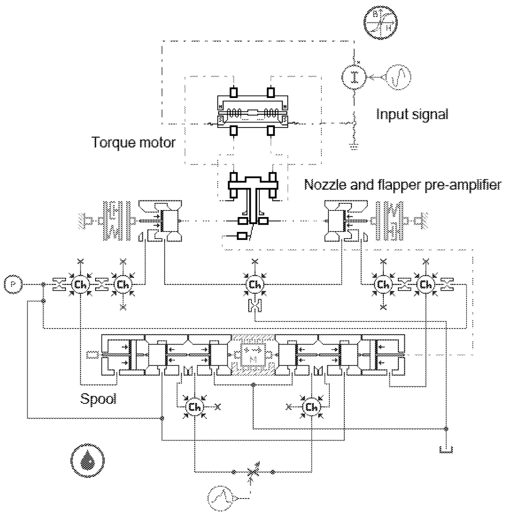


Figure 4 Servo valve model in AMESim environment

The flow from an idealized hydraulic supply represents the input of the system, while the output is the pressure differential that is applied on the ends of the pistons used to model the spool of the valve. The valve`s spool was modeled using standard AMESim components, a series of six pistons with a concentrated mass in the middle. The end pistons are driven by the pressure differential created by the hydraulic preamplifier while the middle four pistons with corresponding hydraulic chamber elements model the spool geometry and four way connections of the valve.

SIMULATION RESULTS

The theoretical steady state characteristic was obtained by numerical simulation. The servo valve was supplied with a low frequency sine signal (0.05 Hz) with adequate amplitude (10 V). With this data, the steady state diagram for this simple servo valve was built (figure 5).

As expected from a high tech flow control valve, the steady state characteristics of the servo valve is very good. However, a small hysteresis is always present. The saturation and the insensibility around the null are very small. Figure 6 presents the steady state characteristic of the valve in the null region obtained for a complete period of sine input.

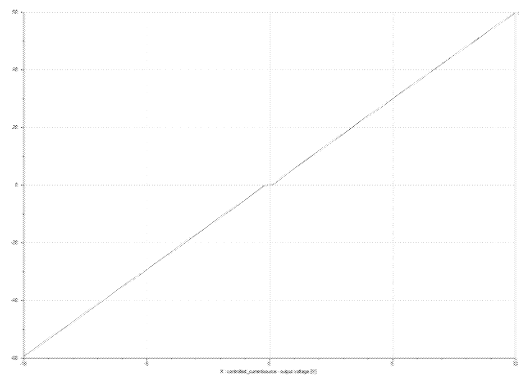


Figure 5 Simulated steady state characteristics for a simple servo valve

It is clear that even if the modeled valve was considered with a zero overlap, the behavior around the null shows a slight insensibility. This insensibility around the null region can lead to problems if the servo valve is being use in high precision positioning system. Therefore an additional feedback loop was considered, the position of the valve`s spool was supplied by a position transducer and the feedback loop was closed electronically through summing blocks. The model of the servo valve with additional feedback loop is presented in figure 7. The simulation was conducted with a signal of the same parameters, sine wave function with amplitude of 10 V and 0.05 Hz frequency. The steady state characteristic is given in figure 8. Figure 9 presents the servo valve behavior around the null.

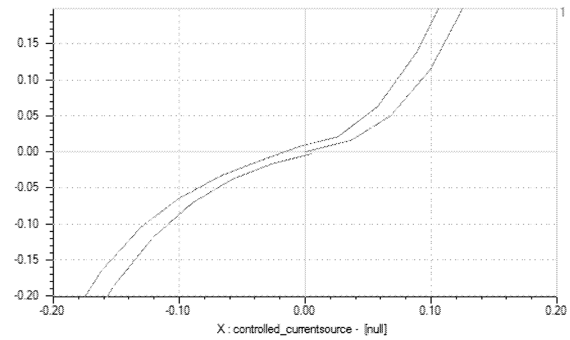


Figure 6 Simulated steady state characteristics around the null

From the two figures we can see that the additional feedback loop has three major effects. First, it dampened the overall gain of the loop, therefore lowering the maximum flow of the servo valve. Second, the hysteresis of the valve was also lowered in comparison with the results from the servo valve without additional feedback. Most importantly, the insensibility around the null was reduced significantly. This can be regarded as the main advantage of adding the electrical position feedback loop, even outweighing the overall damping of the system and the necessity for modifying the structural components of the valve in order to compensate the increase of the spool weight.

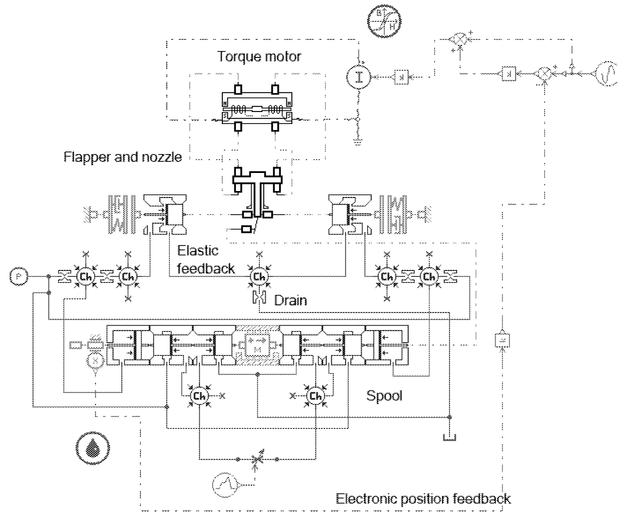


Figure 7 Servo valve model with position feedback loop on the spool

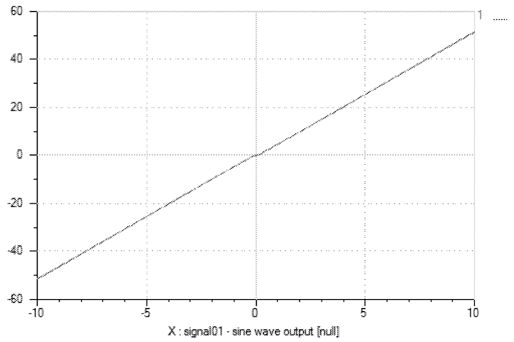


Figure 8 Steady state characteristic for the servo valve with additional position feedback loop

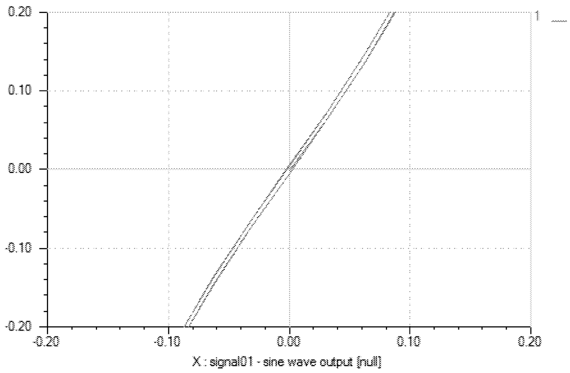


Figure 9 Characteristics around the null for the servo valve with additional feedback

EXPERIMENTAL RESULTS

The experimental test to validate the AMESim model was performed in the Electro Hydraulic Test Laboratory at the University POLITEHNICA of Bucharest. The laboratory is certified by the National Accreditation Body (RENAR) to conduct tests on servo valves and other proportional electro-hydraulic flow and pressure valves. The tests were conducted on a test bench which includes a constant pressure supply unit, a manifold for mounting various valves and a hydraulic cylinder with a very low friction, low inertia piston used to compute the flow in conjunction with a LabVIEW powered data acquisition system. It is clear from the experimental results that the servo valve with the additional feedback loop has low hysteresis and very little insensibility around the null. Therefore it is suited for applications were high precision in positioning is necessary and very small levels of command signal are expected for most of the working time. Also the dynamic performances of such a servo valve are above average, the high cutoff frequency displayed by such valves is due to the improvement in response time obtained by the addition of the electrical position feedback system.

The results from the dynamic tests show that the servo valve has a high cutoff frequency and a very low response time (Figure 13). The servo valve frequency response is very well described by the following transfer function:

$$H(s) = \frac{88.32s + 1100000}{s^2 + 1100s + 1100000} \qquad (7)$$

From it we were able to compute the natural frequency of the valve at 167 Hz and the damping coefficient - ζ=0.524.

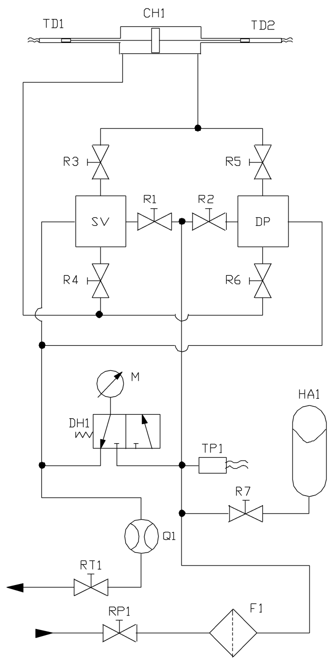


Figure 10 Test bench for the static characteristics of the electro hydraulic servo valves and DDV's

CONCLUSION

The contribution of this research is the simple implementation of a complex servo valve model in the AMESim simulation environment. This creates the possibility of simulating more complex systems including this kind of servo valve. The use of the position additional feedback in the servo valve can be regarded as a redundant feature which increases the system performance too. The paper confirms that the ever closer merging of hydraulics and electronics which we see as a constant trend in proportional hydraulics over the past years is a good way of increasing the control systems performance.

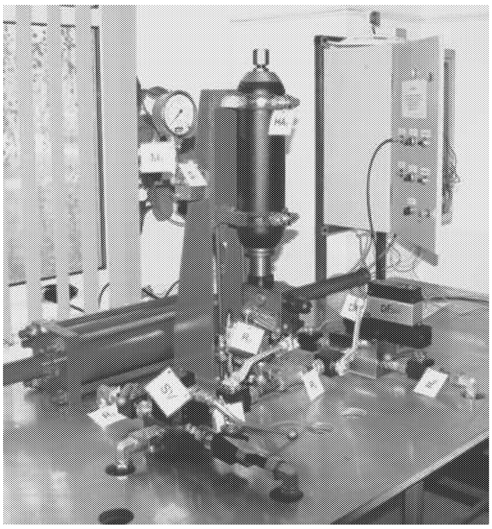


Figure 11 Test bench front view

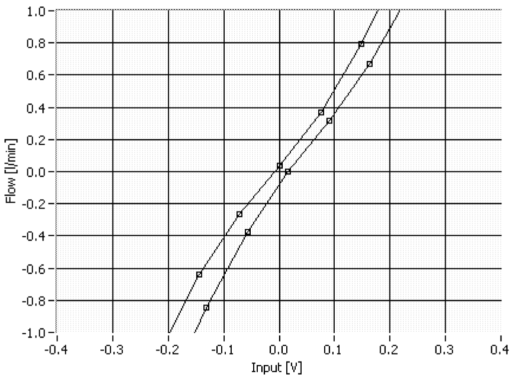


Figure 12 Experimental steady state characteristic around the null for SE2E servo valve

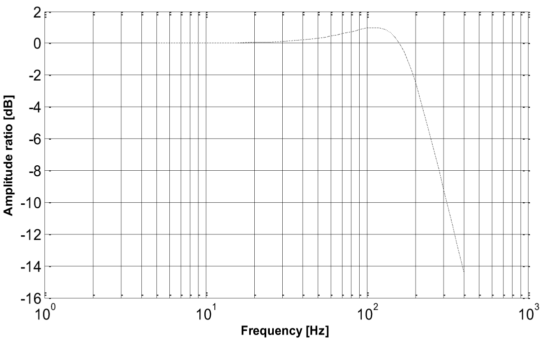


Figure 13 Experimental frequency response of a SE2E servo valve

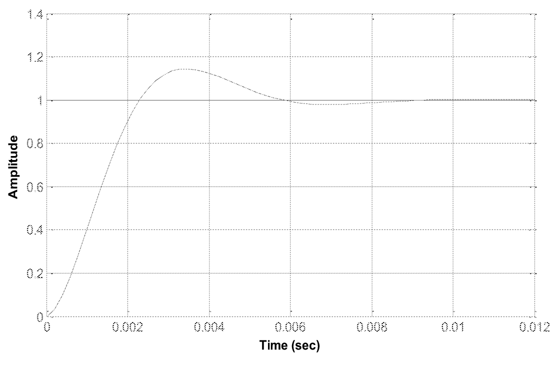


Figure 14 Step input response of a SE2E servo valve

The future work regarding the position additional feedback will be devoted to the use of this type of electrohydraulic amplifier in high precision redundant servomechanisms.

REFERENCES

Mare, J.-Ch., 1994. "*Dynamics of the Electrohydraulic Rotary Servomechanisms*". Doctor Thesis, I.N.S.A. Toulouse.

Lebrun, M., Claude, R. 1997. "*How to create Good Models without Writing a Single Line of Code*". Fifth Scandinavian International Conference on Fluid Power, Linköping.

Vasiliu, N., Vasiliu, D., C.Calinoiu, Mihalescu, B. Negoita, G.C.. 2007-2009. "*e-Learning System for Fluid Power Systems*", Romanian State Grant No.162.

*** LMS INTERNATIONAL, 2011. "*Advanced Modeling and Simulation Environment*", Release 10 User Manual, Leuven.

*** Parker Hannifin. 2004."*Servovalves series SE*". Hydraulic Valve Division, Catalog HY14-1460, Elyria, Ohio, USA.

*** MOOG. 1971. "*Specification Standards for Servo Valves*". Internal Standard 117, East Aurora.

*** MOOG. 1971. "*Performance Estimation for Electro Hydraulic Control Systems*", Internal Standard 126, East Aurora.

BIOGRAPHIES

Bogdan Mihalescu graduated in Hydropower Engineering from University POLITECHNICA of Bucharest in 2007. He became assistant professor and PhD student in the Fluid Power Laboratory from the same university. His main areas of interest include modeling, simulation and dynamic identification of hydraulic systems.

Constantin Calinoiu graduated in Hydropower Engineering from University POLITEHNICA of Bucharest in 1975. In 1980 he became a member of the Hydraulic Lab from the Romanian Aerospace Institute. In 1998 he became PhD and associated professor in the Fluid Power Laboratory from U.P.B. He is working mainly in modeling, simulation, and identification of the hydraulic and electro hydraulic control systems.

Nicolae Vasiliu graduated in Hydropower Engineering from University POLITEHNICA of Bucharest in 1969. He became a PhD in Fluid Mechanics after a research stage in Ghent State University and Von Karman Institute from Bruxelles. He became state professor in 1994, leading the Fluid Control Laboratory from the Power Engineering Faculty of the above university. He worked always for the industry, as project manager or scientific advisor. He was also leading five years the Romanian Innovation Financing Agency.

SIMULATION IN TRANSPORT AND LOGISTICS

A MILLION VARIABLE NONLINEAR TRANSPORTATION PROBLEM

William Conley
Austin E. Cofrin School of Business
480B Wood Hall
University of Wisconsin at Green Bay
Green Bay, Wisconsin 54311-7001 U.S.A.
conleyw@uwgb.edu

KEYWORDS
Transportation, large businesses, Monte Carlo, logistics, management, complex systems, shipping, statistical optimization.

ABSTRACT

International businesses of considerable size can have enormous transportation problems. These difficulties can involve substantial costs. However, in our computer age these problems can also be viewed as opportunities to reduce basic costs by using computer simulation techniques such as multi stage Monte Carlo optimization (MSMCO) to organize the solution to and hopefully reduce shipping costs while delivering products to important customers. Presented here will be a little six variable two by three classic linear transportation problem, shipping a product in bulk from two factories to three customers. Then a larger one million variable nonlinear transportation problem involving the international shipping of many millions of units of a company's product will be presented and addressed with the MSMCO simulation technique. Then a review of the answer, the MSMCO technique, computer speeds of small and large computers, other transportation problems and the philosophy of statistical optimization using simulation techniques will be presented.

INTRODUCTION

There are many different and varied transportation problems that small and large businesses and companies face daily, monthly and yearly in our computer age. We focus here, for a start, on a classic linear transportation problem of shipping a product in quantity from two locations to three destinations. Then several solution approaches are mentioned for this example. Further reading on this classic type of problem is available in books such as (Barnett and Ziegler, 1994) and (Mizrahi and Sullivan, 1993). The pioneering work on linear programming theory was done by George Dantzig and others (Dantzig, 1963) over half a century ago (Dantzig, G. Ramser, J. 1959).

However, prominent researchers and economists such as (Tinbergen, 1978) have warned applied mathematicians and logistics departments of large businesses about the difficulties in linearizing essentially nonlinear practical problems just to use the easier, well developed linear simplex method (Dantzig, 1963). Therefore, presented here will be a large nonlinear transportation problem (as a second example) whose solution will be approximated with the MSMCO simulation technique. Comments on the fields of transportation, computer speeds and simulation will follow, after multiple approximate solutions are given.

A SMALL EXAMPLE

A company is shipping 3600 units of its product from its two factories to three customers.

Table 1: Summary of Problem

	Customer 1	Customer 2	Customer 3	
Factory 1	5 dollars	7 dollars	4 dollars	1200 suppliers
Factory 2	12 dollars	1 dollar	8 dollars	2400
	360	3000 demands	240	

The question is, given the six unit shipping costs for the six shipping routes and the factories' supplies and customers' demands, how much should be shipped from each factory to each customer to satisfy the demands and minimize the total shipping cost?

Most managers would have access to a linear programming simplex software package (on their computer) and could type in the problem and quickly get the solution of 2400 units shipped from factory 2 to customer 2, with 360,600 and 240 units respectively going to customers 1, 2 and 3 from factory 1. This yields a total cost of 9360 dollars.

Also, just shifting values around in the chart (Table 1) and noticing how attractive the route of factory 2 to customer 2 is (because of the one dollar unit cost) could lead to the same solution rather quickly. Another approach is to Minimize

$$C=5X_{11} + 7X_{12} + 4X_{13} + 12X_{21} + X_{22} + 8X_{23}$$

subject to all $X_{cj} \geq 0$) and

$$X_{11} + X_{12} + X_{13} \qquad = 1200$$
$$X_1 \quad X_2 \quad X_3$$
$$X_{21} + X_{22} + X_{23} \qquad = 2400$$
$$X_4 \quad X_5 \quad X_6$$
$$X_{11} \qquad X_{21} \qquad = 360$$
$$X_1 \qquad X_4$$
$$X_{12} \qquad X_{22} \qquad = 3000$$
$$X_2 \quad X_3 \qquad X_5$$
$$X_{13}$$
$$+ X_6 \qquad = 240$$
$$X_{22}$$

using multi stage Monte Carlo optimization (MSMCO) on the transformed problem we minimize

$$\begin{aligned} C= & \left| 5X_{11} + 7X_{12} + 4X_{13} + 12X_{21} + X_{22} + 8X_{23} - IXM \right| \\ & + \left| X_{11} + X_{22} + X_{13} - 1200 \right| \\ & + \left| X_{21} + X_{22} + X_{23} - 2400 \right| \\ & + \left| X_{11} + X_{21} - 360 \right| \\ & + \left| X_{12} + X_{22} - 3000 \right| \\ & + \left| X_{13} + X_{23} - 240 \right| \end{aligned}$$

subject to all $X_{ij} \geq 0$

The IXM total cost answer can be found by having an outer loop of IXM=500 to 15000 controlling these 1001 MSMCO tries. This ran quickly on a desktop computer. It also produced the 9360 dollar answer.

However, it should be mentioned that the real value of MSMCO is not using it on small linear problems. It can be useful for approximating a solution or finding the exact optimal as in (Conley, 1991a), (Conley, 1991b), and Conley, 1993) to difficult large optimization problems of many types.

The linear programming simplex technique developed in the mid-20th century is an "exterior algorithm," because it goes along the "edges" of the feasible solution space from corner point to corner point to find the optimal solution. This is because the fundamental theorem of linear programming is that the optimal solution is at a "corner point" of the feasible solution space.

However, the MSMCO simulation optimization technique is an "interior algorithm" because it searches the interior of the feasible space looking for answers. Many nonlinear problems have useful solutions in the interior of the feasible solution space. Therefore, the MSMCO simulation can try to locate these. Also, it can end up to be an exterior point (on an "edge" or "corner point") if that is where the answers are. So it is a general purpose solution technique that can involve much sampling of feasible solutions and closing in (with ever decreasing in size and repositioning n dimensional "rectangles" for a function of n variables) on the solution with many stages of the simulation.

Therefore, it can involve considerable computation in the computer program. Please see the section on current computer speeds for more information on this. Additionally, the section on the statistical optimization field (MSMCO) may be helpful.

A MILLION VARIABLE NONLINEAR SHIPPING STUDY

A large company has its product stored in quantity in its 64 warehouses in 23 countries ready for shipment to 15,625 stores that have placed orders for it. The warehouses have the product available in the 64 amounts of 124,995+5i units for i=1, 2, 3, . . . 64 warehouses, or a little over 8,000,000 units available during this hopefully temporary shortage of supply.

The 15,625 stores are requesting the amounts of 1,200+.01j units for j=65, 66, 67, ... 15690, or a little less than 20,000,000 units.

Therefore, there is an excess of demand (hopefully supply and demand will balance shortly). There are 64X15,625=1,000,000 shipping routes. The company would like to ship its product at minimum cost (of course) or a near approximation as a true optimal solution will be difficult and expensive to find given the one million shipping costs (one for each shipping route) are all nonlinear and depend on the amount shipped. The estimated cost equation is

$$C=\sum_{i=1}^{64} \sum_{j=1}^{15625} .00199*(i+j)*X(i,j)**(.65+.00002*(i+j))$$

Subsequently, the managers in the company's logistics department set the cost equation equal to 175,000/.0019 about 92.1 million Euro dollars in an attempt to drive the cost below the estimated 100 million Euros involved in using the old (last year's) method.

Then they use an MSMCO simulation optimization computer run to try to minimize the sum of the absolute values of the differences in the left and right hand sides of this cost equation and the 64 attendant warehouse shipping equations. Before running the program the absolute value from the cost equation is multiplied by .0019 in an attempt to reduce its magnitude to more in line with the other 64 equations for a smoother simulation. Note that if the warehouse equations have greatly different right hand side constraints they can be similarly balanced for a better MSMCO simulation also.

Forty-three stages involving ten thousand sample feasible solutions (in each stage) were looked at in the computer run in an ever narrowing and repositioning MSMCO "random" search for an approximate solution. The computer program ran for about 47 hours on a desk top PC computer, and produced the approximate solution which is summarized here with the equation error terms and the total cost in Euros of 92,105,263±111,579 or a savings of about eight million Euros over the old method of shipping in a similar situation.

The complete answer file (too big for this presentation here) is available by e-mail upon request. Please note that this is a hypothetical large nonlinear transportation problem. However, these types of problems are very real in the world of global economics and large international corporations. Additionally, such large companies would certainly have powerful computers which would probably run this simulation in 47 seconds or less. (Please see sections on computer speeds and MSMCO statistical optimization.) Even an outer loop could be added where successive runs tried to reduce the cost equation total cost while reasonably solving the 64 delivery equations.

COMMENTS ON THIS SOLUTION

The 64 equation errors in this first (43 stage MSMCO) solution attempt are in Table 2. The average equation error is about 3 units which may be acceptable. However, it should be noted that 62 of the equation errors average about 2 or 3 tenths of a unit error.

Equation 4 (warehouse 4) has an error of 34.21094 units it theoretically shipped but did not actually have (too high). Also, equation 51 (warehouse 51) has an error of 158.88281 units it had but did not ship (too low). As a practical matter, the dispatcher could quickly have the freight department send 34.21094 units from warehouse 51 to balance equation 4's undersupply (eliminating its error completely). This would leave 158.88281-34.21094=124.67187 excess units in warehouse 51 which could be shipped to waiting customers during this shortage of supply.

Specifically (looking at the printout of the one million Xij values (again available by e-mail upon request) an easy solution fix would be to ship all 124.67 excess units from warehouse 51 to customer number 15,625, who requests 1356.89 units but according to the complete answer printout was only shipped 549.130 units.

Specifically, the dispatcher could ship the A(51,1)=6.537, A(51,2)=5.492, A(51,3)=13.326, A(51,4)=6.905, A(51,5)=3.223, A(51,6) =13.027, A(51,7)=12.488, A(51,8)=6.910, A(51,9)=5.689, A(51,10)=8.760, A(51,11)=2.437, A(51,12)=8.608, A(51,13)=7.593,

Table 2: The 64 Warehouse Equation Errors

Equation Number	Error Term
1	.14063
2	.06938
3	.31250
4	34.21094
5	.14844
6	.46875
7	.02344
8	.05469
9	.21094
10	2.09375
11	.07813
12	.08594
13	.35938
14	.05469
15	.58594
16	.00000
17	.14844
18	.03125
19	.03906
20	.03906
21	.06250
22	.11719
23	.03125
24	.21875
25	.06250
26	.01563
27	.06250
28	.11719
29	.10156
30	.31250
31	.70313
32	.42188
33	1.04688
34	.36719
35	.99219
36	.10156
37	.00781
38	.12500
39	.15625
40	.14063
41	.29688
42	.36719
43	1.25781
44	.06256
45	.07031
46	.07813
47	.85156
48	1.47656
49	.04688
50	.00781
51	158.88281
52	.09375
53	.10156
54	.42696
55	.88281
56	.06250
57	.31250
58	.10938
59	.06256
60	.46875
61	.15625
62	.06250
63	.04688
64	.04688

A(51,14)=2.276, A(51,15)=13.089, A(51,16)=7.610 and A(51,17)=.700 leaving 9.807 units in A(51,17). This totals to about 124.67 units to be sent to the last customer (number 15,625). This would increase customer 15,625 units delivered to 549.130+124.67 still well below what they requested in this period of excess demand. This would require only the slightest percentage change in the overall about 92.1 million Euro cost equation.

ADDITIONAL SOLUTIONS

The MSMCO (or multi stage Monte Carlo optimization) simulation was rerun a few more times with different sequences of random numbers producing the 430,000 sets of feasible solutions (43 stages of 10,000 each). Additionally, the lengths of the one million dimensional "rectangles" of ever decreasing in size (and repositioning) as they follow the trail of better and better answers to our approximate solution in a goal programming setting further randomizes the solution helping to preserve the integrity of the "random" part of the MSMCO solution process.

Presented here are two more solutions (summarized by their equation errors) in Tables 3 and 4. We discuss the Table 3 solution first.

These MSMCO (43 stage) simulations took about 47 hours on a desk top PC. However, companies involved in big business on a scale that would ship a product in bulk from 64 warehouses to 15,625 stores or customers in a period of excess demand would surely have a more powerful, faster computer such as the ones produced and sold by International Business Machines (IBM, also known as Big Blue) (Krantz and Swartz, 2011). This program would run quickly (perhaps in a second or less) and allow the luxury of putting in an outer loop to say run 100 or 1000 of these simulations successively reducing the cost equation right hand side from in the 90 millions down into the 80 or 70 millions to see if there would be any much cheaper approximate solutions that had manageable 64 warehouse shipping equation errors. However, the only errors of any size here are 12.86719, 4.34375, 3.95313, 1.07813, 1.29688, 1.81250, 1.57813, 1.01563, 2.22656 and 1.70313. Perhaps the 12, 4, and 3 warehouses would shift those amounts and ignore the other small errors. Also, again note that if the warehouses have greatly differing amounts to ship their left and right hand sides could be easily balanced so that all equations look about the same size to the computer. This also makes for an easier, more accurate simulation.

Table 3: An Approximately 92.5 Million Euro Dollar Cost Solution

Warehouse Numbers	Warehouse Equation Errors	Units Available in Warehouse
1	12.86719	125000
2	0.00781	125005
3	4.34375	125010
4	0.15625	125015
5	0.21875	125020
6	0.05469	125025
7	0.14063	125030
8	3.95313	125035
9	0.14063	125040
10	0.04688	125045
11	0.39844	125050
12	0.99219	125055
13	0.42188	125060
14	0.05469	125065
15	0.03906	125070
16	0.09375	125075
17	0.08594	125080
18	1.07813	125085
19	0.27344	125090
20	1.67969	125095
21	0.37500	125100
22	0.16406	125105
23	0.01563	125110
24	0.01563	125115
25	0.10938	125120
26	1.29688	125125
27	0.04688	125130
28	0.03125	125135
29	0.03125	125140
30	0.27344	125145
31	0.27344	125150
32	0.00781	125155
33	0.24219	125160
34	0.17188	125165
35	0.20313	125170
36	0.87500	125175
37	0.09375	125180
38	0.02344	125185
39	0.00781	125190
40	0.81250	125195
41	0.26563	125200
42	0.49219	125205
43	0.92969	125210
44	0.08594	125215
45	1.81250	125220
46	1.57813	125225
47	0.21875	125230
48	0.02344	125235
49	0.02344	125240
50	1.01563	125245
51	0.00000	125250
52	2.22656	125255
53	1.70313	125260
53	0.39063	125265
55	0.17188	125270
56	0.17969	125275
57	0.87500	125280
58	0.00781	125285
59	0.10156	125290
60	0.08594	125295
61	0.57031	125300
62	0.21875	125305
63	0.16406	125310
64	0.01563	125315

Table 4 highlights a third solution with very manageable small errors. The average of the 64 warehouse equation errors in this third solution is .41005 of a unit.

Table 4: About a 92,000,000 Euro Cost Solution

Warehouse Number	Warehouse Equation Errors
1	0.2348
2	3.78125
3	0.06250
4	0.01563
5	0.13281
6	0.02344
7	0.10938
8	0.96094
9	2.49219
10	0.06250
11	0.29688
12	0.46094
13	0.30469
14	0.19531
15	0.02344
16	0.02344
17	1.45313
18	0.15625
19	0.38281
20	0.03125
21	0.07031
22	0.35156
23	0.19531
24	3.01563
25	1.52344
26	0.14063
27	1.14844
28	0.21875
29	0.11719
30	0.07813
31	0.23438
32	1.63281
33	0.67969
34	0.25000
35	0.21875
36	0.00000
37	0.07813
38	0.39063
39	0.17969
40	0.21875
41	0.15625
42	0.00781
43	0.21094
44	0.00781
45	0.19531
46	0.01563
47	0.29688
48	0.03125
49	0.77344
50	0.20313
51	0.14844
52	0.32813
53	0.29688
54	0.08594
55	0.82031
56	0.18750
57	0.57813
58	0.07813
59	0.03906
60	0.35938
61	0.22656
62	0.03125
63	0.00781
64	0.21875

CURRENT COMPUTER SPEEDS AND MSMCO

The million variable nonlinear problem worked on here certainly took a considerable amount of computer run time on the desk top PC available for this study. However, more powerful computers do exist and are becoming cheaper and ever more ubiquitous in our 21st century. Again, a large corporation that would be involved in shipping its product internationally in bulk from 64 factories (or warehouses) to 15,625 valued customers (or stores) would certainly have access to (or own) a large powerful computer facility that its managers and logistics staff would use on a daily basis.

The first thing they should do (in relation to this nonlinear transportation problem) is at least triple the sample size from 10,000 to 30,000 feasible solutions for each of the 43 ever narrowing MSMCO stages, and have an outer loop of 30 to 35 runs and reruns of this program. Then they should take that best answer and implement it. Additionally, in that outer loop (of 30 to 35 runs) the right hand side value in the cost equation (the total shipping cost) could be lowered on each of those 30 to 35 trips through the outer loop to also try to lower the shipping cost (as long as the shipping equation errors remain manageably small).

The run time on a mainframe computer for such a massive simulation would be tolerably short. An illustration of a very powerful computer would be the ones used by the National Oceanographic and Atmospheric Administration (NOAA) of the United States to forecast the world's weather. The International Business Machines Corporation (called IBM or Big Blue) makes powerful computers that do many trillions of calculations a second (Krantz and Swartz, 2011) and (Dongarra, 2011). It would certainly "solve" our 65 equation 15,625 less than or equal constraint system satisfactorily in a few seconds of computer time, or a fraction of a second.

The NOAA, of course, needs large fast computers to simulate the solution to the mathematical models (involving enormous equations) of the world's weather for forecasting purposes.

All simulation professionals can benefit (and their customers) from increased speeds and capacity and reduced prices and sizes of powerful new computers.

SUPER COMPUTER SPEEDS TODAY

Jack Dongarra, professor of computer science and electrical engineering at the University of Tennessee, Knoxville, compiles a list of the 500 most powerful computers in the world (Dongarra, 2011). A look at the top 100 of the list reveals these computers in 21 countries on 5 different continents. The current fastest one is the

Japanese "K" computer (Kopytoff, 2011) and (Dongarra, 2011) which does 8.3×10^{15} calculations per second. This computer would probably run the 47 hour PC simulation used here on the million variable nonlinear transportation problem in a tiny fraction of a second. The other 499 computers on Dongarra's list would possibly all run this MSMCO simulation in a fraction of a second also. The old days of slow mainframe computers, the size of a small city block (filled with vacuum tubes that have to be constantly replaced), are over. The whole field of simulation can only benefit from these advances.

None of this presentation means that we can just abandon reasoning and overwhelm our technical problems with larger computers. Much reasoning is needed to ask the right questions (however complex they may be in the computer age) and set up useful simulation programs to solve them.

A computer can produce the wrong answer a million times faster, but it can also produce a right answer or a useful approximate solution to the right nonlinear question if the work is done carefully and is technically correct.

STATISTICAL OPTIMIZATION AND CLASSICAL STATISTICS

The field of classical statistics can be and is used to assist in complex decision making in virtually all areas of science, business and industry. It is a well-established and respected subject whose key is that every statistic of interest has a sampling distribution which is determined by the null hypothesis (the point of contention) and the actual statistic itself (usually a formula) that has been proven to have a certain distribution (normal, t, chi square, F and gamma are common ones, etc.) under repeated sampling if the null hypothesis is true.

A typical example would be to use the statistic $Z = (\bar{x}-u)/(S/\sqrt{n})$ for large n which is known to be standard normally distributed if the null hypothesis of u=? (the point of contention or guessed value) is correct. Therefore, if a company guessed that its hourly production was u=500 units, and was willing to take a 5 percent chance of being wrong, then the critical values in its bell shaped curve sampling distribution would be ±1.96. Therefore, 95 percent of the area of this symmetric bell shaped distribution would be between ±1.96. So if a sample of say 120 hours of production yielded a value of $Z = (\bar{x}-500)/(S/\sqrt{120}) = -2.15$ when the 120 sample values are inserted into this formula, one could reject the null hypothesis (with confidence) that the true mean u=500. Therefore, it would appear that the mean has dropped because the calculated value has traveled across the sampling distribution to its lower area. (An equivalent confidence interval could also be presented as desired.)

However, the new and emerging field of statistical optimization (also called multi stage Monte Carlo optimization and other names too) views every system of equations, and all minimization or maximization problems as having a sampling distribution made up of the arrangement or pattern of all the millions (or billions or trillions, etc.) of function values of the formula to be optimized under all feasible (possible) solutions.

The computer statistical simulation techniques such as multi stage Monte Carlo optimization (MSMCO) are used to create n dimensional rectangles of ever decreasing in size and shape that rocket across the feasible solution space to the minimum area (or exact minimum) left hand side of the distribution of the feasible solutions for a minimization problem (or solving a system of equations). Alternatively, by changing the less than sign to a greater than sign in the computer simulation program, the MSMCO driven ever decreasing in size and repositioning n dimensional rectangles (for a function of n variables) will rocket across the sampling distribution of the feasible solution space to the extreme right hand side to maximize a profit or revenue or output equation for example.

This new field of "statistical optimization" also attempts to cross the sampling distribution of the statistic in question just like the classical statistics testing of hypothesis does. However, their purposes are different. Classical statistics attempts to land in the extreme upper or lower end of the statistics sampling distribution in order to reject the null hypothesis, with a high level of confidence. Whereas, the new subject of "statistical optimization" attempts to cross the sampling distribution of the optimization problem at hand to the extreme left hand side to find its minimum (or useful approximation). Alternately, it will, as directed, cross the distribution to its extreme right hand side in pursuit of its maximum value (profit, revenue, etc.) as required.

The key to both of the exciting and useful fields of classical statistics and the newer "statistical optimization" is the sampling distribution of the problem of interest. The goal in both fields is to try to cross that sampling distribution for the purposes of decision making (in classical statistics) or optimization and solving systems of equations. Again the multi stage Monte Carlo optimization (MSMCO) technique was used here and elsewhere in the references to try to cross the feasible solution space for purposes of optimization or solving a system of equations.

Las Vegas optimization, simulated annealing, genetic algorithms, multi stage Monte Carlo optimization, etc. (by whatever name) are all trying to cross the sampling distribution in some kind of "random" yet ever more "focused" search to find and use the optimal solution region. The driving force behind these journeys through n dimensional space (for a function of n variable) is small powerful, inexpensive and ever more ubiquitous PC computers in our 21st century. Main frame computers work too if one's company has that capacity

Recent press announcements of the IBM's (the International Business Machines Corporation) new computers and Japan's and China's machines (Dongarra, 2011) have included estimates of a mainframe computers that will do in the area of $1x10^{15}$ calculations per second. Currently Japan's Kei computer is $8.3x10^{15}$ per second (Kopytoff, 2011). A computer of that level of speed would reduce the 47 hour desk top MSMCO simulation runs presented here to about one one-hundredth (or .01) of a second each, and set off the possibility of a more thorough MSMCO simulation run of a few minutes that would get very close to the extreme left hand side (or exact optimal) of the sampling distribution of our 65 equation one million variable transformed optimization problem.

Big Blue's (IBM) Watson and other computers (Krantz and Swartz, 2011) challenged and defeated some of the world's finest brains and champion chess players, perhaps for some great publicity. They certainly can also be of great assistance to the world's best managers, logistics and transportation experts in dealing with large scale shipping (and other) problems in business and industry.

APPROACHES TO OTHER TRANSPORTATION PROBLEMS

This presentation is featuring the classic linear transportation problem and its nonlinear generalization, where there are discounts for shipping in quantity from N locations to M destinations. However, there are of course other approaches to transportation problems and other problems that can be viewed as transportation problems. A quick example would be to do these two well-known types of problems and to maximize a revenue stream or profit equation. Also, in any of these four settings, special deals (or minimum content delivery guarantees) could be offered to one's best customers or to a new country or market that a company wants to establish a presence in.

Additionally, a manager or logistics specialist might wish to replace air freight deliveries with barge, railroad, truck or large container carrying ships or some combination of those five types for delivering one's product in bulk. Also, should a company purchase a trucking firm to have more direct control over its shipping costs? Pipelines or even electronic delivery of products (in very special cases) could occasionally be an option.

Companies could consider redesigning a product so it is cheaper to ship (more could fit in a standard size railroad boxcar) (Conley, 1981). However, redesigning a product to save shipping costs must be done with care to see that the customers like the redesign and it's still as good or better of a product.

Companies can consider moving closer to its customers or sources of raw materials (or to a more favorable tax or labor jurisdiction) to reduce shipping costs. If deliveries to many customers are made out of any one truck or car traveling on the shortest route (Krolak, Felts and Marble, 1970), (Lawler, Lenstra, Rinnooy-Kan, and Shmoys, 1984), (Conley, 1991) and (Conley, 1996) could save time, fuel and overall delivery costs. It can even be modified for multivariate forecasting (Conley, 2007). A good review of classic forecasting (Hayter, 2002) could precede this new forecasting and correlation approach. However, the classic linear transportation problem and its nonlinear generalization featured here remain important and useful. Multi Stage Monte Carlo can also be useful on a wide variety of operations research problems (Clark, 1984).

CONCLUSION

Presented here was a brief review of the classic linear transportation problem of delivering a product in quantity from N locations to M destinations (N times M shipping routes) at minimum cost. A small example accompanied this discussion. Then a large nonlinear generalization of that type of transportation problem (with 64 times 15,625 or 1,000,000 shipping routes) was presented and several approximate solutions were produced in a goal programming setting using a small desk top PC. This was followed by a review of some of the larger mainframe computers available today (Dongarra, 2011), (Kopytoff, 2011) and (Krantz and Swartz, 2011) which could solve the one million variable nonlinear transportation problem very quickly using multi stage Monte Carlo optimization (MSMCO) (Conley, 2010) which was used here with a small desk top PC computer. The new field of statistical optimization was also reviewed.

The shortest route (for delivery trucks, etc.) theory (sometimes called TSPs, an acronym for traveling salesman problem) also can be modified for K dimensions to use the CTSP to find multivariate nonlinear relationships between variables. Powerful new computers are making transportation theory and practice useful in many and varied fields in our computer simulation age.

Again, the featured second example presented could take advantage of today's powerful computers and the MSMCO simulation's power and versatility. It involved shipping on a million routes with nonlinear discounts for shipping in quantity, from 64 locations to 15,625 customers in a period of excess demand (shipping a little over 8 million units when the customer demand was more than 19 million units).

Multiple approximate solutions were produced and presented after using a desk top PC to run this minimization problem involving 65 equations (including the cost equation) with 15,625 major less than or equal constraints.

This was followed by a discussion of today's large and powerful computers that could find a useful approximate solution more quickly than a PC. Also, other types of transportation problems were reviewed (featuring the MSMCO solution technique's versatility). Additionally, a section on the new area of statistical optimization was mentioned.

Again, thinking over Kopytoff's article (Kopytoff, 2011) about the Japanese "K" computer (and other fast ones too) that can do $8.3x10^{15}$ calculations a second, it probably means that the current "K" computer can do more calculations in a second or two than all of the calculations done by hand or with small calculation aids by the entire human race in the history of the world. The field of simulation in particular and humanity in general can greatly benefit from these computer advances.

REFERENCES

Barnett, R. A. and Ziegler, M. R., 1994. Applied Mathematics for Business, Economics, Life Sciences and Social Sciences. 5th Edition. MacMillan, New York.

Clark, K., 1984. "Operations Research and Policy Problems: Simplified Computer Optimization Techniques for Policy Analysis." Policy and Information, Vol. 8, No. 1, pp. 63-67.

Conley, W. 2011 "The Other Global Warming Problem." Proceedings of 2011 Industrial Simulation Conference, Venice, Italy. ETI EUROSIS, Ghent, Belgium, pp. 257-261.

Conley, W. 2011. "A three Thousand Variable Nonlinear Transportation Problem with Excess Demand." Proceedings of 2011 Industrial Simulation Conference, ISC 2011, Venice Italy, ETI EUROSIS, Ghent, Belgium, pp. 177-179.

Conley, W. 2007. "Simulation, Optimization and Correlation with Multi Stage Monte Carlo Optimization." *International Journal of Systems Science*, Vol. 38., No. 12, pp. 1013-1020.

Conley, W. 2006. "Discovering Relationships that are Not Functions." In Proceedings of 4th International Industrial Simulation Conference, ISC 2006 (Palermo, Italy, June 5-7), EUROSIS-ETI, Ghent, Belgium, 163-167.

Conley, W. 1991a. "Programming an Automated Punch or Drill." *International Journal of Systems Science*, Vol. 22, No. 11, pp. 2039-2056.

Conley, W. 1991b. "Multi Stage Monte Carlo Optimization Applied to Selected Management Problems. In Proceedings of the 1991 Summer Computer Simulation Conference (SCS), Baltimore, August 1991 SCS, San Diego, pp. 1010-1014.

Conley, W. 1989. "A Shipping Route Minimization Technique with Thirty and One Hundred Point Examples." *Journal of the Japan Society for Simulation Technology*, Vol. 8, No. 3, Tokyo, pp. 162-166.

Conley, W. 1981. Optimization: A simplified Approach. Petrocelli Books, Princeton and New York.

Dantzig, G. 1963. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey.

Dantzig, G., Ramser, J. 1959. "The Truck Dispatch Problem." Management Science, Vol. 6, pp. 80-91.

Dongarra, J. June 25, 2011. Personal communication.

Hayter, A. 2002. *Probability and Statistics for Engineers and Scientists*. 2nd Edition. Duxbury, Pacific Grove, California.

Kopytoff, V. C. 2011. "Japanese 'K' Computer is Ranked Most Powerful." New York Times, New York. June 20, 2011, p. B6.

Krantz, M. and Swartz, J. 2011. "IBM Shows Secret to Corporate Longevity: Big Blue Survives 100 Years by Always Adapting." Arlington, Virginia, U.S.A. USA Today, June 16, 2011, p. B1.

Krolak, P., Felts, W., Marble, G. 1970. "A Man Machine Approach Toward Solving the Traveling Salesman Problem." Design Automation Workshop. San Francisco, California.

Lawler, E., Lenstra, J., and Rinnooy-Kan, and Shmoys, D. 1984. *The Traveling Salesman Problem*. John Wiley & Sons, New York.

Mizrahi, A. and Sullivan, M. 1993. *Mathematics for Business Sciences and Social Sciences*. John Wiley & Sons, New York.

Tinbergen, J. 1978. "Effects of the Computerization of Research." Compstat, 1978. Physical-Verlag, Wurzburg & Vienna, pp. 10-20.

AUTHOR BIOGRAPHY

WILLIAM CONLEY received a B.A. in mathematics (with honors) from Albion College in 1970, an M.A. in mathematics from Western Michigan University in 1971, an M.Sc. in statistics in 1973 and a Ph.D. in mathematics - computer statistics from the University of Windsor in 1976. He has taught mathematics, statistics, and computer programming in universities for over 30 years. He is currently a professor emeritus of Business Administration and Statistics at the University of Wisconsin at Green Bay. The developer of multi stage Monte Carlo optimization and the CTSP multivariate correlation statistics, he is the author of five books and more than 200 publications world wide. He is a member of the American Chemical Society, a fellow in the Institution of Electronic and Telecommunication Engineers and a senior member of the Society for Computer Simulation.

DESIGN OF A HYBRID NANOSCOPIC/MICROSCOPIC ROAD TRAFFIC SIMULATION

Tomas Potuzak
Department of Computer Science and Engineering
University of West Bohemia
306 14, Plzen,
Czech Republic
E-mail: tpotuzak@kiv.zcu.cz

KEYWORDS
Hybrid traffic simulation, nanoscopic simulation, microscopic simulation.

ABSTRACT

The computer simulation of road traffic is an important tool for analysis and control of road traffic networks. Its evolution began with simple modeling of traffic flows without. Today, four types of road traffic simulations depending on the level of detail are commonly used – the macroscopic, mesoscopic, microscopic, and nanoscopic simulations. Since the detailed simulations are more time-demanding, it is also possible to utilize a hybrid simulation, which represents various parts of the simulated area at different level of detail. In this paper, we present the design of a hybrid simulator for urban traffic, which uses a nanoscopic simulation for the crossroads and a microscopic simulation for the lanes between the crossroads.

INTRODUCTION

The computer simulation of road traffic is an important tool for analysis and control of actual or designed road traffic networks. It is a subject to research for more than a half of the century. The evolution of the road traffic simulation began with simple modeling of traffic flows in particular traffic lanes in 1950s (Lighill and Whitman 1955). In 1980s, simulations considering particular vehicles and their interactions began to emerge (Gipps 1981). Today, it is even possible to simulate individual behavior of drivers (Ni 2003).

However, such a great detail is very computation-intensive, even for contemporary computers, especially for large traffic networks. A way, how to speed up the simulation, is to use a hybrid approach where various parts of the traffic network are simulated at different level of detail. In this paper, we present the design of such a hybrid simulator of urban traffic.

ROAD TRAFFIC SIMULATION

Before we proceed with the description of the design of our simulator, we will focus on the road traffic simulation.

Time-flow Mechanism

One of the most important issues of each simulation is the way the simulation time is advanced. In the field of road traffic simulation, there are two commonly used time-flow mechanisms – an *event-driven* and a *time-stepped* approach.

Using the event-driven time-flow mechanism, the simulation time is divided into a sequence of time-stamped events. With each event an action is associated. The action is an incremental change to the simulation state, which is performed at the time specified by the time stamp. So, the simulation time is advanced irregularly from one time stamp to another (Fujimoto 2000).

Using the time-stepped time-flow mechanism, the simulation time is divided into sequence of equally-sized time steps. In each time step, the entire simulation state is recomputed. So, the simulation time is advanced regularly from one time step to another (Fujimoto 2000).

Level of Detail

According to the level of detail, there are four types of road traffic simulations.

The *macroscopic* simulation deals only with aggregate traffic flows in particular traffic lanes described by a set of parameters (e.g. mean speed, vehicle density, etc.) Macroscopic simulations are the oldest ones (Lighill and Whitman 1955) and can be very fast. Both time-flow mechanisms are commonly used.

The *mesoscopic* simulation usually considers particular vehicles in some form, but represents their interaction at very low detail (Burghout 2004). There are several approaches, which are considered mesoscopic, such as queuing networks (Nizzard 2002). Again, both time-flow mechanisms are commonly used. The mesoscopic simulations are generally slower than the macroscopic simulations.

The *microscopic* simulation models every single vehicle as an object with its own position, direction, speed, and acceleration. The position of the vehicles is periodically recomputed based on their current speed and the traffic conditions in their vicinity. The most common approaches are the cellular automaton model (Nagel and Schreckenberg 1992) and the car-following model (Gipps 1981). Both models define the set of rules for movement of the vehicles, but using different means. Most often, the time-stepped time-flow mechanism is utilized. The microscopic simulation is very time-consuming, because of the high detail.

Even more detailed (and more computation-intensive) is the *nanoscopic* simulation. It can incorporate individual behavior of the drivers, acceleration power and other features of the individual vehicles (Vanderschuren 2007; Ni 2006) and traffic, which are often neglected in less-detailed simulations.

Combining of Simulations with Different Level of Detail

The combination of simulations with different level of detail is a way how to speed up a computation-intensive road traffic simulation. There are several existing designs of such hybrid simulations (Hartman and Herout 2007; Ni 2003; Ni 2006; Huang et al. 2008). The utilized simulations are usually one level of detail apart (e.g. a macroscopic simulation is combined with a mesoscopic simulation), but it is not a rule (Hartman and Herout 2007).

One of the important aspects of the hybrid traffic simulation is to select, which areas will be simulated at higher and which at lower level of detail. It is possible to base the selection on preferences. This means that areas, which are more important for the given case, will be more detailed. Example of such approach can be found for example in (Hartman and Herout 2007). Another possibility is a systematic approach. It means that the areas, which incorporate complex interactions of the vehicles and other traffic elements (e.g. traffic crossroad), are simulated at higher level of detail and simpler areas (e.g. traffic lanes between the crossroads or freeways) are simulated at lower level of detail.

Conversion of Traffic Flows between Simulations

Regardless the selected approach, at the edges of the parts simulated at different level of detail, it is necessary to convert one simulation to another (Hartman and Herout 2007; Huang et al. 2008). The actual implementation of such conversion highly depends on both simulations. An extreme example can be found in (Hartman and Herout 2007), where the conversion from a microscopic to a macroscopic simulation and vice versa is performed. For this purpose, two special objects are employed – the *micro-macro* and *macro-micro* converters. These objects perform the conversion from the particular vehicles to traffic flows and vice versa, but not without a loss of information (Hartman and Herout 2007).

DESIGNED HYBRID SIMULATOR

Now, we can proceed with the description of the design of our hybrid nanoscopic/microscopic road traffic simulator.

Combining of Nanoscopic and Microscopic Simulation

Using the systematic approach (see above), the traffic crossroads will be simulated using the nanoscopic simulation and the traffic lanes between the crossroad will be simulated using the microscopic simulation.

The reason for usage of the nanoscopic simulation in the crossroads is that there are many complex interactions, which are neglected by the microscopic simulation. Some examples are the influence of the pedestrians, influence of the inclination of the traffic arms, different reaction time of the particular drivers on green signal, and so on. Moreover, the vehicles are often moving very slowly in the crossroad. Hence, it is convenient to use a short time step to recalculate their positions. Time step length of one second commonly used in microscopic simulations is inadequate for fine modeling of the slow-moving vehicles. Of course, usage of a shorter time step and detailed modeling of more aspects of the traffic lead to steep increase of the number of performed computations, which means the slow down of the simulation.

On the other hand, in the traffic lanes, the vehicles are moving relatively fast (e.g. 50 km/h), and their interaction is not so complex. There are usually no pedestrians, all vehicles are moving at similar speeds, and so on. Hence, for the traffic lanes, the microscopic simulation with one second long time step is more than sufficient. The utilization of the nanoscopic simulation in the traffic lanes is unnecessary and would lead to further slow down of the simulation.

Utilized Microscopic Simulation

The hybrid simulator is based on the Distributed Urban Traffic Simulator (DUTS) developed at Department of Computer Science and Engineering of University of West Bohemia (DSCE UWB). It is a discrete time-stepped microscopic simulator of urban traffic, which incorporates three traffic models inspired by three different simulators – JUTS (Hartman and Herout 2007), TRANSIMS (Nagel and Rickert 2001), and AIMSUN (Barceló and Casas 2002). All these models can and will be used in the hybrid simulator.

The JUTS-based model is inspired by the JUTS system also developed at DSCE UWB. It is based on a modified Nagel-Schreckenberg's cellular automaton model (Nagel and Schreckenberg 1992; Hartman and Herout 2007) where the crossroads and traffic lanes are divided into 2.5 meters long traffic cells. Vehicles can have various lengths and hence occupy from one to six cells. The TRANSIMS-based model is inspired by the TRANSIMS simulator developed at Los Alamos National Laboratory (Nagel and Rickert 2001). Similar to the JUTS system, it is based on the Nagel-Schreckenberg's cellular automaton (Nagel and Schreckenberg. 1992), but the cells are 7.5 meters long and the vehicles have only one length correcponding to one cell. The AIMSUN-based model is inspired by the AIMSUN simulator (Barceló and Casas 2002). Unlike the previous two models, it is based on a modified Gipps' car-following model (Gipps 1981), where the vehicles can be located at any position of the traffic lane. Different lengths of the vehicles are considered. All three models utilize one second long time step. For more information, see (Potuzak 2009).

Utilized Nanoscopic Simulation

The nanoscopic simulation is now under development. It is loosely based on the Gipps car-following model (Gipps 1981). It is focused only on the traffic crossroad. The crossroad consists of several traffic arms divided into incoming and outgoing traffic lanes. In the center of the crossroad, there are traffic paths connecting particular incoming and outgoing traffic lanes. The vehicles use these paths for movement through the crossroad (see Fig. 1).

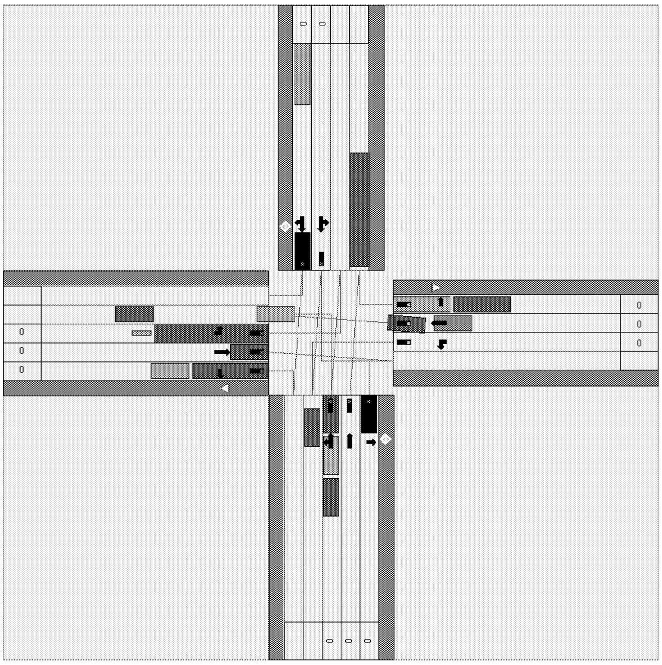


Figure 1: Visualization of the Nanoscopic Simulation

The vehicles have various lengths, various widths and it is also possible to set further parameters, such as maximal acceleration, reaction time of the drivers, and so on. The vehicles can be located at any position of the traffic lane or the traffic path. The vehicles respect the traffic lights and all other traffic regulations. For example, a vehicle will not enter the crossroad if it is unable to continue in driving behind it. Nevertheless, this "perfect" behavior can be changed in order to simulate the reality of the road traffic more accurately. Modeling of the various behaviors of the particular drivers and of the conflicts with pedestrians and other vehicles are planned as well. However, this part of the designed nanoscopic simulation is in very early stage of development.

The length of the time step of the nanoscopic simulation is adjustable and preset to 0.1 seconds. This means that for one step of the microscopic simulation, ten steps of the nanoscopic simulation will be performed. The value of 0.1 seconds should be sufficient for most traffic situations. It is possible to set even shorter time step, but it should be taken into account that the shorter the simulation step is, the slower is the simulation.

Conversion of Nanoscopic and Microscopic Simulation

The conversion of the nanoscopic to microscopic simulation and vice versa is simpler than that of microscopic and macroscopic ones described in (Hartman and Herout 2007).

Both simulations consider particular vehicles. These vehicles can incorporate attributes necessary for both nanoscopic and microscopic simulations for whole time the vehicle is moving in the simulated area. The nanoscopic simulation can then use the attributes, which it requires, and similar the microscopic simulation. When the vehicle crosses the edge of the parts with different simulation, the attributes of the target simulation are updated using the attributes of the source simulation.

CONCLUSIONS

In this paper, we presented the design of a hybrid nanoscopic/microscopic simulator of road traffic, which is currently under development. The simulator utilizes the nanoscopic simulation for modeling of the complex interactions in the traffic crossroads in order to achieve precise modeling of the traffic. The microscopic simulation is used for modeling of the less complex movement of the vehicles in the traffic lanes between the crossroads in order to speedup of the simulation.

The finishing of the simulator implementation is the main aim of our future work. Since the microscopic simulation is complete, we will focus on the nanoscopic simulation and then on the integration of both simulations.

REFERENCES

Barceló J. and J. Casas. 2002. "Dynamic network simulation with AIMSUN", In *Proceedings of the International Symposium on Transport Simulation*, Yokohama.

Braban-Ledoux C. 2000. *METACOR – A Macroscopic Modelling Tool for Corridor Application to the Stockholm Test Site*. Final Report CRT2000:05, Center for Traffic Engineering & Traffic Simulation, Kungl Tekniska Högskolan, Valhallavägen.

Burghout W. 2004. *Hybrid microscopic-mesoscopic traffic simulation*. Doctoral thesis, Royal Institute of Technology, Stockholm.

Fujimoto R.M. 2000. *Parallel and Distributed Simulation Systems*. John Wiley & Sons, New York.

Gipps P.G. 1981. "A behavioural car following model for computer simulation". In *Transp. Res. Board*, 15-B(2), 403-414.

Hartman D. and P. Herout. 2007. "Construction of a Hybrid Traffic Model Based on JUTS Cellular Model". In *Proceedings of the 6th EUROSIM Congress on Modelling and Simulation*, Ljubljana.

Huang W., J. Sun, and K. Li. 2008. "A Multiple Representation Entity-Based Approach to Hybrid Traffic Simulation Model". In *2008 International Conference on Information Management, Innovation Management and Industrial Engineering*, Taipei, 7-12.

Lighill M.H. and G.B. Whitman. 1955. "On kinematic waves II: A theory of traffic flow on long crowed roads". In *Proceedings of the Royal Society of London*, s. A, 229, London, 317-345.

Nagel K. and M. Rickert. 2001. "Parallel Implementation of the TRANSIMS Micro-Simulation". In *Parallel Computing*, 27 (12), 1611-1639.

Nagel K. and M. Schreckenberg. 1992. "A Cellular Automaton Model for Freeway Traffic". In *Journal de Physique I*, 2, 2221-2229.

Ni D. 2003. "2DSIM: A Prototype of Nanoscopic Traffic Simulation". In *Proceedings of Intelligent Vehicles Symposium*, Columbus, 47-52.

Ni D. 2006, "A Framework for New Generation Transportation Simulation". In *Proceedings of the 2006 Winter Simulation Conference*, Monterey, 1508-1514.

Nizzard L. 2002. *Combining Microscopic and Mesoscopic Traffic Simulators*. Raport de stage d'option scientifique, Ecole Polytechnique, Paris.

Vanderschuren M. 2007. "Calibrating Microscopic Simulation Models". In *Proceedings of the 26th Southern African Transport Conference*, Pretoria, 610-620.

Potuzak T. 2009. *Methods for Reduction of Interprocess Communication in Distributed Simulation of Road Traffic*. Doctoral thesis, University of West Bohemia, Pilsen.

A Multi Start Iterated Local Search algorithm for the multi compartment vehicle routing problem

Cadet David JOSEPH[1]
Christian PRINS[2]
1: cadet_david.joseph@utt.fr
2: christian.prins@utt.fr

Lionel AMODEO[3]
Farouk YALAOUI[4]
3: lionel.amodeo@utt.fr
4: farouk.yalaoui@utt.fr

Institut Charles Delaunay, University of Technology of Troyes (UTT)
BP 2060, 10010 Troyes Cedex, France

KEYWORDS

Transportation, Multi Compartment Vehicle routing problem, Multi Start Iterated Local Search

ABSTRACT

Multi Compartment Vehicle Routing Problem (MC-VRP) which is a generalised version of VRP is studied in this paper. MC-VRP has the particularity to allow transportation of inhomogeneous products on the same vehicle and is widely encountered in the food and petrol industries. Besides capacitated VRP characteristics, customers may order incompatible product families which must be transported separately into compartmented vehicles. An efficient optimization based on a Multi Start Iterated Local Search (MS-ILS) metaheuristic is proposed for this problem. Computed results are shown to be more competitive compared to the best known solutions from previous investigations in the literature.

INTRODUCTION

The Multi-Compartments Vehicle Routing Problem (MC-VRP) is a generalization of VRP. This problem covers applications of real life such as petroleum, food and retail industries. As an extension of VRP, it is also NP-hard. MC-VRP allows transporting heterogeneous or incompatible products in the same vehicle. In food industry distribution, compartments are sometimes dedicated to specific products types.

Several variants were studied in research about MC-VRP. Most of the papers focused on petroleum distribution industries. Van der Bruggen *et al.* (1995) solve a large oil distribution problem involving, at an operational level, the resolution of routing problems for gas trucks with compartments. Their resolution methodology consists in three steps: (1) Customers assignment to depots, (2) Computation of delivery frequencies (3) Routes construction. They propose a heuristic which inserts customers with large demands and located far from depot in priority in routes construction.

Avella *et al.* (2004) solved a periodic routing problem. A specific loading rule is applied to each tank which must be completely full or empty. They proposed a greedy-like heuristic and a set partitioning-based exact method for this problem. Periodic routing problem was also investigated by Cornillier *et al.* (2008) in a context of multi-period petrol station replenishment problem. Their goal consists of determining periodically the set of stations to deliver, the allocation of loads to compartments and vehicles routes. They developed an iterative heuristic which embeds a route packing procedure. In their study, contrary to Avella *et al.* (2004), visit dates and delivery amounts are not *a-priori* specified.

Some recent works investigated in the field of food and groceries distribution. Chajakis and Guignard (2003) studied a distribution optimization problem dealing with convenience stores supplies. The objective is to find an assignment of customers (stores) to vehicles which satisfy all demands and keep the sum of delivery and cooling costs minimal. Considering two layouts of compartments structures, they proposed two integer programming models. Approximations were made based on Lagrangean Relaxation.

El Fallahi *et al.* (2008) resolved a MC-VRP where each compartment is dedicated to one product in a food industry distribution application. In their study, they considered some specific compatibility rules between products and compartments for cross-contamination prevention reasons. Each compartment is dedicated to only one type or family of products. They developed two algorithms to tackle this specific problem: a tabu search (TS) and a memetic algorithm (MA).

Mendoza *et al.* (2010) studied a particular case of MC-VRP where customers' demands are stochastic. The compatibility rule of one compartment / one product is taken into consideration as in El Fallahi *et al.* (2008). They developed a memetic algorithm based approach which proceeds in two phases. During the first phase, the algorithm applies a spare capacity strategy to solve the deterministic counterpart of the problem (MC-VRP) by reserving a fraction of the capacity on each compartment to respond to high demand realizations. The second part incorporates fluctuations in demands.

Derigs *et al.* (2010) presented a global approach to solve MC-VRP. They took into account several specificities like the number of compartments, the compartments' capacities and the incompatibilities between products and between products and compartments. They developed a solver suite of heuristic components which covers a broad range of alternative approaches for construction, local search, large neighborhood search and meta-heuristics.

Our study is motivated by a specific VRP problem met in cattle food distribution. An application is found in a cattle food industry located in the North-East region of France, their distribution policies require products separation into vehicle compartments. In order to avoid cross-contamination, products mix is prohibited. Sometimes, a rule of products allocation to specific compartments is adopted. Especially, El Fallahi *et al.* (2008) investigated a problem found in food industry context. In their study, they took into account cross contamination and products allocation to compartments rules. The specific problem that they addressed is quite unexplored in the literature. In this paper, we will focus our research on developing an optimization method to tackle this problem. We will take into consideration the specific product allocation rule where, for each product, one compartment is a-priori assigned for packing. In the remainder, the paper will be structured as follows: problem formulation, our method description, computational results and conclusions and perspectives.

PROBLEM FORMULATION

MC-VRP is defined on a complete undirected network with a set of nodes $N = \{0, 1,\dots, n\}$ including one depot (node 0) and a set N' of n customers. Cost of travelling from location (customer) i to j is assumed to be symmetric ($c_{ij} = c_{ji}$). The set of costs is assumed to satisfy the triangle inequality. A set $P = \{1, 2,\dots, p\}$ of p products are delivered from the depot by a fleet $V = \{1, 2, \dots, v\}$ of identical vehicles partitioned into p compartments. A compartment p which is assigned to product p has a capacity Q_p while each vehicle has a total capacity $Q = \sum_{t\in T} Q_p$. Each customer orders at least one product. The quantity of an ordered product p by a customer i is denoted by q_{ip} such that $0 < D_{ip} \le Q_p$.

<u>*Specific assumptions*</u>

- Any client global demand quantity fits vehicle capacity
- Split deliveries to clients are not allowed
- Fleet size is big enough to satisfy all deliveries
- Load into a compartment concerns a full or a partial customer product order

<u>*Decision variables*</u>

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ visits arc } (i,j) \\ 0 & \text{else} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{if vehicle } k \text{ visits node or client } i \\ 0 & \text{else} \end{cases}$$

<u>*Linear model*</u>

$$\min \sum_{i,j\in N}\sum_{k\in V} (c_{ij} \times x_{ijk}) \qquad (1)$$

Subject to

$$\sum_{k\in V} y_{ik} = 1 \quad \forall i \in N' \qquad (2)$$

$$\sum_{k\in V} y_{0k} \le v \qquad (3)$$

$$\sum_{j\in N} x_{ijk} = \sum_{j\in N} x_{jik} = y_{ik} \quad \forall i \in N, \forall k \in V \qquad (4)$$

$$\sum_{i,j\in S} x_{ijk} \le |S| - 1 \quad \forall k \in V, \forall S \in N', S \ge 2 \qquad (5)$$

$$\sum_{i\in N} y_{ik} \times D_{ip} \le Q_p \quad \forall k \in V, \forall p \in P \qquad (6)$$

Constraints from (2) to (5) are general VRP constraints. They specify that clients have to be visited once (2), impose respect of fleet capacity (3). The first part of equations (4) ensures routes connectivity: any vehicle k that enters node i must leave it. The second part of (4) forces each node to be entered and left the same number of times as the number of vehicles that visit it. Equations (5) are classical subtour elimination constraints which oblige all tours to start and end at the depot. Constraints (6) deal with products dispatch into compartments. It is imposed that their cumulative ordered quantity doesn't violate corresponding compartment size limit.

This model provides a formal description of the problem that we are studying. As MC-VRP is a generalization of VRP, it is obvious that its computational complexity is harder than VRP. For this reason, our exact resolution experiments based on this model were limited to small instances. In order to solve big size problems encountered in industries, we develop a metaheuristic approach.

MULTI START ITERATED LOCAL SEARCH PROCEDURE

A Multi Start Iterated Local Search Procedure (MSILS) has showed to be very competitive compared to some usual methods applied to VRP by Prins (2009). Besides its high performance, it has a simple design which makes it relatively easy to be implemented. It proceeds by restarting an Iterated Local Search Procedure (ILS) with different randomly generated initial solutions as illustrated in *algorithm 1*. This ILS metaheuristic performs by iteratively applying a perturbation method followed by a local search procedure to an initial solution. The general framework is based on iterating an ILS procedure by randomizing the subroutine which provides starting solutions to be improved. At each ILS iteration, an initial heuristic solution (*Soln*) is first improved by a local search method (LS). Iteratively, incumbent solution structure is attempted to be modified by a perturbation procedure (Perturb) followed by the application of a LS. In each ILS, number of attempts for solution improvements is limited. More specifically, each new iteration of Perturb and LS methods is run under the condition that a maximum number of iterations (*MaxGlobalIter*) is not violated. Also, after a number (*MaxStuck*) of unsuccessful attempts to improve current solution or in case of maximum number of LS iterations (*MaxIter*) is reached, a new ILS is started. The best solution (*Soln**) computed by each ILS is compared with the best general solution (*BestSoln*) which is then updated when necessary.

Algorithm 1. General structure of MSILS

```
Initialize BestSoln with a great value
GlobalIter = 0
Repeat
    // Inside this loop ILS procedure is repeated
    A randomized Clark and Wright heuristic generates
    Soln
    BestSoln = Soln*
    Soln* = Soln
    LS(Soln)
    Iter = 0
    Stuck = 0
    Repeat
        GlobalIter = GlobalIter + 1
        Iter = Iter + 1
        Perturb(Soln)
        LS(Soln)
        If Cost(Soln) < Cost(Soln*) Then
            Stuck = 0
            MaxPerturbationLevel = 1
            Soln* = Soln
        Else
            PerturbationLevel = Min ( PerturbationLevel
            + 1, MaxPerturbationLevel)
            Stuck = Stuck + 1
        End if
    Until (Iter = MaxIter Or Stuck = MaxStuck Or
    GlobalIter = MaxGlobalIter)
    If Cost(Soln*) < Cost(BestSoln) Then
        BestSoln = Soln*
    End If
Until (GlobalIter = MaxGlobalIter)
```

Starting solution

The parallel version of the Clarke and Wright savings heuristic (Clarke and Wright, 1964) is adapted to the specificity of the MC-VRP studied here. This method is a constructive heuristic which proceeds by first assigning each customer to a distinct route. Savings are computed by comparing the distance of the shortest route connecting each possible couple of customers to the sum of their individual distances to the depot. Then, a savings matrix is created and is sorted in descending order of savings values. Final routes are created by merging routes listed in savings matrix from the beginning to the end while checking new routes feasibility constraints. For multiple starts purpose, we randomize the Clarke and Wright by slightly disorganizing the sorting order of savings matrix. Strictly sorted saving values are divided into small groups of consecutive values. Into each group, initial order is randomly perturbated in order to obtain a randomized Clarke and Wright method.

Local Search

Local search procedure consists of exploring a solution neighborhood seeking for better solutions. The exploration mechanism is based on some specific moves to go from one solution to another in the search space around. Our local search procedure involves several well-known moves. Inter-trip and intra-trip moves which operate transformation respectively on one and two trips are investigated. We refer to first improvement local search method to explore the solution space. Exploration is done by applying moves features successively to each customer in the same order as the trip sequence. The moves used in the search process are relocation, exchange and 2-Opt.

Relocation move: This operator relocates a customer x_i from an initial position to another one. One or two trips may be involved.

Exchange move: This type of move swaps customers x_i and x_j belonging to two distinct trips or the same route.

2-Opt move: This move operates modification on two edges with no common node. Two different situations may be encountered. In case edges are part of a same route, the section from the node at the beginning of the first edge until the starting node of the second edge is inverted. Let's consider a route A with the following sequence of customers $\{x_1, \dots, x_i, x_{i+1}, x_{i+2}, ..., x_j, x_{j+1}, x_{j+2}, \dots, x_n\}$. We apply 2-Opt move on edges $\{x_i, x_{i+1}\}$ and $\{x_j, x_{j+1}\}$. We thus obtain this new sequence $\{x_1, \dots, x_j, ..., x_{i+2}, x_{i+1}, x_i, x_{j+1}, x_{j+2}, \dots, x_n\}$ where subsequence from node x_i to x_j is inverted. When the edges are in different routes, the mechanism first splits both edges and connects the initial section of route A to the ending one of the route B and vice versa. Let's consider two routes A and B, such that $A = \{x_1, \dots, x_i, x_{i+1}, x_{i+2}, ..., x_j, x_{j+1}, x_{j+2}, \dots, x_n\}$ and $B = \{y_1, \dots, y_i, y_{i+1}, y_{i+2}, ..., y_j, y_{j+1}, y_{j+2}, \dots, y_n\}$. In order to apply 2-Opt move on edges $\{x_i, x_{i+1}\}$ and $\{y_j, y_{j+1}\}$, we first divide them so that their respective extremities are disconnected. Afterwards, the remaining sections of routes A and B are reconnected in way that they are modified like this: $A = \{x_1, \dots, x_i, y_{j+1}, y_{j+2}, \dots, y_n\}$ and $B = \{y_1, \dots, y_i, y_{i+1}, y_{i+2}, ..., y_j, x_{i+1}, x_{i+2}, ..., x_j, x_{j+1}, x_{j+2}, \dots, x_n\}$.

Perturbation

Perturbation procedure reduces probability of being trapped in local optima during the search process by diversifying solutions structures. It consists in applying one or a few random moves to a solution. Our perturbation method used exchange move on a couple of trips selected randomly and customers to be swapped are also randomly chosen. During exchange process, a customer from one trip is exchanged to another one. In the perturbation process, a random exchange move is attempted iteratively until a stopping condition (*MaxPerturbationIteration*). This condition is set to true if the maximum number of exchange move attempts is reached or an improved solution is obtained while applying perturbation movements.

By default, a moderate level of perturbation is applied to a solution before calling local search for improvement. When local search fails to improve a perturbated solution, the level of perturbation is intensified in order to escape local optimum of the current neighborhood. As soon as new improvements are achieved, perturbation is reset to a moderate level.

Algorithm 2. Perturbation

```
Soln1= Soln
Improve = false
PerturbationLevel = 0
Repeat
    PerturbationLevel = PerturbationLevel + 1
    PerturbationIteration = 0
    Repeat
```

```
      PerturbationIteration = PerturbationIteration+1
      Pick up randomly two distinct trips (T1, T2) from Soln1 list of trips
      For each trip, pick up randomly two customers (C1, C2), one from each trip
      // Apply perturbation move
      Soln1 = Customers Exchange (C1, C2)
      If (Feasibilty (Soln1)) Then
         PerturbationMoveValid = true
         If (Cost(Soln1) < Cost(Soln)) Then
            Improve = true
            Soln = Soln1
         End If
      Else
         PerturbationMoveValid = false
         Soln1 = Soln
      End If
   Until (PerturbationMoveValid Or PerturbationIteration = MaxPerturbationIteration)
Until (PerturbationLevel = MaxPerturbationLevel Or Improve)
```

COMPUTATIONAL RESULTS

For experimental purpose, our MS-ILS algorithm is coded in C++ and tested on a computer equipped with a dual core processor at 2.26 GHz and 2GB of RAM. As MC-VRP is a quite new problem, to the best of our knowledge, very few specific instances exist in literature. Kalkoff (2006) developed an instance generator capable to create MC-VRP instances according to predefined problem characteristics. El Fallahi *et al.* (2008) created two sets of modified Christofides VRP instances specially adapted to a MC-VRP with 2 compartments. One of the set of instances has been obtained by dividing the capacity of each vehicle and each customer request into two equal parts. This set is used for computation in this paper. MS-ILS results are compared to those computed by a tabu search and a memetic algorithm developed by El Fallahi *et al.* (2008).

Parameters tuning

A sensitivity analysis was made in order to find a compromise between solution quality and computation time. Mainly, the impact of the number of local search procedure iterations (*MaxGlobalIter* and *MaxIter*) on computation time is tracked. In order to make parameters tuning, we run tests on an adapted version of the first Christofides instance (*vrnpc1*). For example, we noticed that from a number of 1500 local search iterations, there is very slight solution improvement for longer running time. Finally, parameters values are set in order to reach a good compromise between time cost and solution quality. The different parameters are set to the values indicated in Table 1.

Results

Compared to results found by El Fallahi *et al.* (2008) (Table 2) in the case of MC-VRP without split delivery, our MS-ILS brings better results. Their Tabu Search, which provides better average results than their Memetic Algorithm, is outperformed by approximately 2% (1.9% precisely). Computation time is 15.4% less than El Fallahi *et al.*'s reported time on our machine equipped with 2.26 GHz dual core processor. By the contrary, running time increased by 52.2% comparatively to their Memetic Algorithm while its results are improved by 2.7%. For any Christofides instance, MS-ILS performs at least as well as their Tabu Search. Individual results improvement varies from 0% to 5.9% for TS and from 0% to 8.3%, which proves that MS-ILS is strictly more competitive than El Fallahi *et al.* (2008) methods.

CONCLUSION

Through this paper, MC-VRP, a quite unexplored problem in VRP literature is investigated. We propose an efficient optimization method based on a Multi Start Iterated Local Search (MS-ILS) algorithm to solve it. Experiments have been conducted on modified Christofides instances. A comparative analysis is made between MS-ILS and the best known method developed by El Fallahi *et al.* (2008). They proposed a Tabu Search and a Memetic Algorithm to solve MC-VRP. MS-ILS developed here is shown to be strictly more competitive than each one of El Fallahi *et al.* (2008)'s best method. Their best algorithm in terms of solution quality, the Tabu Search, is outperformed by almost 2% by our metaheuristic.

MS-ILS parameters

MaxGlobalIter = 1600	*MaxIter = 900*	*MaxStuck =400*
MaxPerturbationIteration = 50	*MaxPerturbationLevel = 3*	

Table 1: Metaheuristics parameters values

MS-ILS and ILS results

File	El Fallahi et al. (2008) MC-VRP (*without splitting*)				MC-VRP (*without splitting*)			Relative gain			
	MA		**TS**		**ILS**	**Multi Start ILS**		**MS-ILS vs MA (El Fallahi)**		**MS-ILS vs TS (El Fallahi)**	
	Cost	***Time***	***Cost***	***Time***	***Cost***	***Cost***	***Time***	***Cost***	***Time***	***Cost***	***Time***
vrpnc1	**524.6**	17.4	**524.6**	19.6	538.9	**524.6**	15.7	0.0%	9.8%	0.0%	19.9%
vrpnc2	855.8	25.4	850	56.3	854.7	843.0	21.2	1.5%	16.5%	0.8%	62.3%
vrpnc3	876.8	21.8	831.3	50.9	828.3	828.3	64.4	5.5%	-195.4%	0.4%	-26.5%
vrpnc4	1089.6	93.9	1061.1	285.4	1049.0	1048.2	109.2	3.8%	-16.3%	1.2%	61.7%
vrpnc5	1389.6	115.9	1348.3	403.6	1330.2	1330.2	161.9	4.3%	-39.7%	1.3%	59.9%
vrpnc6	571.4	6.1	575.9	12.3	576.6	560.3	8.1	1.9%	-32.8%	2.7%	34.1%
vrpnc7	933	39.2	970.8	21.8	925.1	925.1	18.1	0.8%	53.8%	4.7%	17.0%
vrpnc8	969.2	18.7	888.6	60.4	888.4	888.4	40.0	8.3%	-113.9%	0.0%	33.8%
vrpnc9	1230.9	98.7	1232.1	98.3	1198.1	1192.2	31.4	3.1%	68.2%	3.2%	68.1%
vrpnc10	1520.1	140.2	1538.6	35.2	1448.2	1448.2	61.6	4.7%	56.1%	5.9%	-75.0%
vrpnc11	1046.1	47.8	1043.3	142.5	**1042.1**	**1042.1**	75.2	0.4%	-57.3%	0.1%	47.2%
vrpnc12	820.6	18.2	819.5	63.1	**819.6**	**819.6**	66.4	0.1%	-264.8%	0.0%	-5.2%
vrpnc13	1547.4	76.4	1582.2	44.3	1544.7	1544.7	371.5	0.2%	-386.3%	2.4%	-738.6%
vrpnc14	866.9	23.3	868.6	43.5	**866.4**	**866.4**	86.1	0.1%	-269.5%	0.3%	-97.9%
Total	**14242**	**743**	**14134.9**	**1337.2**	**13910.3**	**13861.3**	**1130.8**	**2.7%**	**-52.2%**	**1.9%**	**15.4%**

Table 2: Comparison of results found by MS-ILS for MC-VRP without split deliveries El Fallahi et *al.*'s solutions

REFERENCES

Avella P, Boccia M, Sforza A (2004). "Solving a fuel delivery problem by heuristic and exact approaches". Journal of Operational Research 152(1):170–179

Chajakis E, Guignard M (2003). "Scheduling deliveries in vehicles with multiple compartments". Journal Global Optimization 26(1):43–78

Christofides N, Mingozzi A, Toth P (1979). "The vehicle routing problem". In: Christofides N, Mingozzi A, Toth P, Sandi C (eds) Combinatorial optimization. Wiley, Chichester, pp 315–338

Clarke G, Wright J (1964). "Scheduling of vehicles from a central depot to a number of delivery points". Operations Research 12:568–581

Cornillier F, Boctor F, Laporte G, Renaud J (2008). "A heuristic for the multi-period petrol station replenishment problem". European Journal of Operational Research 191(2):295–305

Derigs U., Gottlieb J., Kalkoff J., Piesche M., Rothlauf F., Vogel U. (2010). "Vehicle routing with compartments: applications, modelling and heuristics". OR Spectrum. doi:10.1007/s00291-010-0194-3

El Fallahi A, Prins C, Wolfler Calvo R (2008). "A memetic algorithm and a tabu search for the multicompartment vehicle routing problem". Computer & Operation Research 35(5):1725–1741

Kalkoff J (2006). "Generierung von Benchmarks und empirische Analyse von Metaheuristiken für Tourenplanungsprobleme mit teilbaren Frachträumen". Diplomarbeit, Lehrstuhl für Wirtschaftsinformatik I, Universität Mannheim

Mendoza JE, Castanier B., Guéret C, Medaglia AL, Velasco, N (2010). "A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands". Computer & Operation Research 37(11):1886 – 1898

Prins, C. (2009). "A GRASP×Evolutionary local search hybrid for the vehicle routing problem". In:Pereira FB,Tavares J, editors.Bio-inspired algorithms for the vehicle routing problem, studies in computational intelligence, vol.161. Berlin: Springer, p.35–53

Van der Bruggen L, Gruson R, Salomon M (1995). "Reconsidering the distribution structure of gasoline products for a large oil company". European Journal of Operational Research 81(3):460–473

BIOGRAPHY

Cadet David JOSEPH was born in Port-au-Prince, Haiti and graduated in industrial engineering program from Quisqueya University (Haiti) in 2008. After, he travelled to France where he successfully attended a master program in operations research proposed by Arts et Métiers ParisTech engineering school and University of Metz. From October 2009, he is a Ph'D student at the Industrial System Optimization Laboratory (LOSI) of University of Technology of Troyes (France). The scope of his thesis is about vehicle routing problems optimization in the context of a food industry supply chain.

SIMULATION APPROACH FOR AN INTEGRATED DECISION SUPPORT SYSTEM FOR DEMAND RESPONSIVE TRANSPORT PLANNING AND OPERATION

Ana Dias
José Telhada
Maria Sameiro Carvalho
University of Minho, Systems and Production Department, Campus Gualtar, 4710-057 Braga, Portugal
E-mail: {anadias, sameiro, telhada}@dps.uminho.pt

KEYWORDS
Demand responsive transport (DRT), integrated decision support system (IDSS), simulation approach, performance evaluation.

ABSTRACT

Rural areas are becoming more desert from day to day, leading to complex dispersed and scarce demand patterns for public transport. As a consequence, conventional transport services are becoming less frequent, reducing levels of service (e.g., low occupancy rates, usage of old vehicles). With rigid predefined routes and schedules, they are inappropriate to operate in such environments.

Demand Responsive Transport (DRT) systems have been seen as an interesting alternative solution, providing flexible transport services to meet trip requests. This solution has already been adopted in several countries as a way to increase user's mobility and mitigate social exclusion. There are however some issues concerning DRT scheme design and evaluation requiring further developments. Namely, there is still a lack of adequate tools to support some of the strategic and tactic level decisions that must be made at the design phase.

In this research, we propose an Integrated Decision Support System (IDSS) and general action methodology that will allow achieving better planning decisions and allowing the evaluation of alternative scheme designs prior to its implementation. The IDSS and methodology are based on an event-driven simulation framework which emulates real-world customers' behavior and vehicles movements. The paper will concentrate its analysis on this framework. An illustrative numerical experiment is presented and briefly discussed.

1. INTRODUCTION

Rural areas are characterized by low levels of population density, which is unsuitable for a conventional public transport service (with fixed routes, fixed stops and fixed schedules). In fact, traditional public transport systems had revealed low success and large economic inefficiencies in areas with widely dispersed trip patterns, due to low vehicle occupation rates. As a consequence, transport operators tend to cease all or part of their operating concessions, leading to mobility and functional exclusion problems for residents.

In the last decades some efforts have been developed to overcome some of these problems. Some countries have been approving legislation to allow the implementation of flexible public transport systems, such as Demand Responsive Transport (DRT) systems, which services (including their schedules, routes, stops and vehicle size allocation) are triggered and defined dynamically by demand calls. Additionally, a substantial number of on-site DRT experiences have been promoted by central governments funding schemes. For example, in USA, central government policies (ADA 1990 - Americans with Disabilities Act (Brake *et al.* 2007)) require that transport operators running fixed-route services also provide complementary paratransit services for persons with disabilities within their service areas. In England, the Rural Bus Challenge funding that was launched in 1998 promotes the adoption of innovative transport schemes that includes flexible transport services (Enoch *et al.* 2006). In some other EU countries the DRT supply was also supported by EC funds, integrated in projects with the aim of assessing the most effective ways of providing rural transport services and producing a set of recommendations to serve as a guide for the planning and implementation of rural transport systems (Brake *et al.* 2007).

During the last decade, an increasing number of scientific articles have been published, supporting the greater concern of academics on new transportation solutions (including DRT) to address people´s mobility problems. All of these contributions and field experiences allow us to identify the most critical factors (financial, social, environmental, operational, etc.) that should be carefully analyzed when setting up a new DRT scheme.

However, a systematic sustainability analysis of transport system is still a complex issue since it faces a number of specific characteristics associated not only to the nature of its demand and supply, as well as to all its externalities. Transport problems are inherently complex and difficult to handle, and intelligent decisions must be oriented towards maximizing the advantages of the new transport provision while minimizing their costs and undesirable side-effects. Since resources are always scarce, a major effort in adopting efficient investment decisions is required. In this way, further research is still required to assess to which extent they contribute to the success of a DRT scheme or what combination of options should be adopted.

In this research, an integrated decision support system (IDSS) is proposed aiming to support decision makers in designing and implementing by firstly investigating the impacts of alternative potential specifications of the DRT system on its viability and sustainability for a given territorial area. The decision support system combines supply and demand data with analytical and simulation tools in order to provide information that will support top management strategic and tactical decisions.
The objective of this paper is to describe, more particularly, the simulation module and its role in the IDSS, and illustrate results of some preliminary tests in order to register, in advance, the quality of solutions or management strategies.
The remainder part of the paper is organized as follows: Section 2 presents a literature review on DRT systems in order to highlight the main conceptual and design aspects, technologies and methodologies applicable. Section 3 describes a new IDSS framework for setting up and run a DRT. Section 4 presents a more detailed description of the simulation component, including some aspects that concern the creation of demand and supply sub-models from real data and information. Section 5 briefly introduces the general procedure to conduct a DRT design process by using the IDSS framework and its simulation component, in particular. Section 6 illustrates the applicability of this procedure by a simple real-like case. Finally, Section 7 reports the main conclusions and final considerations regarding future developments.

2. LITERATURE REVIEW

Flexible transportation systems, in particular, DRT systems have been adopted over the last decades, as reported in some studies (Brake *et al.* 2004; Mulley and Nelson 2009). These systems provide transport on demand from users, using flexible schedules and routes to satisfy their travel needs. A DRT system receives trip requests either for an immediate service or as an advanced reservation and organizes routes and schedules to accommodate trip requests aiming to respond in real time to user's mobility needs. Its implementation typically involves the use of information and communication technologies comprised in an Operational Decision Support System (ODSS) in a Travel Dispatched Center (TDC), as shown in Figure 1.

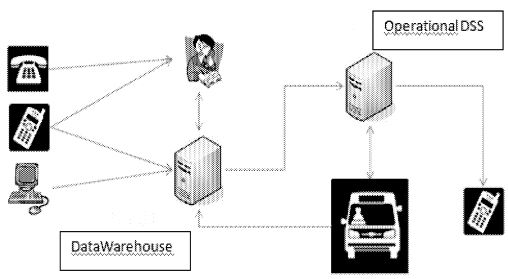


Figure 1: Elements of a TDC in a DRT System

Trip requests are typically made using telephone. In most advanced systems, requests (and related monitoring) can also be made by interacting to a WEB portal. Requests are then stored in a data warehouse which also holds all the relevant data concerning the transportation network. A fleet of vehicles of mixed types (buses, mini buses, taxis, etc.) is generally provided by one or more public and private operators (taxi owners, bus operators, community transport, etc.). Services can be operated on their own or integrated with traditional transportation systems, acting as feeder services for buses or rail services. The heterogeneous fleet of vehicles is coordinated by the TDC that, in some cases, is operated with advanced information and communication technologies such as on-board integrated Global Positioning System (GPS) and/or continuous General Packet Radio Service (GPRS) connection. Vehicles are assigned to trip requests frequently by the ODSS that incorporates some intelligence or rationality to the system, allowing obtaining adequate transport solutions according to the area characteristics and demand patterns.
The ODSS should integrate three different types of functions: requests management, scheduling and routing dynamic planning, and effective communication system between the TDC and customers and TDC and vehiches (*e.g.* SAMPLUS 2000; McDonagh 2006). Furthermore, it should export detailed reported service data, allowing carrying out performance analyses to define corrective measures for future implementation. Over the last two decades or so, a few ITS have been developed such as Mobisoft and Trapeze (Enoch *et al.* 2006), and more recently, Astra (Dias *et al.* a) 2011).
The use of modern Information and Communication Technologies (ICT), including Intelligent Transport Systems (ITS), allied to adequate strategy planning services has been pointed out as the solution to improve the costs-effective performance of DRT services by promoting more complex organizational structures, i.e. collaboration of multiple service providers, and improving flexibility and popularity by providing intelligence solutions to process trip requests using dynamic routes definition and an accurate fulfillment of the requirements, in order to respond in real time to users mobility needs (Nelson and Phonphitakchai 2012; Mageean and Nelson 2003; Ambrosino *et al.* 2004; Brake *et al.* 2004; Brake *et al.* 2007; INTERMODE 2004; Mulley and Nelson 2009).
Most of the work developed so far identifies as key success factors a set of strategic level decisions concerned with system conceptualization issues: specific policy goals, target market, area factors (population density, income level, demographic details, land use pattern, etc.), stakeholders involved, financial model and the regulatory framework. From these issues the financial aspect is assumed as one of the most critical aspects: what sources of funding are available to support innovative transport services? What is the time limit of those funds? (Very often, those funds, including UE funds, only apply during the setup phase, putting at risk its permanent sustainability. The lack of funds and

subsidies for innovative transport services can be a crucial aspect or, at least, an important obstacle to the success of the systems, in many cases.) Which institutions are going to give a contribution to provide financial means? What is going to be the passenger contribution (fares)?

In fact, a major concern pointed out in several analysis concerning DRT implementations is related with its financial sustainability. Indeed, DRT are labor-intensive and the cost per passenger is relatively high, comparable to taxis: fares revenues are unlikely to cover costs in any market (Mageean and Nelson 2003). Therefore, these projects tend to close after the first pilot stage (Battelino 2009) or when funding schemes end. In fact, according to Enoch *et al.* (2006), the imbalance between income and costs is only manageable when it is supported by (central, regional or local) government funding, and financial viability tends to end when funding finishes.

At the tactical level, decisions aim to devise ways to implement the transport system according to the strategy previously defined. This planning stage is crucial for the design of the transport scheme and several authors had identified the most critical decisions (*e.g.* Giannopoulos 2004; Mulley and Nelson 2009; Parragh *et al.* 2008; Quadrifoglio and Li 2009), such as:

- What level of flexibility should be adopted, in terms of routes, timetables, frequencies, time-windows (ex, arrivals at stops)?
- What pre-booking mechanisms and rules should be adopted?
- What resources are going to be used (fleet, drivers, informatics, TDC center and staff)?
- What fare structure should be implemented?
- Which level of integration with public transport network (schedule buses/train network, etc.)?
- How is the system going to be evaluated?

It is well-known that all these factors have an impact on the success of DRT systems. However, further research is still required to investigate what combination of options should be adopted and to assess in which extent they contribute to that success. Additionally, it is fundamental to estimate the impact that some of these decisions have in terms of overall system's performance (cost, quality, etc.).

Modeling is a fundamental tool to support large investments decisions which involve social, economic, financial and institutional complex issues such as new and innovative transport solutions. In spite of this, only a few attempts have been made to develop approaches capable of giving insight on the relationship between design options and their impacts on system's performance. In general, several measures or indicators can be used to evaluate a DRT service. Examples of those are (ex, Mageean and Nelson 2003; Brake and Nelson 2007; Fernández *et al.* 2008; Palmer *at al* 2008; Quadrifoglio and Li, 2009, 2010): the service reliability (customers' satisfaction, willingness to pay, non-accomplished valid requests, general mobility improvement), core background information, service passenger restrictions (ex, only disabled and elderly people, or other mobility-constrained persons), trip purpose, coverage of service (which days it works, where it works), easiness of reservations, pre-booking antecedence and time windows for reservations, passenger convenience (the time spent in transit becomes less satisfactory as the service area increases), need for intermodal transfers, satisfaction with the TDC, driver satisfaction, walking time to stops, waiting time for delayed vehicles. However, according to Battelino (2009), there is a lack of research work into evaluation methods and definition of real time evaluation and monitoring systems.

Wilson *et al.* (1970) pioneered the use of simulation to compare different heuristics to assess the influence of the service area, demand density, and service quality on the fleet size requirements. Some studies (ex, Feuerstein and Stougie 2001) have investigated changes on performance when the dial-a-ride system runs with alternative number of vehicles.

Dias *et al.* (2011a) proposed a framework that uses a simulation approach to evaluate alternative DRT solutions and select the most adequate for a given territorial rural area and for each of different scenarios. Their approach will permit to achieve better planning decisions and will allow evaluating operating strategies prior to the implementation of such a complex system, as it is the case of a DRT system. The conception and the performance evaluation of a DRT system includes establishing user requirements and expectations, by considering all perspectives from the beginning, running computational models and creating simulated prototypes, analyzing all the main issues (economic, social, and environmental) at its conceptual stage in a holistic way. The simulated DRT system solution should integrate all sectors and alternative scenarios in order to produce better services for users and more efficient and sustainable operations for operators.

In the next section, an extension to the proposal of Dias *et al.* a) (2011) will be presented and discussed. This proposal consists on a broader framework that integrates different advanced technologies and analytical methodologies, and allows a comprehensive decision support to analysts and managers at the different levels of decision: strategic, tactical and operational.

3. INTEGRATED DECISION SUPPORT SYSTEM

Figure 2 illustrates the proposed integrated decision support system (IDSS) for a DRT system, pointing out its main components and sub-components, their relationships and the different levels of decision: strategic, tactical and operational (ODSS). For example, at the design phase of the DRT system, no real information exists yet on its functioning, so it is considered as a strategic and tactical decision process. In this case, the simulator component must be used in order to emulate what could happen at a real-world scenario, in order to allow the evaluation of different options (essentially, system objectives and rules) by the analyst

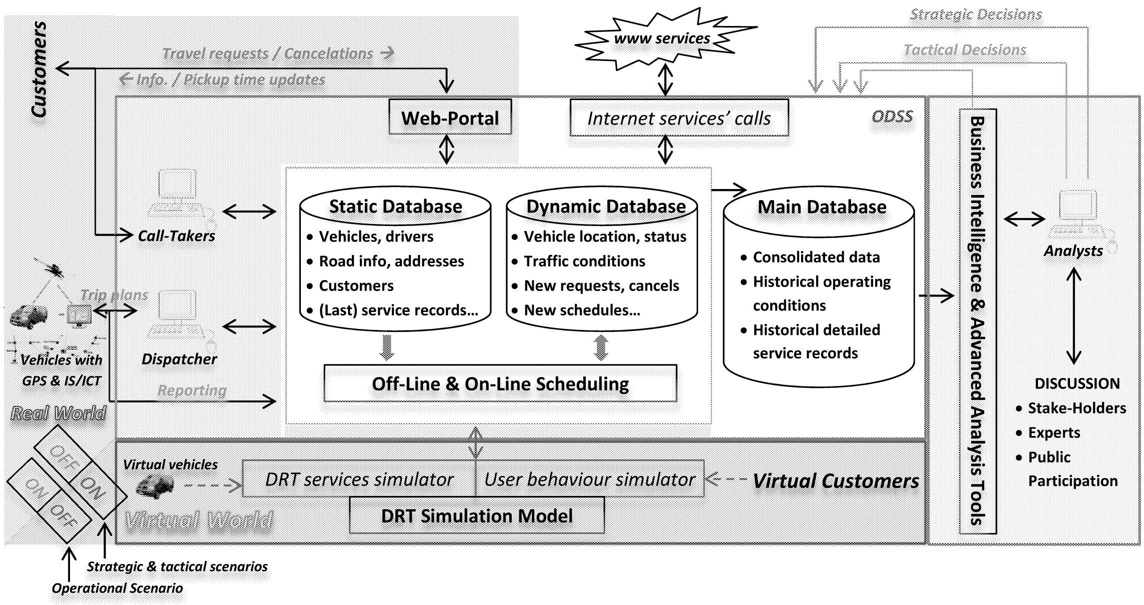


Figure 2: IDSS for Different Levels of Decision in DRT Systems (adapted and extended from Fu, 2002)

component. Details of the simulation component are described in the next section thereafter.

The ODSS component represents the heart of the TDC center of the DRT system, receiving trip calls, processing them in terms of rational/optimized service (route) plans and scheduled services, giving advanced information to customers, monitoring performance and recording detailed historic data. Routes and schedules are solved by applying alternative algorithms, automatically selected according to the patterns of trips. For example, in the case where groups of persons are at the same stop (work-to-home, market-to-home and school-to-home trips), a Vehicle Routing Problem savings-like heuristic can be applied, whereas in the general case where vehicles have to pick up and drop off different customers at/to different stops across the territory, a dial-a-ride problem (DARP) should apply. Currently, the ODSS component incorporates two alternative solution methods for DARP: (1) a fast heuristic adapted from Xiang *et al.* (2006), and (2) an exact method adapted from Lu and Dessouky (2004). Note that the exact solution method aims to solve small instances of the DARP (number of customers per trip less than 8 to 10) and there is enough time to find the solution. So, such a method is adequate to be applied in a real operating context only; so, it is not applied at the design phase of the DRT, where fast processing simulation runs are needed.

The analyst component is used to properly analyze and evaluate DRT options. It is powered by a Business Intelligence (BI) type framework that starts to extract relevant data from the historic main database, transform such data and load it into a proper database system for multi-specialized analyses. It comprises different methodologies: simple and advanced statistical reporting and inference techniques, data mining and operational research inference and prospective models.

Social, environmental and economic impacts are increasingly important aspects for decisions in transport investments. To take care of this, the analyst component has been incorporating a large number of key performance indicators (KPI) hierarchized by their measuring aim (e.g., evaluate client satisfaction, financial performance, organizational performance, etc.) and categorized by the three basic assessment dimensions of sustainability: economic, environmental and social. Details and examples of such KPIs, currently implemented in the proposed framework, can be consulted in Dias *et al.* b) (2011).

The IDSS integrates several types of advanced modern technologies (Figure 3). According to Liu *et al.* (2010), an IDSS which combines the use of models, analytical techniques, data access and retrieval functions, by using advanced differentiated technologies, as it is the case in here, will have the advantage (in relation to traditional "static" DSS) in providing consistent, coordinated, active and global support for the various managers/analysts on their different levels of decision-making processes.

Furthermore, beside the simulator and business intelligence modules, the ODSS proposed includes other characteristics, such as, support to advanced communication systems, Geographic Information Systems (GIS) for analyses and map display purposes, access to Google Maps API services tools integration.

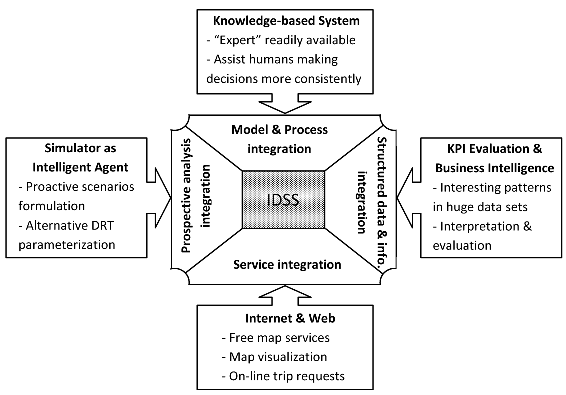


Figure 3: Types of Technologies to Enhance Integration for IDSS in DRT Systems (adapted from Liu *et al.*, 2010)

The IDSS helps DRT managers (TDC coordinators, system designers and analysts) at their different levels of decision:

- Strategic level (objectives, aim, strategies and general policies) – the analyst component decides how the DRT should be operating in the long-term in order to be viable and sustainable in the three basic terms: economic, social and environmental (what the main objectives of its existence? which type of services must offer? at what level(s) of flexibility? at what price levels (whether taking account or not potential subsidization)?
- Tactical level (detailed rules according to objectives and general strategies and policies) – the analyst component monitors and analyses current performance of the system, tries to identify hidden patterns of operational data, and continually tries to devise better solutions to tackling operational problems; some solutions are automatically devised and incorporated into the ODSS component (e.g., a recurrent set of day to day fixed travel patterns are identified and a shortcut route planning procedure automatically generates a fixed service plan); however, the most part of solutions requires the investigation and validation of human analysts before their practical implementation (which new operating rule should be applied? is there any actual rule of policy that must be re-parameterized? what are the new parameters?).
- Operational level (every-day decisions, according to rules) – the ODSS component drives the TDC center, receiving travel requests and determining the most appropriate service plans (which vehicle? what route? what schedule?).

4. SIMULATION COMPONENT APPROACH

The simulation component comprises two main models: (1) a demand-side model implemented as a travel request generator and users' behaviour simulator, and (2) a supply-side simulator that simulates the functioning DRT services, including the decisions made by the ODSS and the vehicles operations.
Both of these models are based on a micro-simulation event-driven approach. The main demand-side events are: trip reservation, trip cancelation, user arrival to stops (origin and destination), user no-show and non-reserved shows at stops. The main supply-side events are: trip planning events (such as hourly and daily time-scheduled planning of advanced reservations, real-time acceptances or rejections and re-planning), vehicle departure and arrival from/to stations, stops and strategic waiting and interface points.

Demand-side model

This model simulates the behaviour of users, from travel requests at each stop of the network by specifying the desired destination and the time of departure or arrival, to the user behaviour of each user with respect to their arrival time at the origin stop, waiting "patience" for delayed vehicles, boarding times, and so on.
Travel requests are generated based on socio-economic characteristics of the resident population (from Census), from domiciliary questionnaires and local authorities interviews, as well as acquired knowledge about the main attracting poles for visiting (workplaces, schools, hospitals, markets, general services, transfer stops to inter-urban transport services, etc.). Such data and acquired information is used to compose macro-descriptive variables small zones of the territory (highest level of disaggregation in accordance to Census). The resulting model starts by simulating the individual patterns of each user by using those macro-descriptive variables. Some degree of freedom (in choosing any random destination) is also modelled by a given probability (as a model parameter) and performed by using a gravity model.
User time arrivals to origin stops is modelled by recurring to alternative distribution probabilities such as triangular, truncated normal and truncated negative exponential distributions.

Supply-side model

This model reproduces a virtual functioning of the supply elements of the DRT system, namely certain modules of the ODSS (DARP planning, vehicle allocation and dispatching, historical data storage in database), and all vehicles' movements.
Travel times between two adjacent schedule stops are generated by parameterized normal distributions. Mean times are taken from OD trip times previously stored in the system database and obtained by invoking Google Maps internet services (shortest route between two points). Some non-recurring real events, such as vehicle breakdowns, road obstacles and bad weather conditions, are also considered according to parameterized Poisson processes.
Recurrent congestion inside (small) urban areas are also modelled by using: (1) mean travel time forecasting models based on chronologic time series of experienced trips and their representation in terms of exponential moving average and ARIMA mechanisms; and (2) lognormal distributions.

5. DRT DESIGN APPROACH

As it was discussed before, in Section 2, the design phase of a new DRT system involves the investigation of a large set of issues and the answer to a large set of strategic and tactical questions.
In terms of the IDSS usage, the simulator component must be "turned on", "tuning off" automatically the sub-components of the ODSS that communicates to real customers and vehicles (i.e., the call-taker, the dispatcher

and the Web portal sub-components) – see Figure 2 above. The simulator will take implicitly account all of their necessary functions, communicating directly with the remaining operational sub-components.
A framework was identified as essential to the process evaluation (Dias *et al.*, b) 2011) based on the literature review (Figure 4). The referred framework is used to determine the system performance to evaluate the alternative specifications for a certain area. It allows choosing the better specifications and the better working rules.

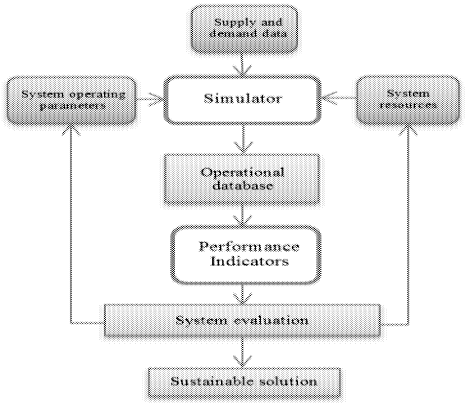


Figure 4: Conceptual Framework of the DSS

An iterative approach is used between the simulator and the analyst components. At each iteration, a solution option is simulated and evaluated. And, for each iteration (option) a large set of simulation runs (days) are realized and their operational details are stored into the main database of the ODSS. The number of days is set in order to infer statistics (e.g., mean values) within a given high level of accuracy. After that, such set of data is analyzed in terms of KPIs that are automatically computed by the analyst component's routines.
In the next Section, some illustrative results of a given iteration of the above process is reported and discussed. It shows the software viability in theoretically speaking (demand is defined randomly).

6. ILLUSTRATIVE EXAMPLE

A small case study has been implemented in a rural area in Minho region, north of Portugal (Amares, Terras de Bouro and Vila Verde counties, with a population density of 155.3 inhabitants/Km2, in average). However, the illustrative results presented hereafter refer to the area of Terras de Bouro, a county with the lowest population density of the region (mean of 26.14 inhabitants/Km2). The area is very large and the population is low and highly dispersed. As a result it is expected that the resulting vehicle occupation rates from simulation runs are low.
The ODSS uses GIS technology by integrating the *Google Maps* service. *Google maps* allows the representation of geo-referenced information on a map in a very user-friendly (and it is a free service). So, the graphical visualization of a particular day run, for a particular vehicle, is displayed in Figure 5. Green points represent pick up points and red points represent drop off points (or both, if there is anyone to pick up there).

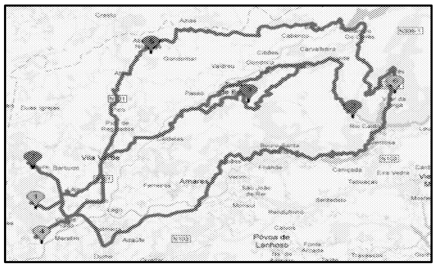


Figure 5: Routes of a Vehicle during a Particular Day

The GIS technology, by integrating the Google Maps service, would be also interesting for customers, as they can see their planned trips online (Web-Portal), for example for a given week. Every line with a different color corresponds to a different route of the vehicle (Figure 6).

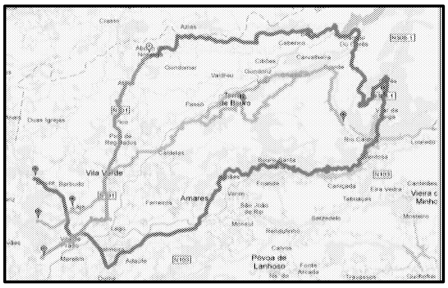


Figure 6: User Interface of the System

A large set of KPIs are automatically produced by the analyst component of the IDSS in order to allow analysts and decision-makers to assess the performance of the DRT specification being simulated (or in use, if the system was implemented). Table 1 shows some KPIs obtained from a series of simulation runs (working-day unchanged pattern) for a particular DRT specification, i.e. set of fixed rules and parameters.

Table 1: Some KPIs for a Series of Simulation Runs (Working-Days) – values are rounded to integer numbers (means per day)

Requests number	51
Passengers number	103
Working hours: from 5:30 am to 8:30 pm	15
Mean requests per hour	3 - 4
Mean passenger number per hour	7
Mean travel distance (passenger)	30 km
Mean travel time (passenger)	42 min
Mean delays (passenger)	16 min
Mean waiting time (delays vehicle)	4 min
Mean distance between stops	11 km
Mean time between stops	16 min
Vehicle number	4
Mean occupation rate (total)	24%
Mean occupation rate (partial)	44%
Mean vehicle time (without passengers)	2h 49min
Mean vehicle time (with passengers)	5h 32min

The “Requests number” and “Passengers number” (mean carried per day in one direction) are produced by the demand model, generated randomly. It was considered a working day of 15 hours from 5 am to 8:30 pm. As a result, the “mean request per hour” varies from 3 to 4, and the “mean passenger per hour” is about seven. The “mean travel distance per passenger” is approximately 30kms in about 42 minutes. The “mean passenger delay” is nearly 16 minutes, but the “vehicles waiting time” in the stop is only about 4 minutes. The “mean distance between stops” is approximately 11kms, taking, on average, 16 minutes “mean time between stops”.
The “vehicle number” used to serve all the requests is four (one with four seats, another with six seats and two vehicles with 8 seats available). The values in the “Mean occupation rate (total)” and “Mean occupation rate (partial)” refers to the four used vehicles. The partial occupation rate only considers the time and distance with passengers on-board (during the 5 hours and 30 minutes).
As expected the total occupation rate (including all the vehicle times and distances, with or without passengers on-board) is low. But it is also expected to achieve high standards by experimenting improved operating rules of the system (for instance, using smaller vehicles, adjusting supply to demand, would improve utilization rate of vehicles and therefore, the overall performance of the system). Also, it is expected that high occupancy rates will be higher when taking in account all DRT service (and not only this part where the population density is considerably lower than the overall density).
Both, vehicle movement and arrival frequency of the clients to a stop can be critical to define how the system must react in a real situation. In this illustrative example, from the total of 103 passengers, 82 (about 80%) arrived to the destination before the desired time (approximately 33 minutes per passenger) and 21 (about 20%) arrived after (approximately 35 minutes per passenger).
The “No-show” clients are also considered in the simulation. It is a parameter represented by a percentage of the potential customers’ population and therefore can be properly estimated along the real operating of the system (if it were implemented). For now, we can only perform some parameterization studies in order to analyze the impacts of different values on the overall performance solution. In addition, there are other parameters to take in account for future simulations, namely, the length of the time window, tolerance delays (clients and vehicles in the stop waiting), and the fleet size, among others.
There are issues concerning the costs of DRT system that require further analyses and still need to be addressed, in order to evaluate the transport system viability and sustainability. Some preliminary studies and correspondent results were reported by Oliveira *et al.* (2011a).

7. CONCLUSIONS AND FUTURE WORK

In order to achieve a successful DRT service it is important to develop a framework that provides different levels of decision-support to enable decision makers to perform systematic analysis leading to intelligent strategic solutions.
The IDSS proposed in this paper, along with the proposed general evaluator procedure, will ultimately assure the adoption of a sustainable DRT system, by properly adequate supply (e.g. fleet of vehicles and typology of services provided) to estimated demand levels and patterns, taking into account financial, economic and social decision criteria.
Since the outcome of the evaluation is highly dependent on the DRT specification (in terms of operational parameters, such as the level of spatial and temporal flexibility of their services), this framework must comprise an iterative approach that consists on defining an initial DRT specification, estimating their impacts in terms of performance indicators, redefining the specification and re-estimating the new impacts, and so on until a suitable solution is found, in terms of technical and economic viability and sustainability.
These results demonstrated that the system, based in an event-driven simulation approach allows calculating the necessary performance measures or PKIs to the assessment.
It was highlighted the importance of combining different technological and methodological means (web-portal, internet services, GIS, intelligent agents, advanced statistical, operational research and artificial intelligent tools), to enhance efficiency in transport operations, as well as, to enhance efficiency in the analyses and assessments. For example, graphical displaying devised solutions can allow further statistical analyses of spatio-temporal indicators; data mining techniques allow finding hidden patterns in demand that can lead to improved routing planning, etc..
The use of the simulator will ultimately allow identifying which are (probably) the best policies and strategies for the system to follow in the implementation phase, depending on the needs and characteristics of the area. Also through a currency demand it is possible to define the structure to analyze the system viability.
From the brief analysis and discussion based on the illustrative example of Section 6, it is suggested that simulation approach works. Currently, a large set of validation tests are taken place by using theoretical extreme cases data and real data (from the case study area).
Further validation tests are currently being done and some more will be performed, along with parameterization tests, for example estimating the effects (on the overall and nature-specific DRT performances) of parameters such as: spots of population concentration within the different counties of the study area; DRT system integration with regular transport service; flexibility of services as a function of economic efficiency, costs effectiveness and resources availability.
Additional studies will follow: IDSS overall validation, and proposal of a DRT specification to the study area (based on the basic principles of viability and sustainability).

ACKNOWLEDGEMENTS

The authors acknowledge the financial support provided by the European Union through FEDER – "Programa Operacional Factores de Competitividade" (POFC) and by the Portuguese Government through FCT – "Fundação para a Ciência e a Tecnologia". Research grants: FCT/TRA/72871-2006 and FCT/SEN-TRA/116216-2009.

REFERENCES

Ambrosino G., Nelson J., Romanazzo M. (Eds.), 2004, "Demand Responsive Transport Services: Towards the Flexible Mobility Agency" ENEA, Rome.

Battellino H., 2009, "Transport for the transport disadvantaged: A review of service delivery models in New South Wales", *Transport Policy*, 16, 123–129.

Brake J., Nelson J., 2007, "A case study of flexible solutions to transport demand in a deregulated environment", *Journal of Transport Geography*, 15, 262–273.

Brake J., Nelson J., Wright S., 2004, "Demand responsive transport: towards the emergence of a new market segment", *Journal of Transport Geography*, 12, 323-337.

Brake J., Mulley C., Nelson J., Wright S., 2007," Key lessons learned from recent experience with Flexible Transport Services", Transport Policy 14, 458–466.

Dias A., Carvalho M., Telhada J., 2011(a), "A decision support system for a flexible transport system", in book proceedings of 17th European Concurrent Engineering Conference, London, UK, 75-79.

Dias A., Carvalho M., Telhada J., 2011(b), "Economic evaluation of a demand responsive transport in rural area", in book proceedings of 1st International Conference on Project Economic Evaluation, Guimarães, Portugal, 269-277.

Enoch M., Ison S., Laws R., Zhang L., 2006, "Evaluation Study of Demand Responsive Transport Services in Wiltshire", Transport Studies Group Department of Civil, Building Engineering, Loughborough University, Leicestershire LE11 3TU, United Kingdom.

Fernández J., Cea J., Malbran R., 2008, "Demand responsive urban public transport system design: Methodology and application", Transportation Research Part A 42, 951–972.

Feuerstein E., Stougie L., 2001, "On-line single-server dial-a-ride problems", *Theoretical Computer Science* 268, 91–105.

Fu L., 2002, "A simulation model for evaluating advanced dial-a-ride paratransit systems", *Transportation Research Part A* 36, 291-307.

Giannopoulos G., 2004, "The application of information and communication Technologies in transport", *European Journal of Operational Research*, 152, 302-320.

INTERMODE Consortium, 2004, "INTERMODE: Innovations in Demand Responsive Transport", prepared for Department for Transport and Greater Manchester Passenger Transport Executive, London.

Li X., Quadrifoglio L., 2010, "Feeder transit services: Choosing between fixed and demand responsive policy", *Transportation Research Part C* 18, 770–780.

Liu S., Duffy A., Whitfield R., Boyle I., 2010, "Integration of decision support systems to improve decision support performance", *Knowledge Information Systems* 22, 261–286.

Lu Q. and Dessouky M., 2004, "An exact algorithm for the multiple vehicle pickup and delivery problem", *Transportation Science* 38 (4), 503-514.

Mageean J., and Nelson J., 2003, "The evaluation of demand responsive transport services in Europe", *Journal of Transport Geography* 11 (4), 255–270.

McDonagh J., 2006, "Transport policy instruments and transport-related social exclusion in rural Republic of Ireland", *Journal of Transport Geography*, 14, 355-366.

Mulley C., and Nelson J., 2009, "Flexible transport services: A new market opportunity for public transport", *Research in Transportation Economics*, 25, 39-45.

NCHRP, 2006. *Performance Measures and Targets for Transportation Asset Management.* USA. Transportation Research Board.

Nelson J., Phonphitakchai T., 2012, "An evaluation of the user characteristics of an open access DRT service", *Research in Transportation Economics* 34, 54-65.

Nelson J., Wright S., Masson B., Ambrosino G., Naniopoulos A., 2010, "Recent developments in Flexible Transport Services", *Research in Transportation Economics*, 29, 243-248.

Oliveira J., Afonso P., Telhada J., 2011, "An activity-based costing model for strategic decisions in transportation on demand projects*", in proceedings of the 1st International Conference on Project Economic Evaluation*, Guimarães, Portugal, 127-134.

Palmer K., Dessouky M., Zhou Z., 2008, "Factors influencing productivity and operating cost of demand responsive transit", *Transportation Research Part A* 42, 503–523

Parragh N., Doerner K., and Hartl R., 2008, "A survey on pickup and delivery problems, Part II: Transportation between pickup and delivery locations", *JfB*, 58, 81-117.

Quadrifoglio L., and Li X., 2009, "A methodology to derive the critical demand density for designing and operating feeder transit services", *Transportation Research Part B: Methodological*, 43, 922-935.

SAMPLUS, 2000, "A Basic System Architecture and Technical Solutions for DRT", WP4, Deliverable 4.1, version 4, In: ftp://ftp.cordis.europa.eu/pub/telematics/docs/tap_transport/samplus_d4.1.pdf. Accessed in: 28 July 2008.

Xiang Z., Chu C. and Chen H., 2006, "A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints", *European Journal of Operational Research*, 174, 1117- 1139.

Oliveira J., Telhada J. and Ferreira P. (2011a), "A cost-benefit analysis of an intelligent demand-responsive public transport system for elderly and disabled", in Proceedings of the 1st International Conference on Project Economic Evaluation (ICOPEV'2011), April 28-29, Guimarães, pp. 235-242.

Oliveira J., Afonso P. and Telhada J. (2011b), "An activity-based costing model for strategic decisions in transportation on demand projects", in Proceedings of the 1st International Conference on Project Economic Evaluation (ICOPEV'2011), April 28-29, Guimarães, Portugal, pp. 127-134.

Wilson N., Sussman J., Higonnet B., Goodman L., 1970, "Simulation of a computer-aided routing system (CARS)", *Highway Research Record* 318, 66–76.

AUTHORS BIOGRAPHY

Maria Sameiro Carvalho is graduated in Computer and Systems Engineering (University of Minho – Portugal, 1986) and holds a MSC degree in Transportation Planning and Engineering (University of Leeds - UK, 1990) and a PhD degree in Transport Planning and Engineering (University of Leeds - UK, 1996). Her main research interests are in Transportation, Supply Chain Management and Operational Research.

José Telhada holds a PhD degree in Transport Planning and Engineering (ITS-University of Leeds – UK, 2001). His main research interests are Operational Research and Logistics applied to transportation systems.

Ana Dias is graduated in Mathematics (University of Minho, 2004) and Master in Industrial Engineering (University do Minho, 2008). At the present she works as PhD student at University of Minho.

SUPPLY CHAIN SIMULATION

SUPPLY CHAIN REPLENISHMENT OPTIMIZATION UNDER UNCERTAIN DEMAND

S. T. Enns
Suvra Saha
Department of Mechanical and Manufacturing Engineering
University of Calgary
Calgary, Alberta, T2N 1N4 Canada
E-mail: enns@ucalgary.ca

KEYWORDS
Simulation-optimization, supply chain replenishment, reorder pont system, demand uncertainty

ABSTRACT

The problem addressed is that of finding optimal decision variables for a supply chain replenishment system when the future demand rates are uncertain but the probability distribution is known. A capacity-constrained supply chain with continuous-review reorder point replenishment for two product types is considered. Decision variables include reorder points, reorder quantities and a maximum backorder constraint. Optimization is based on maximizing profits, considering revenue along with holding, tardiness and lost sales costs. A key contribution is the development of a sampling methodology that allows this problem to be solved experimentally using simulation-optimization Results are compared when using the demand rate probability distributions versus using only the mean demand rates as inputs for optimization. Higher profits are observed when using the probability distributions. It is also shown that optimizing the shop load constraint has a large impact on profits.

INTRODUCTION

Supply chains are always managed under uncertainty. However, relatively little research considers how this uncertainty affects the selection of optimal decision variables. Ideally a practitioner would like to be able to model uncertainty and optimize decision variables based on this uncertainy. This research considers the optimal selection of replenishment and shop load decision variables under uncertain future demand rates.

The body of literature dealing with optimal replenishment under uncertain future demand is largely identified with the classical Newsboy problem as formulated by Magee (1956), although related work goes back even further. The objective is to order the number of newspapers for resale that will minimize the expected sum of overage and underage costs under uncertain demand, described by a discrete probability distribution. This is a non-capacitated, single-product, single-period problem with a single decision variable.

Many extensions have been considered, only a few of which are identified here. Karlin (1960) considered periodic replenishment in a multi-product, multi-period environment with time-varying demand distributions. Evans (1967) studied replenishment in a multi-product, multi-period environment with capacity constraints. Harrison and Van Mieghem (1999) studied replenishment in a multi-product environment with multiple capacity constraints. Further work on the single-period Newsboy problem is reviewed by Khouja (1999), while Matsuyama (2006) examines multi-period Newsboy problem generalizations. Most of this research has used analytical methods to come up with exact or heuristic solutions.

This paper seeks to provide new insights into supply chain optimization under the assumption that future demand rates can be described by known probability distributions. An economic measure of performance is used, with cost structures similar to those found in Newsboy problem research. The study seeks to extend the related research in several ways. First, a methodology is developed that allows simulation-optimization to be used for replenishment system decision variable optimization in place of analytical optimization methods or experimental methods, such as those used by Shang, *et al*. (2004). The advantage of using simulation-optimization is that more complex and realistic supply chain scenarios can be studied. Second, this research incorporates the optimization of shop load control in capacity-constrained supply chains. Shop load control is an important factor that appears not to have been considered in previous optimization research. Third, this study quantifies the difference in performance due to optimization based on future probability distributions versus the mean, or expected, rates of demand.

SCENARIO AND METHODOLOGY

This section first describes the specific research problem. The experimental supply chain scenario, probabilistic demand rate assumptions, cost structures and performance measures are then presented. Finally, the simulation-optimization methodology and data collection procedures that allow sampling under demand rate uncertainty are described.

Problem description

A small supply chain model is contrived to investigate the setting of replenishment decision variables in supply chains. A key assumption is that the future customer demand rates at a warehouse are uncertain and that management must decide on setting decision variables for a span of time into the future, during which time the true demand rate may not be immediately apparent. Replenishment of inventory to the warehouse is from a capacity-constrained manufacturing system, controlled through a continuous-review reorder point system. This is a make-to-stock environment and the the decision variables are the reorder quantity and reorder point for each product type. Reorder quantities, or lot sizes, that reduce flow times and keep shop inventories low are desirable. Reorder points must provide satisfactory customer service levels. As well, there is a decision variable that allows customer requests to be rejected when manufacturing shop loads become excessive.

The objective is to maximize the expected profit given uncertain demand rates for the next period. Costs considered include inventory holding, tardy customer delivery and lost sales costs. It will be recognized that the decision variable interactions and cost tradeoffs makes this a difficult problem to optimize.

The assumption is made that the future demand rate probability distributions are discrete and known. These probability distributions can therefore be taken into account when choosing decision variables to maximize profits. Alternatively, only the means of the demand rate distributions can be used in optimizing the decision variables. Obviously it is easier to pursue optimization using only the means of the demand rate distributions since these are single values. A key question is whether using the probability distributions as opposed to the means during decision variable optimization leads to higher average profits.

Experimental supply chain scenario

A simple, capacity-constrained supply chain consisting of customer demand, a warehouse for finished goods inventory and a manufacturing unit was modeled. Two types of product are assumed to be supplied. Customer arrivals to the warehouse are dictated by a Poisson process, with each customer requesting only one item of a product. Replenishment to the warehouse is based on a continuous-review reorder point system. The manufacturer performs the necessary processing and then ships the lot-size order of the product to the warehouse. Replenishment of raw materials to the manufacturer is instantaneous since the supplier is assumed to have infinite stock. Figure 1 illustrates the scenario and replenishment logic.

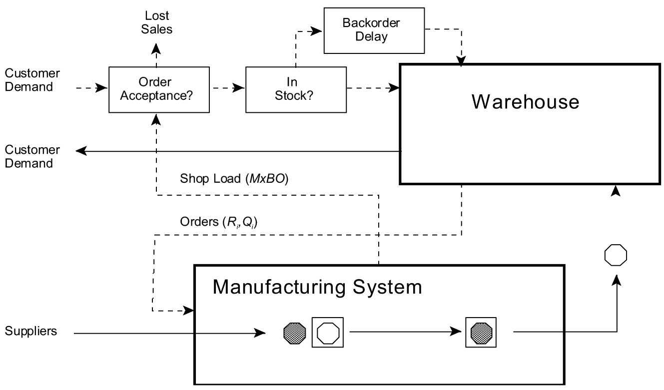


Figure 1: Supply Chain Scenario

When a customer arrives and there is inventory in the warehouse, a single item of the requested product type is removed. If a customer arrives and there is no stock, one of two things can happen. First, the requested item is backordered and customer must wait until inventory becomes available, at which time the request is immediately filled. This incurs a tardiness cost that is proportional to the time the customer has had to wait. Alternatively, the customer is notified that the requested item cannot be supplied and the sale is lost. This latter situation only occurs when the backorders have already reached a threshold level. This maximum backorder threshold is determined by a decision variable called *MxBO*.

The inventory position for each product type *i* is monitored independently. This inventory position includes the warehouse inventory plus orders in process minus backorders. Replenishment is controlled by two decision variables for each product type; the reorder point and the reorder quantity. Once the inventory position falls to the reorder point, R_i, an order is placed for quantity, Q_i. There are no order transmission delays so arrival at the manufacturer is instantaneous.

The manufacturing unit has two resources. The orders are processed through these resources sequentially on a first-come-first-serve (FCFS) basis, with the production lot size being equal to the order quantity. There is a setup time associated with the production of each lot on each resource. As well there is a processing time per item. The lot processing time is equal to the item processing time multiplied by the lot (or order) size. The assumed setup and processing times are stochastic, with the distributions shown in Table 1.

Table 1: Manufacturing Time Distributions

		Lot Setup time	Item Proc. Time
Product 1	Resource 1	Triangular (0.4,0.5,0.6)	Uniform (0.04,0.13)
	Resource 2	Triangular (0.3,0.6,0.6)	Uniform (0.08,0.10)
Product 2	Resource 1	Triangular (0.3,0.5,0.6)	Uniform (0.06,0.17)
	Resource 2	Triangular (0.3,0.3,0.6)	Uniform (0.07,0.14)

Once the processing of an order at the manufacturer is complete, it is transported to the warehouse. The transport delay follows a triangular distribution with parameters (0.5, 1, 1.5). Inventory in transport is considered to be working-in-process inventory, becoming finished goods inventory only at the time of receipt at the warehouse. All units of time can be assumed to be in hours.

Experimental demand scenario

The behavior of the system was explored using both deterministic and probabilistic demand rate assumptions. In the deterministic case, shown at the top of Table 2, the demand rates for product types 1 and 2 were assumed to be 4 and 3 respectively. Since the total item demand rate per hour is 7 and demand is deterministic, this demand level is referred to as 7D.

Probabilistic demand rate distributions are designated as PD. Under probabilistic assumptions the demand rate observed in the next future period could take on one of three discrete levels. The probability distribution was assumed to be symmetrical so that the mean of the distribution is also the medium demand level. The demand levels and probabilities associated with the probabilistic demand rates are shown in the lower part of Table 2. All rates are assumed to be in items per hour.

Table 2: Demand Rate Probability Distribution

Demand Assumptions	Demand Level	Probability of Level	Demand Rate	
			Prod. type 1	Prod. type 2
Deterministic Rate - 7D	7D	1.00	4	3
Probabilistic Rate – PD	5D	0.25	3	2
	7D	0.50	4	3
	9D	0.25	5	4

Profit and cost structures

This research is based on the use of economic measures, with profit determined by throughput, profit margins and relevant costs. Profit margins were defined to be equal to revenue minus the cost-of-goods sold per item. The cost-of-goods sold includes things like material, labor and fixed overhead inputs. It does not include costs related to holding inventory or satisfying customer demand. The profit margins, pm_i, were assumed to be \$44 and \$35 per item for product type 1 and 2 respectively.

The relevant costs included inventory holding, delivery tardiness and lost sales costs. Holding costs included work-in-process and finished goods inventory. The work-in-process holding cost, *wc*, was assumed to be \$1.50 per item per hour for both product types. The finished goods holding cost, *fc*, was \$3.00 per item per hour for both product types.

Tardiness costs, *tc*, were incurred when a customer request was not filled from stock and had to be backordered. Tardiness is the amount of time a customer spends waiting for a backordered item. If the customer order is filled from stock, the tardiness is zero. Mean tardiness is the average tardiness across all orders filled, either immediately or after being backordered. The tardiness cost was assumed to be \$10 per item per hour for both product types.

Lost sales costs represent the profit foregone if a customer request is rejected due to an overloaded production system. The lost sales cost, lc_i, was assumed to be the same as the profit margin for each product type. In other words, it was equal to \$44 and \$35 per item for product type 1 and 2 respectively.

Profit maximization measures

The objective was to maximize the total profits, *P*, as shown in the following equation. Profit is a function of the replenishment system decision variables for each product type, R_i and Q_i, and the maximum backorder threshold, *MxBO*. The equation given is the potential revenue minus the finished goods inventory holding cost, work-in-process holding cost, tardiness cost and lost sales costs.

$$Max\ P((R_i, Q_i, MxBO) = \sum_{i=1}^{2} pm_i\, D_i - \left(\sum_{i=1}^{2} fc\,(FG_i) + \sum_{i=1}^{2} wc(WIP_i) + \sum_{i=1}^{2} tc\,(MT_i)(D_i - L_i) + \sum_{i=1}^{2} lc_i\,(L_i)\right)$$

i - index for product type i
P - total profit per unit time
D_i - demand rate per unit time
FG_i - time-averaged finished inventory
WIP_i - time-averaged work-in-process inventory
MT_i - mean tardiness across all completions, in units of time
L_i - lost sales per unit time
pm_i - profit margin per item (rev. minus cost-of-goods sold)
fc - finished goods holding cost per item per unit time
wc - work-in-process holding cost per item per unit time
tc - tardiness cost per item per unit time
lc_i - lost sales cost per item

Maximizing this objective function will lead to lot sizes, Q_i, that tend to minimize flowtimes. Flowtimes are known to be a convex function of lot sizes in a capacity-constrained system with setup times. Reducing flowtimes leads to reduced holding costs. At the same time, reorder points, R_i, will be found to balance the tradeoff between finished goods inventory holding costs and delivery tardiness costs. The biggest effect of the *MxBO* decision variable is on lost sales costs. However it also impacts both holding and tardiness costs since it controls maximum throughput.

Simulation and optimization methodology

The experimental scenario was modeled using *Arena*® 10 discrete-event simulation software (Kelton, *et al*., 2007). The simulation model was linked into *OptQuest*®, an optimization search engine that facilitates finding optimal solutions experimentally. The simulation component provides performance outputs for a given set of input decision variables. The optimization component uses the outputs from previous simulation runs to determine a new set of simulation input decision variables. This iterative simulation and optimization process will search for decision variables that maximize or minimize the objective function. The procedure *OptQuest*® uses relies on a combination of scatter search, tabu search and neural networks. Further details on the simulation-optimization procedure can be found in Grewal, *et al*. (2010).

A key issue in this research is that optimization must be based on sampling from a probability distribution. Therefore the usual simulation-optimization procedures cannot be applied and a new procedure had to be developed. Instead of optimizing over a continuous simulation run, it is necessary to optimize over a number of time periods, not all of which have the same demand rates. In other words, like the Newsboy problem, the objective is to find optimal decision variables given an uncertain demand rate for a single future period.

Figure 2 illustrates the approach taken during part of one simulation run. The current state of demand rate is assumed to be 7 items per time unit and the objective is to determine the best decision variables for one future period (year), given that the demand rate could be 5, 7 or 9 items per time unit. The first three periods of the simulation run represent a warm-up stage. This is relatively long, given that the simulation model is started empty. Following this, data collection takes place every second period, with re-initialization taking place in the periods between. In other words, the simulation is set back to the current real demand rate of 7 items per time unit. The period after each reinitialization, represents a possible demand rate level for the the future time period of interest. The number of times each demand rate level occurs during the simulation run is controlled by the probability distribution.

In this research it was assumed the time period was a year and that there were 8760 hours per year. The simulation run length was set to 35 years, with a warm-up of 3 years. Re-initialization between each data collection year was also a year. Therefore data was collected over 16 one year periods in a single simulation run. Using the combined demand rate notation for both product types and the probability distributions in Table 2, it is apparent that 4 of these years will be run at the 5D demand rate level, 8 at the 7D demand rate level and 4 at the 9D demand rate level.

Each simulation-optimization experiment required many simulation runs to find the optimal decision variables. Typically 300 to 400 combinations of decision variables, each run for three replications, were evaluated before terminating the experiment due to a lack of further improvement in the profit objective. A single simulation-optimization experiment took in the neighbourhood of 20 hours to perform. The best decision variables at this point were chosen to be optimal. Although there is no guarantee of optimality, these decision variables were evaluated to ensure they were at least very close to maximizing profit.

Once the optimal decision variables were determined, simulation alone was used to generate the experimental results used for analysis. The optimal decision variables were used in 40 replications at this stage. Common random numbers were used as a variance reduction technique across different experimental factor settings. All other data collection procedures were the same as those used in the simulation-optimization experiments.

The requirement is for *OptQuest®* to find one set of decision variables that will optimize performance over all the data collection periods. In other words, it must find the decision variables to maximize profit given the demand rate uncertainties. To obtain proper measures, special variables were used in the *Arena®* simulation model to turn Counters and Tallies for the performance outputs on or off, depending on whether or not the current simulation time was within a data collection period. The collection of time-persistent statistics, such as machine utilizations, also required correction for the experimental data collection time versus the simulation data collection time.

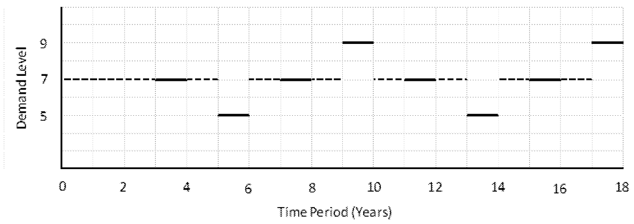


Figure 2: Simulation Run Data Collection Pattern

RESULTS UNDER UNCERTAIN DEMAND

This section examines the experimental results obtained using the probability distribution of demand rates versus the mean demand rates for optimization.

Experimental design for optimization under uncertainty

An experiment was performed using two factors. These were the actual demand rate assumption, identified by 'D', and the decision variable optimizing demand rate assumption, identified by 'V'. The actual demand rates were assumed to be either deterministic (7D) or probabilistic (PD), as shown in Table 2. The deterministic demand rate, 7D, is equal to the mean of the probability distribution, PD.

The demand rate assumptions used in optimization need not necessarily be consistent with the actual demand rate assumptions used in experimentation. The simulation-optimization experiments were based on assuming either the deterministic demand rate (7V) or the probabilistic demand rate distribution (PV). In other words, 7V and PV represent the way the decision variables were obtained, not the way demand was modeled during data collection.

Experiments involving all combinations of these factor settings are shown in Table 3. Data collection in all cases was as illustrated in Figure 2, the only difference being that under a deterministic demand rate all data collection periods were at the same demand rate level.

The shaded cells in Table 3 indicate the combinations where the actual demand rate and optimization demand rate assumptions were consistent. Experiment 7D_7V indicates the demand rates were deterministic and that this assumption was also used in optimization of the decision variables. Experiment PD_PV indicates the demand rates were probabilistic and that this assumption was also used in optimization. Experiment 7D_PV indicates that the actual demand rates assumed were deterministic but that probabililistic demand rates were assumed during optimization. Experiment PD_7V indicates the demand rates the actual demand rates were assumed to be probabilistic but that deterministic demand rates were assumed the decision variables were assumed during optimization.

Table 3: Demand Rate and Optimization Settings

		Demand Rate Assumptions	
		Deterministic - 7D	Probabilistic - PD
Decision Variable Settings	7V	7D_7V	PD_7V
	PV	7D_PV	PD_PV

If in reality the true demand rates in the future period are probabilistic, the two experimental combinations of most relevance are PD_7V and PD_PV. Comparing results from these two experimental combinations should indicate whether it is beneficial to take uncertainty into account during optimization or whether assuming a mean demand rate is satisfactory. Note that optimization based on deterministic demand rate assumptions is much easier.

Experimental results for optimization under uncertainty

The shaded columns in Table 4 show the optimal decision variables obtained from simulation-optimization experiments using either the deterministic or probabilistic demand rate assumptions consistently. The other two set of experiments, PD_7V and 7V_PD, use the same sets of decision variables as experiments 7D_7V and PD_PV respectively.

Table 4: Decision Variable Settings

Experiment	7D_7V		PD_PV		PD_7V		7V_PD	
Product Type	1	2	1	2	1	2	1	2
Reorder Point, *R*	17	13	15	10	17	13	15	10
Reorder Quantity, *Q*	12	10	11	10	12	10	11	10
Max. BO, *MxBO*	5		4		5		4	

These decision variables were used in simulation experiments, as outlined in the previous subsection, to obtain the results in Table 5. Profit rates are also shown plotted in Figure 3. In all cases the results presented are the averages over 40 replications.

Comparing results under probabilistic actual demand rates shows that expected profit is \$160.07/hr when using probabilistic demand rate assumptions for optimization, versus \$157.44/hr when using deterministic demand rate assumptions for optimization. These values are shown shaded in Table 5. Probabilistic demand rate assumptions during optimization result in higher lost sales costs but lower inventory holding costs. Shop utilization, not shown, is also slightly lower.

Under probabilistic future demand rates the shop load constraint, *MxBO*, is very important. If the actual demand rate observed happens to be high, many customer requests will result in lost sales but this will keep shop loads feasible. If the actual demand rate observed happens to be low, the shop load constraint will have little effect. To verify this behavior the two experimental combinations under probabilistic actual demand rate assumptions were rerun but with the *MxBO* constraint relaxed and set at 30. The profits were $23.42/hr and $57.62/hr for PD_PV and PD_7V respectively. These results show that relaxing the shop load constraint causes the profit to drop dramatically. The main reason is that fewer customer requests are rejected so shop loads and flowtimes increase. This causes backorders and tardiness costs to explode.

Table 5: Results under Various Assumptions

Experiment	7D_7V	PD_PV	PD_7V	7D_PV
Profit Measures ($/hr.)				
Potential Revenue	271.99	271.99	271.99	271.99
Lost Sales Cost	9.35	34.30	25.56	21.55
WIP Holding Cost	52.24	43.20	48.84	45.649
FG Holding Cost	23.49	25.24	30.34	20.40
Tardiness Cost	8.11	9.19	9.81	9.24
Total Cost	93.19	111.92	114.54	96.84
Total Profit	178.80	160.07	157.44	175.16
Inventory Measures				
WIP Count	34.83	28.80	32.56	30.43
FG Count	7.83	8.41	10.11	6.80
BO Count	0.81	0.92	0.98	0.92
Total Count	43.47	38.13	43.65	38.15
Delivery Performance				
% Lost Customers	3.4%	12.6%	9.4%	7.9%
% Backorder Del.	18.9%	17.1%	17.9%	19.9%
% In-Stock Del.	77.7%	70.3%	72.7%	72.2%

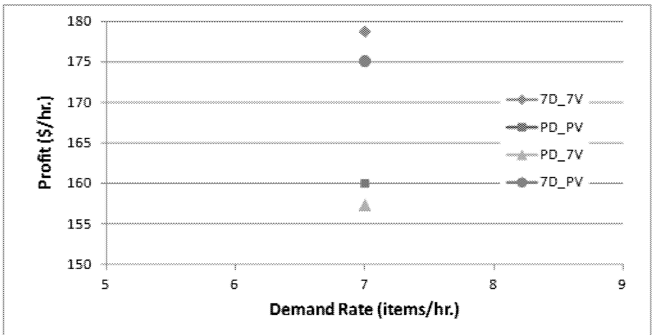


Figure 3: Profit under Various Assumptions

CONCLUSIONS

This research has demonstrated the use of simulation-optimization for sampling when uncertainty can be described by a discrete probability distribution. Experimental results show that it is advantageous to optimize replenishment system decision variables on the basis of the probability distributions versus the means of future demand rates. However, it is also shown that the optimization of shop load control policies is very important.

The procedures illustrated can be used with any discrete probability distribution. A practical limitation is that in reality the probability distribution would not be known and would also need to be estimated. As well, it could be argued that a discrete distribution, such as illustrated, is at best an approximiation for the many different true demand rates that could be observed. Another limitation is that simulation-optimization of larger problems may be impractical due to the time required to find the optimal decision variables.

Future research could take several directions. One would be to compare different types of replenishment systems using a similar methodology. Another direction would be to explore alternative shop control mechanisms using a similar methodology for optimization and comparison. A third would be to explore the effect of the probability distribution standard deviation on relative performance differences.

REFERENCES

Evans, R.. 1967. "Inventory control of a multiproduct system with a limited production resource", *Naval Research Logistics Quarterly*, 14: 173-184.

Grewal, C.; P. Rogers; and S.T. Enns. 2010. "Performance evaluation of inventory replenishment strategies in a capacitated supply chain under optimal parameter settings." *International Journal of Value Chain Management*, 4(3): 195-212.

Harrison, J.M. and J.A. Van Mieghem. 1999. "Multi-resource investment strategies: operational hedging under demand uncertainty." *European Journal of Operational Research*, 113: 17–29.

Karlin, S., 1960. "Optimal policy for dynamic inventory process with stochastic demands subject to seasonal variations." *Journal of the Society of Industrial and Applied Mathematics (SIAM)*, 8: 611-629.

Kelton, W.D.; R.P. Sadowski; and D.T. Sturrock. 2007. *Simulation with Arena*. 4th edition, Mc Graw-Hill, New York.

Khouja, M., 1999, "The single-period (news-vendor) problem: literature review and suggestions for future research." *Omega: The International Journal of Management Science*, 27: 537-553.

Magee, J.F. 1956. "Guides to Inventory Policy", *Harvard Business Review*, 34(3): 57-71.

Matsuyama, K., 2006. "The multi-period newsboy problem." *European Journal of Operational Research*, 171: 170-188.

Shang, J.S.; S. Li; and P. Tadikamalla. 2004, "Operational design of a supply chain system using Taguchi method, response surface methodology, simulation, and optimization." *International Journal of Production Research*, 42(18): 3823-3849.

AUTHOR BIOGRAPHIES

SILVANUS T. ENNS is an Associate Professor in the Dept. of Mech. and Mfg. Eng. at the U. of Calgary.

SUVRA SAHA recently obtained an MSc and currently works on supply chain problems at Suncor Energy.

LOGISTIC CHAIN OPTIMIZATION FOR IMPORTING COAL ON INLAND WATERWAYS IN EASTERN INDIA

T. van de Sande, J.A. Ottjes, G. Lodewijks,
Faculty of Mechanical, Maritime and Materials Engineering
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
E-mail: g.lodewijks@tudelft.nl

KEYWORDS
Discrete event simulation, logistics, chain optimization.

ABSTRACT

The possibilities to start a logistics chain to import coal on inland waterways in India is researched. Direct floating transshipment is the best modality due to lack of suited harbours in the area. The storage on the floating terminal decouples the hinterland logistics chain with the supplying deep-sea vessels. After studying seven different configurations it turned out that the simplest and the most efficient configurations are most promising. A simulation model is used to test the variation on the output when stochastics are introduced. The results from the simulations show that the inland waters of India are very suited for coal import because it's cheaper and more realiable than transport by rail.

BACKGROUND

Due to the growth of the Indian economy the demand for energy is rising. Therefore the demand for coal is also increasing. The Indian energy supply however is unreliable due to unreliable supply of coal on the clogged Indian railway tracks (Kumar, 2009). To create a trustworthy supply chain of coal from abroad to the power plants in the inland of India, the Inland Waterway Authority India (IWAI) has started with a pilot project on the Hugli River. This pilot project explores the possibilities to transport 3 million ton of coal annually over the inland waterways of India. From the Bay of Bengal in the south to a power plant in Farakka, 600 km northwards. Because the Indian government and the Indian commercial parties have no experience with inland waterway transport the Dutch government provides assistance. On behalf of the Dutch government, Royal Haskoning and Marin advice the Indian parties about the logistics chain, environmental aspects and financial impact. The IWAI stated the following requirements:

- Higher productivity than rail: hence, lower handling cost than railway transport: under €7 per handled ton.
- Throughput: 3 MTPA (million ton per annum) with possibilities for growth in the future to 10 MTPA.

This paper will discuss the best option for the logistic chain optimization for importing coal on inland waterways in Eastern Inda using simulation.

ENVIRONMENTAL CONDITIONS

The final logistic chain option must be adopted to the current environmental situation in Eastern India, als see figure 1. Therefore the local conditions are given in the graphical outline of the geographic situation of the two rivers: the Hugli River and the Bhagirathi River.

Sagar Island And Further South

The water depth around Sagar Island is fluctuating due to the tide. There are many small tidal dry ports around Sagar Island. South of Sagar Island two channels are guiding the incoming vessels from the anchorage area at Sandheads (not depicted in Figure 1) to the west side of the Sagar Island into the river Hugli. Vessels that want to sail south of Sagar Island must be suited for higher waves and stronger currents, hence must have higher freeboard and more engine power. Sailing east of Sagar Island is impossible due to limited draft.

Haldia Docks

These docks contain a large harbour, however with restricted depth and situated behind locks. The depth is insufficient to berth Panamax vessels. Therefore Haldia Dock is not considered a suitable location for this pilot project.

Diamond Harbour

At this location no real harbour is situated, but rather a small town situated at a wide part of the river (over 2000m). Here floating transshipment is possible.

Tribeni

At the small town of Tribeni the river Bhagirathi flows into the river Hugli. The depth of the Hugli in the south is around 6-8m but the depth of the Bhagirathi in the north is restricted to 2.5m. Therefore Tribeni seems a key location to change from vessels with large draft to vessels with a very limited draft.

Baghirathi River

The largest part of the distance is covered on this river. There are some very tight bends that restrict the use of very lengthy vessel combinations. Exploration of the river by Marin has shown that the year round minimum depth is 2.5m between Farakka and Tribeni. Most parts of the Baghirathi river are deeper, but some small bumps in de river-bed reduce the allowable draft of the complete river. With a minimum required clearance under the keel of 0.2m it is strongly recommended to use only equipment with a maximum draft of 2.3m at the part between Tribeni and Farakka.

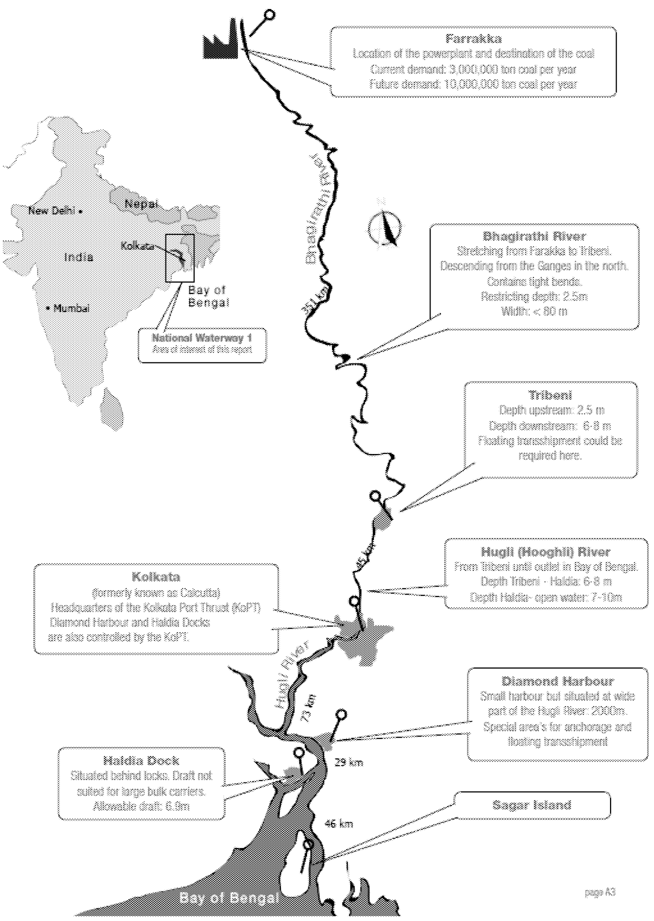


Figure 1. Eastern India waterways

Farakka

At this little town, 544 km northward of Sagar Island, the NTPC Power plant is located. The banks of this channel are undeveloped and require some modifications to be able to unload barges with coal.

METHOD OF SUPPLYING COAL

The whole logistics chain start at the delivery of the coal from the mines in Australia or Indonesia. The transportation of the coal is out of the scope of this article, however the method of coal supply is of influence on the complete logistics chain. The coal can be supplied by small bulk carriers, type Handymax, delivering loads between 35,000-45,000 ton per call. Or the coal can be supplied by larger bulk carriers, type Panamax (55,000-80,000 ton). Both types have their own advantages and disadvantages. Based on the economy of scale the coal from the Panamax vessels is cheaper per ton. Therefore selection of the largest bulk carriers seems to be the best solution, however there are some serious disadvantages when using large bulk carriers:

- Sufficient draft to berth the deep-sea vessel. Extensive surveys of the IWAI (Ghosh, 2011) showed that allowable draft around Diamond Harbour is only 7 meters, while fullu loaded Panamax vessel require at least 14m of draft. Handymax vessels have a draft of 7 meters and could reach Diamond Harbour fully loaded. This is confirmed by monitoring all large vessels (40,000 ton and larger) that entered the Haldia Docks in April 2011 with Sea-web (Sea-web, 2011). Average vessel draft was 6.9m. If vessels with more draft than 7.0m want to unload it have to be done at Sandheads or Kanika Sands. Situated at 145km and 220km south of Diamond Harbour respectively. These lightering area's are situated at open sea and indicated by the IWAI. Here is sufficient depth to unload all types of bulk carriers.
- High peak loads on the logistics chain. Because the operational cost of the larger bulk carriers are higher the time to unload the Panamax vessels is shorter. The allowable unload time for a Panamax vessels is set at 48 hours. (Smits, 2010). This required high unloading capacities at the unloading terminals. This caused high peak loads when the bulk carriers are berthed. The whole inland logistics chain must be tuned to be able to deal with this peak load when the supplier and the connection to the hinterland transport are coupled. This means many inland vessels to transport the large amount of coal to the power plant in Farakka. The hinterland logistics chain can be decoupled from the suppling bulk carriers by installing intermediate storage to level out the peaks on the supply chain.
- Another consequence of the large bulk carriers with high daily operational cost is the demurrage cost. When the allowed time isn't obtained, high penalty cost have to be paid to the stevedore: the demurrage cost.

The research will show whether the disadvantage of draft, peak loads and demurrage can outweigh the advantage of the cheaper delivery of coal for large bulk carriers.

FLOATING TRANSSHIPMENT METHODS

Due to the lack of suited harbours in the area floating transshipment is the only option to unload the bulk carriers. There are three variations of floating transshipment: direct, moving or indirect, also see the figures 2 and 3:

1. Direct floating transshipment: using only floating cranes to transship the coal directly from the hold of the bulk carrier into the waiting smaller vessel. Cannot continue when no small vessel are present.
2. Moving floating transshipment: a floating terminal with internal storage unloads the bulk carriers until both have sufficient draft to sail to shallower waters. There the bulk carrier is emptied into inland vessels. When the bulk carrier is empty, the floating terminal unload it's own stored coal in inland vessels.
3. Indirect floating transshipment: the bulk carrier is completely emptied into the storage of the floating terminal or into small vessels. The presence of storage enables continuous unloading of the bulk carrier.

After an Multi Criteria Analysis based on the Analytic Hierarchy Process (Lootsma. 1978) the best suited option was clearly the indirect floating transshipment. Most important reason is the decoupling of the hinterland logistics chain. Which allows minimal inland vessel equipment. This is advantageous because extra investments in larger floating storage is financially more attractive than obtaining more sailing inland equipment.

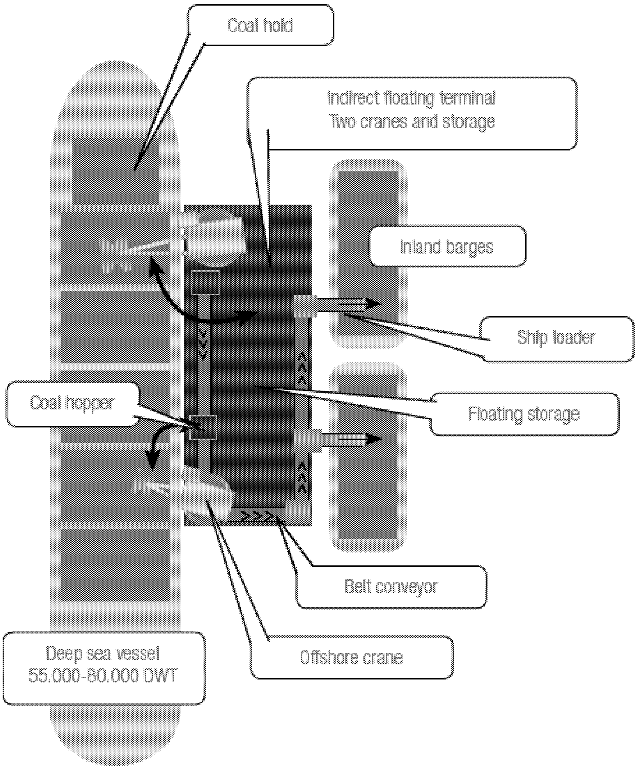


Figure 2. Floating transshipment principles

Figure 3. Floating transshipment at sea

INLAND VESSELS

Many sort of vessels are able to transport the coal from the floating terminal over the inland waters to Farakka. However, which type of vessels is best depends on the local conditions. For sailing on open water more freeboard and more engine power are required. For sailing on the Bhagirathi river a maximum draft of 2.3 is allowed. The economy of scale also holds here: larger vessels means fewer equipment, less kilometres sailed and fewer staffing cost. However the largest vessels will only be financially attractive if the full draft of the vessels can be exploited. Especially because approximately 50% of vessel costs are fuel expenses. Thus a small fuel efficient engine is preferable. Eight types of vessels are studied and the cost per ton per kilometre was calculated. For transportation on open water the Large Estuarine vessel is cheapest. With cargo size of 9280 ton, length of 135m and 3480kW of engine power this is efficient method to transport coal over open water. However the draft of 4.4 meter require transshipment at Tribeni. For shallow inland water the Class IV vessel is best. With a length of 86m, 1580 DWT and 800 kW this option is cheapest on rivers with only 2.3 meter draft. When both open waters and shallow inland waters must be covered the small estuarine vessel (3030 DWT, 2200kW and 134m) is preferred, als see figure 4.

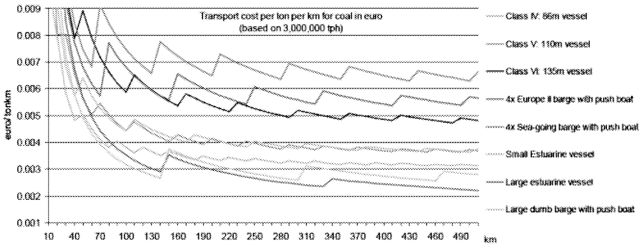


Figure 4. Transport cost per ton per km for different options

STATIC ANALYSES OF SEVEN CONFIGURATIONS

The best performing option can be found from the analyses of all the various options of floating transshipment, coal suppliers and vessel types. However seven configuration are created which cover the most important parts of the complete scope. The seven configurations are based on three main aspects:

- Efficiency: the economy of scale applies to the barges. High efficiency, thus fully loaded vessels with appropriate engine power, results in financial attractive solutions
- Flexibility: the flexibility of the logistics chain depends on the selected barges; dedicated barges which are suited for one specific part of the stretch shall realize low cost but are inflexible, while the sea-going push barges are highly flexible but at higher cost.
- Controllability: increasing number of links reduces the control over the complete link. More links mean more floating terminals and different vessels, but better suited equipment for the specific locations.

The seven configurations are:

1. Supply by Handymax in Diamond Harbour. Indirect transshipment into Class IV vessels to Farakka
2. Supply by Handymax in Diamond Harbour. Indirect transshipment into 10,000 ton push barges to transport to Tribeni. From there in Class IV vessels to Farakka.
3. Moving floating transshipment with Panamax in Sandheads and Diamond Harbour. From there with Class IV vessels directly to Farakka.
4. Moving floating transshipment with Panamax in Sandheads and Diamond Harbour. From there with 10,000 ton push barges to Tribeni. Transshipment into Class IV vessels to Farakka.
5. From Panamax at Sandheads into single sea-going push barges to Diamond Harbour. Convoying into sets of four. The 4-convoys goes straight to Farakka.

6. Supply by Panamax at Sandheads into Large estuarine vessels until Tribeni. From there to Farakka with Class IV vessels.
7. From Panamax at Sandheads transshipped into small estuarine vessels. Directly to Farakka.

The results after analysing these configurations are clear:

- The additional cost for supplying by Handymax are too expensive. #1 and #2 are not affordable.
- Moving floating transshipment is unsuited due to time consuming moving of terminals. #3 and #4 are practically impossible.
- The use of heavy sea-going barges is a financial burden. High fuel cost make #5 very expensive.
- #6 and #7 look promising for further research and fulfil all stated requirements. Achieved productivity was €5.66 and €6.03 for #6 and #7 respectively.

Based on these results a simulation model has been built to test the variation of the output of configuration 6 and 7 when stochastic parameters are introduced in the input.

SIMULATION DESIGN

The simulation model, based on [Veeke & Ottjes. 2000] was built with the following stochastic distributions for input:

- Normal distribution on the inter arrival times of bulk carriers, with a standard deviation of 24 hours.
- Number of holds in bulk carriers. Uniform distributed.
- Velocity of inland vessels: Uniform distributed with a two sided variation of 10% of nominal velocity.

Configuration 6 was performed 80 times because different quantities of barges and different sizes of storage are tested. Configuration 7 was performed 55 times.

If was verified that the simulation run length of 6 years was sufficient because the system output is stable. Due to startup effects the first year of data is removed from the analysis, thus 5 years remain. Only 1 run is performed per set of parameters.

The output of the simulation runs was evaluated on the following criteria:

- Productivity: the handling cost per ton can be at most €7 per ton. More expensive runs are not suited
- Effectivity: the configuration must be able to unload the bulk carriers in time. At least 75% must be serviced in 48 hours and 99% in 96 hours.
- Efficiency: the cranes and barges must be used to achieve high efficiency. At least 80% is required
- Control: When the storage of the offshore floating terminal is often fully loaded the control over the logistics chain is insufficient. The limit for a full storage is set at once every month.

SIMULATION RESULTS

After executing all 135 runs, also see Figure 4, the best performing configurations are:

For configuration #6: Run 54:

- 1 offshore floating terminal with 100,000 ton storage and 3 Liebherr CBG300 cranes
- 3 large estuarine vessels
- 1 inland floating terminal at Tribeni with 20,000 ton of storage and 1 Liebherr CBG350 crane
- 14 Class IV vessels
- 1 land terminal at Farakka with 1 Liebherr CBG300 crane.

This results in the required output: handling cost are €5.95 per ton. Throughput times are within requirements: 75% leaves within 46 hours and 99% within 86 hours.

For configuration #7: Run 37:

- 1 offshore floating terminal with 100,000 ton storage and 3 Liebherr CBG300 cranes
- 13 small estuarine vessels
- 1 land terminal at Farakka with 1 Liebherr CBG300 crane.

This results in the required output: handling cost are €6.10 per ton. Throughput times are within requirements: 75% leaves within 44 hours and 99% within 51 hours.

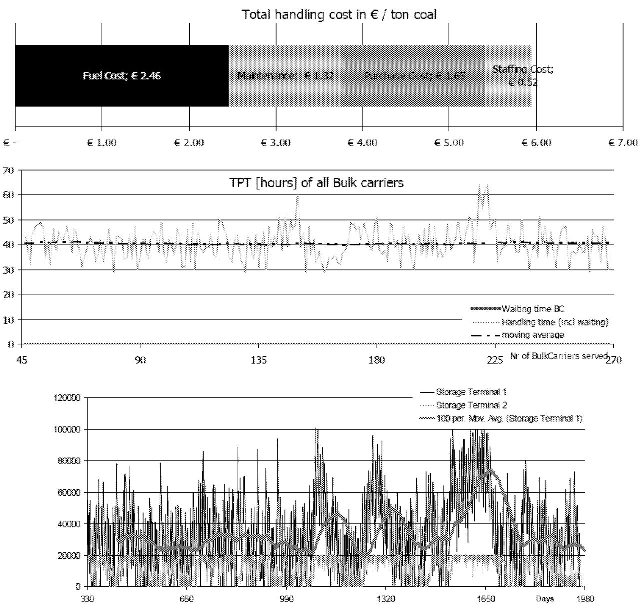


Figure 4. Example of the simulation output

It can be concluded that run 54 of configuration #6 is the best overall performing configuration with a productivity of €5.95 per ton. This is €1.05 less that the maximum allowed handling cost. Also all stated requirements are fulfilled and therefore can be concluded that the problem statement is solved. The IWAI can start implementing the proposed solution. Importing coal over the inland waterways in Eastern India is financially more attractive than transportation over railroad. Besides this inland waterways provide a far more reliable logistics chain. In the three displayed figures above the handling cost (divided in 4 main parts), the throughput time of the bulk carriers, and the storage level of terminal 1 (at Sandheads) and terminal 2 (at

Tribeni) are given. As can be seen almost half of the cost are spent on fuel and only 8% on staffing. The throughput time is fairly constant, but fluctuating due to varying bulk carrier size and varying arrival times. Outliers do happen but can be corrected by the modelled system. The storage of Terminal 1 is fully loaded at occasions but on average fewer than once a month (35 times in 5 year).

GROWING TOWARDS THE FUTURE

Extra simulations were performed to prove that the selected configuration can be up scaled until a throughput of 4.75 MPTA. With a minor adjustment the capacity can even be increased up to 5 MTPA. Due to higher equipment efficiency the productivity increases to €5.08 per handled ton. When the future demand of 10 MPTA must be achieved the complete chain can be duplicated.

CONCLUSION AND RECOMMENDATIONS

It can be concluded that it is possible to transport 3 MTPA from bulk carriers to the power plant in Farakka for less than €7 per ton. With the selected configuration a productivity of €5.95 is possible. All requirements on throughput time are satisfied when using the proposed solution. The required growth of 10 MTPA can also be achieved. It is recommended to execute further research for dredging the shallow parts of the river to increase the scale of the used vessels. Also an investigation should be done to multimodal logistics chains. For example the use of conveyor belts for the part between Tribeni and Farakka.

REFERENCES

Gosh, J., *Estimated Fresh Water Draft Forcast January 2011 – June 2011*, Kolkata Port Trust, 2011.

Kumar, S., *Use Inland Water Transport (IWT) for Movement of Coal to Thermal Power Plants*, [www.iwai.nic.in], 2011.

Lootsma, F.A., *Saaty's priority theory and the nomination of a Senior Professor in Operations Research*, 21 December 1978.

Smits, K., *Dry Bulk Shipping Costs*, Royal Haskoning, Rotterdam, 2010.

Veeke, H.P.M., Ottjes, J.A., *Tomas: Tool for object-oriented modeling and simulation*, in Proceedings of the Business and Industry Simulation Symposiom, Washington D.C., 2000.

IN-HOUSE LOGISTICS

Modeling Individual Decision Makers in a Multi Agent Warehouse Simulation

Christoph Schwarz[1] and Axel Hahn[2]
[1]OFFIS – Institute for Information Technology
[2]Carl von Ossietzky University Oldenburg
26121, Oldenburg
Germany
E-mail: Christoph.Schwarz@offis.de

KEYWORDS
Multi Agent Simulations, Individuals, Intralogistics

ABSTRACT

Multi-agent-based simulations are increasingly studied in recent years. Such simulations are generally adaptable and easily parallelizable. A major advantage of these systems is the possibility to model individuals, for example individual drivers of fork lifts who make different decisions about which order to pick next. Material handling systems are nowadays largely simulated by discrete event simulations. In this paper, we show that a simulation of these systems as a multi-agent based simulation with individual decision makers is possible.
With this simulation the influence of individual decisions on the system can be studied. Beside presenting our approach we also present an evaluation with an existing warehouse.

INTRODUCTION

Material Flow Systems (MFS) are today usually studied with the help of discrete event simulation (DES) (see (Wenzel 2006) or (Kuhn and Wenzel 2007)). A variety of such tools is on the market today which demonstrates their importance again. Plant Simulation[1] and AutoMod[2] are two examples and are widely used.
In discrete event simulations, the events are usually represented as flows of goods at fixed times. With these simulations bottlenecks within the material flow system and feasible clock rates for the facilities may be identified. In DES the system analyst models the concrete entities which are involved in the logistics or the material flow such as cranes, floor conveyors, automated guided vehicles (AGVs), etc. Although these entities may be of different types, they do not have a genuine individuality. (See for example (Bangsow 2008))

In recent years multi-agent based simulations (MABS) are increasingly in the focus of research. Davidsson concludes reasons for this:

"However, compared to other approaches, e.g., traditional discrete event simulation, object-oriented simulation and dynamic micro simulation, MABS has a number of interesting properties which makes it useful also for other domains. For instance, it supports structure preserving modeling of the simulated reality, simulation of pro-active behavior, parallel computations, and very dynamic simulation scenarios. It is argued that MABS is a useful technique for simulating scenarios also in more technical domains. In particular, this hold for the simulation of technical systems that are distributed and involve complex interaction between humans and machines." (from (Davidsson 2001))

Several publications have provided evidence that for intralogistic scenarios MABS are slower than DES but, on the one hand, can be adapted more easily and, on the other hand they are easily parallelizable (see (Becker et al 2006) and (Gehrke et al 2006)).
In the area of MFS, some results were published on the basis of MABS. For example, Follert studied with the help of multi-agent systems, whether large-scale handling systems can be controlled by distributed systems (see (Follert and Roidl 2008)). The idea followed by Günther and ten Hompel called "Internet of things in intralogistics" (see for example (Günther and ten Hompel 2010)) uses multi-agent systems in many studies as Scholz-Reiter does in investigating the possibilities of self-organization in logistic processes (see (Scholz-Reiter and Höhns 2006)).
Our question is, however, the impact of individual decisions made by individual drivers and the individual job allocation decisions rather than the material flow as a whole. More precisely, we want to examine the effects of various alternative strategies and cooperation of conveyors in a warehouse, paving the way for a cognitive logistics system ((Beth et al 2011) and (Ommen et al 2010). In the simulation, in addition to these strategies, the

[1] www.plant-simulation.com
[2] www.automod.de

topology of the warehouse, the various types of conveyors and various load profiles play a role.
To investigate these aspects as a first step, we have developed a MABS, in which it is possible that individual truck drivers can make individual decisions about which order to pick next or about which route to take. We also build an instance off this simulation with an existing warehouse. In this paper we present this simulation and the results of the simulation.

REQUIREMENTS FOR THE SIMULATION SYSTEM

For our research questions, it is necessary that the simulation can model individuals. These individuals should be able to answer the following questions independently: what jobs I take on, or about what jobs I am applying? What route do I take? Exactly these points cannot be achieved with DES systems, as they are on the market today and are therefore the most important requirement for the simulation system. It should be easily possible to model these behaviours separate for each agent and to easily change them.
Besides the ability of modelling individuals the simulation systems should fulfil the requirements on a simulation system for warehouse logistics in general. Such requirements include an easy modelling of the scenarios and the load profiles, automatic calculation of indicators and a visualization of the simulation.

SIMULATION

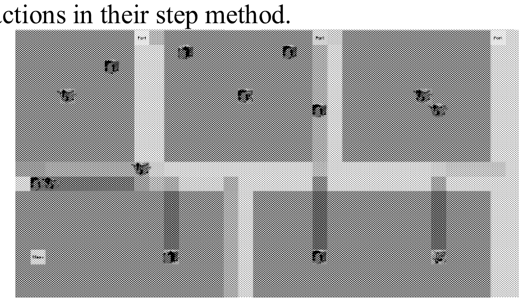
To fulfill all the requirements we had for the simulation system we build up a MASON based simulation system. This chapter describes the structure of this system.

MASON

To simulate the movement of goods in the warehouse, the java library MASON (see (Luke et al 2005)) was selected, which was developed specifically for the construction of multi-agent simulations.
The MASON library is modular. The core of the model library provides the basic tools of MASON. These are essentially a scheduler for temporal control of agents and so-called grids which allow the modeling of the environment in which the agents move. The visualization is provided by the MASON GUI tools. These tools enable a dynamic addition, modification and removal of visualization objects to agents and elements of the environment.
Figure 1 shows a section of a screenshot from the simulation.

Scheduler

The scheduler implements a step-based simulation. Methods of an agent can either be run once in a given simulation step in the future or in a sequence at specific time intervals. To implement an agent that is to be controlled by the MASON scheduler, it must implement the method step from the interface Steppable. Then this class can be marked for execution. The agents then have to calculate their actions in their step method.



Figures 1: Simulation of a warehouse (section)

The scheduler is used in our simulation environment for all of the active agent classes. These are the conveyors and the ports. The timing of the load generation is also performed by the scheduler. Furthermore, the scheduler is used for the control of the disposition of goods.

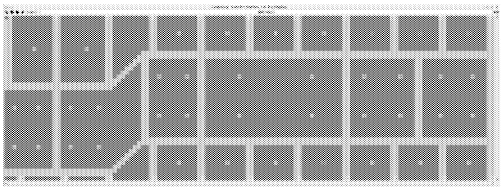
Grids

Grids are used for modeling the environments in MASON. They organize fields in 2- or 3-dimensional matrices. A field contains simple attributes (e.g. encoded with an integer or double value) or objects.
Grids include methods for the determination of proximity relationships. Thus, for example, objects can be determined which are in a specified radius around a particular field. These features are very useful to find other agents to cooperate with or to communicate with. A simulation model can consist of several stacked grids. Neighborly relations cannot be determined across different grids.
In our simulation environment, several grids of different types are used. The most important grids are an ObjectGrid2D to model the static aspects of the warehouse as roadways, ports and storage areas, a SparseGrid2D for modeling the dynamic aspects of the warehouse as the vehicles and two DoubleGrid2Ds to display scheduled and used track assignments. The DoubleGrids2D are also used for the visualization and evaluation of vehicle conflicts (collisions).

Modeling the Scenarios

The modeling of our scenario consists of a model for the environment and of models for the moving objects in the environment. Our environment consists primarily of ports and transport routes.
Figure 2 displays an example of the visualization of our environment. It shows a warehouse with transport routes (light grey) and ports (yellow and green).

Ports are goods transfer points. They are responsible for the identification of goods, for the communication with host systems (e.g. Enterprise-Resource-Planning systems) and for the distribution of orders to transport vehicles. Goods can only be transferred from the outside into the hall or vice versa at these points. In addition, ports serve as buffer areas within the hall.



Figures 2: Simulation of the environment

Ports and Transport Routes

In the simulation the floor plan of a warehouse is modeled by a grid with equal-sized square elements. For each grid field an object accurately represents the particular area in the hall such as a port or a part of a route. The scale of the grid can be chosen. In the example in Figure 2, the fields have an edge length of 1m. In addition, the grid has a graph as an overlay that describes the intersections and the travel routes. The objects that are annotated to the fields represent the conditions of the yard at this field.

Possible objects are: Inport (active), Buffer (active), Outport (active), Driving Area (passive) and Non Driving Area (passive). Here active means that the objects show a behavior controlled by the MASON scheduler. The behavior of the OutPorts is trivial and merely serves to identify indicators. InPorts and BufferPorts have a more complex behavior. Both types are responsible for the management of the goods which are placed on them. This includes the generation of transport orders and the disposition of these orders to the vehicles. The two object types DrivingArea and NonDrivingArea are passive, they do not have an own behavior. Both, active and passive objects are visualized.

PortFloor is the common abstract superclass of all ports. Instances of derived classes define methods to transfer goods (cargo) to ports, to start an auction process and to end the current auction. FloorOutPort requires no auctions, since the OutputPort do not buffer goods. It only receives goods, calculates indicators and removes the goods from the simulation environment.

Goods enter the warehouse always at an input port. In the simulation this is achieved with the Delivery of the InPort (which represents for example a truck). Each InPort regularly controls in its step method whether more capacity for goods is available at the port. If this is the case new goods are taken out of the delivery and placed at the CargoContainer of the port. Depending on the port type (FloorInPort or FloorBufferPort) the port will immediately generate a transport order, or will store the goods for later transportation.

Graphs

The graph describes the ways on which the vehicles can drive in the warehouse. Furthermore the graph contains information about the current and the planed occupancy of road segments. The route map is represented as a directed graph, where each intersection, each branch and each endpoint of a way is a node in the graph. The directed edges of the edge set connect nodes and form the ways in the warehouse.

Static attributes are annotated on both the nodes and the edges of the graph. One attribute of a node is the position in space. Edges have the attributes distance, length, the maximum speed and the maximum load capacity of the route.

In addition, a set of assignments is annotated on the edges as a dynamic attribute. An assignment describes which vehicle is planning to enter the edge at which time and at which time it is planning to leave it. These assignments can be used by the individual agents for planning their route. They can either respect the plans of other vehicles or choose to get into a possible conflict (e.g. at a crossroad).

Load Generation

The boundary of the system under consideration is located at the input and output ports. At the input port, goods are identified and prepared for transport, at the output port they are loaded onto trucks. Since handling processes of the warehouse are neglected, goods must be generated within the ports in a realistic manner. That is the task of load generation. Two load generators were implemented, one database related and one profile related.

The first generator is based on a database which includes information about the movements of goods through the warehouse. In each time step of the simulation, the generator checks with the help of this database whether new goods were taken into the hall at the current time step. If that is the case, these goods are then generated in the input ports.

The second generator is based on a stochastic profile. This profile has probabilities for different cargo types to be generated at given time steps. These probabilities can be related to different times (e.g. weekdays vs. weekends, morning vs. noon, vs. afternoon vs. night). At each time step the different cargos are generated according to their probabilities.

Transport Vehicles

In the simulation, mobile conveyors (e.g. fork lift trucks) are responsible for the transportation of goods. These conveyors always have a cargo space on which the goods are stored during the transport. This has been modeled by a cargo area which encapsulates all the essential properties of this area (geometrical dimensions, maximum weight, etc.). The conveyors also have a graph that describes the transport routes in the area, a plan that describes the following operations to be performed as well as other attributes that describe the position and the

physical dimensions of the conveyor. Even basic methods for planning and execution of transports are defined here. Conveyor is an abstract superclass for different types of conveyors. This should make it easier to simulate also continuous conveyors in later scenarios.
Discontinuous conveyors extend conveyors by adding attributes and methods for description and manipulation of physical parameters (e.g. the speed), for acquisition of transport orders and for the creation and execution of transport plans.
Conveyors are modeled as agents whose goal is to acquire transfer orders and to execute them. To fulfill these goals, they need information from their environments. This information is used in conjunction with the internal state of the agent in order to plan and execute tasks. Ultimately, the planned actions are translated into movements which are executed by the conveyors. These movements then alter the environment.
The job allocation takes place in cooperation with the input ports and the buffer ports which hold open orders. Depending on the chosen strategy, these orders can be directly assigned to the trucks or can be placed in an auction process.
At the beginning of the execution of an order, the vehicles plan the route from their current location to the location of the goods and from there to the destination. To calculate the route the vehicles take the current and future truck route assignments into account, which it gets as input from its environment. It can also take individual preferences into account. Examples for such preferences could be less turns or as few other vehicles in the area as possible. In other words each agent can have its own method of calculating its routes. Since the decision which route to take is one of the important decisions a vehicle or a driver has to make in this scenario a good part of his individuality is modeled with these methods. According to our industry partner, also the individuality of real live drivers can be seen by the routes they choose.
The vehicle then allocates the way segments that lie on the determined route. Afterwards, the occupancy information of the environment is updated. Since the occupancy information of the environment, which is calculated by adding the assignments of all vehicles, is dynamically changing in this way, the vehicles regularly need to perform replanning during the execution. This behavior corresponds to the observation of the situation in the warehouse by human drivers.
The execution determines necessary movements and manipulations of goods operations (picking up and setting down of goods) and puts them in order. Plan changes caused by new plans are also considered here.

Order Disposition

An auction based order disposition was developed. In this auction process open transfer orders are distributed through interaction between ports and vehicles. The initiative emanates from the ports. An auction is started when a good is due to be delivered to an output port. The port starts the auction by offering a new proposal which has a specific duration. In addition, all vehicles will be informed about the new transport offer.
Now it is up to the vehicles to bid on the proposal. They only do that when they are currently not involved in any other transport order and when they are capable of carrying out this transport in terms of weight and size of the goods. Finally a vehicle can decide not to bid on a proposal due to individual preferences (e.g. do not bid on proposals in the eastern part of the warehouse or do not bid on proposals that are far away from their current position).
In the case a vehicle wants to bid on a proposal it calculates the time it would need to execute the order (using the information from the environment and its individual routing algorithm) and sends this time to the proposal. It is also possible to modify the calculated time with a preference-factor. In this way a behavior like “I don’t want to take orders in the eastern part of the warehouse but I will if no other vehicle can do it.” can be modeled. Besides the way a vehicle calculate its route, this is the second part of the individuality we modeled for this scenario.
At the end of an auction the port chooses the best bid, namely that one with the lowest costs. Afterwards, it sends a message to all vehicles which participated in the auction: the winner will receive a ProposalAccept message and the others receive a ProposalReject message.

EVALUATION

To evaluate our simulation system we built a simulation of a warehouse. For this we have used the log of a warehouse of one of our industry partners. The design of the warehouse was available and the movements of all goods from 2009-07-01 until 2010-07-01 were collected. The hall has four imports, four outports, and 36 buffer ports. In total 1,388,761 packages were moved during the given interval in the warehouse. For these packages, the following information was provided to us in a database: loading and unloading time stamps, volume, weight, inport and outport. So we used the database related cargo generator for this simulation. Also the rules for order picking and route calculation in the given warehouse were known. We used these rules to model the behavior of the agents.
We had to select indicators in order to obtain a comparison between the data made available by the industry partner and the data of the simulation. The following indicators were used by us for that comparison: delivered packages, delayed packages (absolute and relative), average delay per package,

average transportation time per package and average storage time per package.
The aim was to replicate the processes in the warehouse in our multi-agent based simulation. Table 1 shows that the simulation reflects the real situation to a large extent.
When looking at the indicators it is noticeable that some values differ slightly, including the delivered packages, which differ by 1,229. This difference can be explained by the fact that packages in the database are removed before the end of simulation, and therefore could not be accounted for in the simulation. Similarly, the average processing time per package in the simulation is slightly increased: namely 55 delays cause an average delay of about 2 minutes per package.
Subtracting the average delays from the processing time, a lead time of 11:56:38 is reached, which only causes a difference of 39 seconds. Analogously, the slight difference in the average storage time for each package can be explained.

Table 1: Comparison between simulation and reality

Indicator	Reality	Simulation
delivered pack. absolute	1.388.761	1.387.532
delayed pack. absolute delayed pack. relative	0 0%	55 0,004%
average delay per package (sec)	0	130
average transportation time per package (hh:mm:ss)	11:55:59	11:58:48
Average storage time per package	11:59:46	11:55:48

CONCLUSIONS

The presented multi-agent-based simulation is capable of simulating the operations of a yard or a warehouse. All relevant indicators have been achieved within the tolerance. Furthermore, all the decision makers were simulated as agents, which are capable of individual order picking decisions and of individual routing algorithms. Therefore it is possible to investigate the impact of individual decisions in a warehouse with the help of the presented simulation system.
In the future we will investigate the impact of such individual decisions on the important logistic key indicators. Some questions that arise here are: is it useful that the vehicles have different order picking behaviours (for example that some vehicles only pick orders nearby and some only orders which are urgent)? Can it be useful that some vehicles avoid the middle of a warehouse?
Furthermore different conflict resolution options will be implemented in the simulation system. Conflicts can occur in order for route planning and scheduling (avoid each other in close corridors). In general, we want to use the MABS presented here to investigate questions that cannot be studied with classical discrete event simulations meaning questions that depend on individual decisions.

ACKNOWLEDGEMENTS

This work was partially founded by European Regional Development Fund (ERDF). A large part of the work presented in this paper was done by Carsten Beth and Wilm-Hendrik Scholz.

REFERENCES

Bangsow, S (2008): *Fertigungssimulationen mit Plant Simulation und SimTalk*. Hanser Verlag, München.

Bates, J.J. and Roberts, M. (1983), Recent Experience with models fitted to Stated Preference Data, *Proceedings 11th PTRC Summer Annual Meeting*, University of Sussex, England, 4-7, July 1983.

Becker, M.; Wenning, B.-L.; Görg, C.; Gehrke, J.D.; Lorenz, M.; Herzog, O. (2006): Agent-based and discrete event simulation of autonomous logistic processes. In: Borutzky, W.; Orsoni, A.; Zobel, R. (Hrsg.): *Proceedings of the 20th European Conference on Modelling and Simulation*. Sankt Augustin, Bonn.

Beth, C., Kamenik, J., Ommen, D., Hahn, A. (2011): Design aspects of cognitive logistic systems. In Kreowski, H.-J.; Scholz-Reiter, B.; Thoben, K.-D.: *Dynamics in Logistics*. Springer, Berlin.

Castilho, D. B. and Daganzo, C. F. (1993), Handling Strategies for Import Containers at Marine Terminals, *Transportation Research*, l 27B, 151-166.

Davidsson, P. (2001): Multi Agent Based Simulation: Beyond Social Simulation. In: Moss, S.; Davidsson, P. (Hrsg.): Multi-Agent-Based Simulation. Springer, Berlin.

Follert, G.; Roidl, M. (2008): Evaluation of Routing Strategies for Decentralized Self-Organisation in Large Scale Conveyor Systems. In: Kimberly, E. et. al. (Hrsg.): *Proceedings of the 10th International Material Handling Research Colloqium*. Material Handling Industry of America, Charlotte, USA.

Gehrke, J. D.; Lorenz, M.; Wenning, B.-L.; Becker, M. (2006): Integration of two approaches for simulation of autonomous logistic processes. In Wenzel, S. (Hrsg.): *Tagungsband zur 12. ASIM Fachtagung*. SCS Publishing House e.V., Erlangen.

Günthner, W.; ten Hompel, M. (Hrsg.) (2010): *Internet der Dinge in der Intralogistik* (VDI-Buch). Springer, Berlin.

Johnson, L. W. (1990), Discrete Choice Analysis with Order Alternatives, In M. M. Fisher, P. Nijkamp and Y. Y. Papageorgiou (eds), *Spatial Choices and Processes*, North Holland, Amsterdam.

Kuhn, A.; Wenzel, S (2007): Simulation logistischer Systeme. In: Arnold, D. et al. (Hrsg.), *Handbuch Logistik*. 3. Auflage. VDI Springer, Berlin Heidelberg.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G. (2005): Mason: A multiagent simulation environment. *Simulation 81*: 517–527

Ommen, D., Kamenik, J., Beth, C., Busch, J.C.; Kulas, A.; Cramer, E.; Hahn, A. (2010): An Adaptable Architecture for Intelligent Conveyors. In Filipe, J.;

Andrade-Cetto, J.; Ferrier, J.-L. (Hrsg.): *ICINCO* (1). INSTICC Press.

Scholz-Reiter, B.; Höhns, H. (2006): Selbststeuerung logistischer Prozesse mit Agentensystemen. In: Schuh, G. (Hrsg.), *Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte (VDI-Buch)*. Springer, Berlin.

Wenzel, S. (Hrsg.) (2006): Simulation in Produktion und Logistik. *Tagungsband zur 12. ASIMFachtagung Simulation in Produktion und Logistik*. SCS Publishing House e.V., Erlangen.

Winston, W. L. (1993), *Operations Research Applications and Algorithms*, Duxbury, Belmont, California.

SIMULATION IN THE DESIGN OF AN INTERNAL LOGISTIC SYSTEM – MILK RUN DELIVERING WITH KANBAN CONTROL

Luís S. **Dias** Pavel **Vik** José A. **Oliveira** Guilherme **Pereira**
Production and Systems Department,
University of Minho, ALGORITMI Research Centre
Campus de Gualtar; 4710-057 Braga, Portugal
E-mails:
lsd@dps.uminho.pt, vikpavel@seznam.cz, zan@dps.uminho.pt,
guilherme.pereira@algoritmi.uminho.pt

Keywords: milk run system, kanban system, internal logistics, computer simulation, CAD, Layout

ABSTRACT

This paper describes an internal logistic system that uses Milk Runs (MR) also called Logistic Trains (LT) to transport parts in an electronic industry (production of car radios and GPS navigators).
The MR design and process analysis are performed using computer simulation as part of a developed tool (IDS) that integrates simulation, database and CAD system together. By using simulation models for research and experimentation, we improve our knowledge about the system, prior to implementation or even changing the real system.
The project outcomes, based on the simulation optimization, lead to the reduction of logistic costs and improvement in the availability of material in production lines. These in turn, reduce work in process material (by reducing stock levels and buffers in the production lines) and setting the minimum number of MR and kanban units.

1 INTRODUCTION

1.1 Description of MR principle

Milk **R**un (MR) is one method for aggregated supplying of production lines.
Figure 1 shows main differences between a "taxi" concept (e.g. forklifts) and logistic train (milk-run).

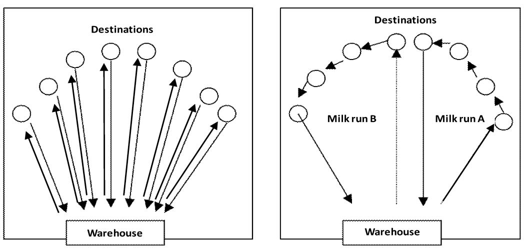


Figure 1 – Taxi and Train supplying concepts

The main principle of the MR is supplying goods from deposits to delivery points in fixed defined routes and times (Costa, Dias, Oliveira and Pereira, 2007; Jiang, Huang and Wang, 2010; Satoh, 2008; "http://www.ipaslovakia.sk" 2011). It can be controlled using a kanban pull system to achieve higher system's performance.
MR's can be compared with the subway train that is directed by a defined schedule: it stops at each station, takes in a few passengers while a few get out. Originally, this idea and name, came from the past when the cars delivered milk at defined times from the milk farms. (Tanchoco, 1994; Taylor, 2009). This principle can be used in internal and external logistics. The most used facilities for MR are the logistic trains that consist of one engine (locomotive) and wagons (www.fantozzi.com).
Figure 2 represents such a system, where the use of kanban is integrated with the use of a MR. In the "Warehouse", kanban units (e.g. boxes with required items, where each box is claimed by a kanban card) are loaded (1)(2). After moving to the delivering point near production line *(Workplace#1)*, full boxes are unloaded, empty boxes are loaded (3)(4). Each empty box, with a kanban card, is replaced by a full box in the "Warehouse", to be delivered in line, in the next MR turn. The number of kanban cards limits the number of boxes/units that can be in the production line, avoiding high WIP (*Work in Process*) and many associated costs (Gross and McInnis 2003).

1.2 Description of the project and tasks definition

MR delivering system and Kanban system need the following information (considered in this paper):

- number of MRs in the system and its % utilization
- structure of delivering items
- cycle time of MRs and scheduling (to avoid traffic jams/queuing in the common routes)
- routes and network of stations (factory layout)
- politics of kanban system (e.g. number of units)
- kanban transport units properties (e.g. size, capacity)

1.3 MR system specification

For the design of the MR logistic system, we have to consider the MR system boundaries, working policies, routes, stops and MR physical features.
In Figure 3, there is a schema of the supplying processes in the factory, producers (External Suppliers) of electronic parts supply the *Warehouse*

in no-regular cycles (days, weeks) in larger packages of goods (e.g. pallets). Every 120 minutes, the *Supermarket* is supplied from the *Warehouse*. From Supermarket, stations (buffer areas near production lines) are supplied by a milk-run system, every 20 minutes.

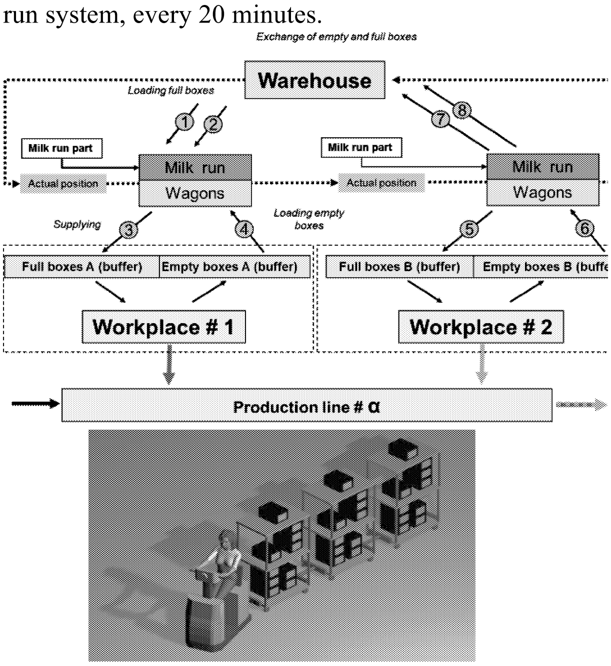


Figure 2 - Principle of MR supplying with wagons and kanban boxes

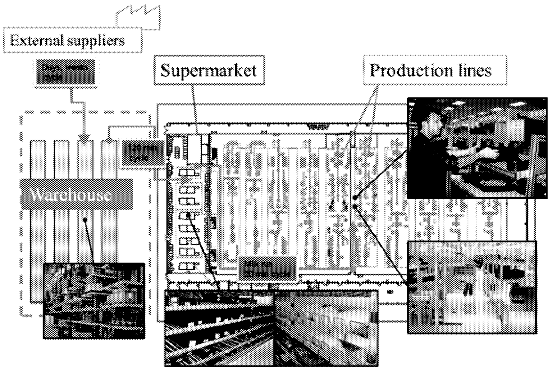


Figure 3 - Inputs for the MR project

The factory production facilities layout (Figure 4) has 10 production lines and supermarket (warehouse). This logistic supplying system, using MR, is the subject of this paper work.

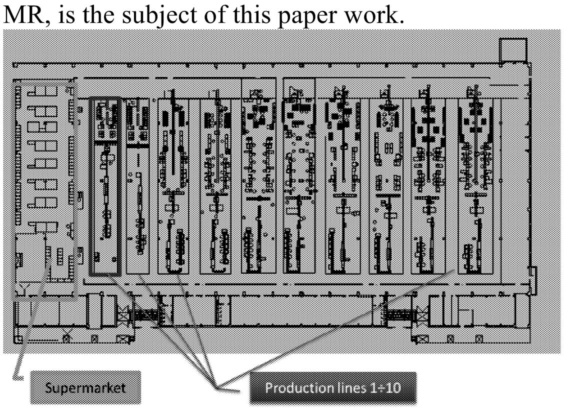


Figure 4 - Production Plant Layout

Each Loading/unloading place (station) in supermarket and production lines is defined along MR path. Each MR has around 10 stations.

The MR tractor used by the company, has the physical features shown in Figure 5. (source: "http://www.fantozzi.com," 2011)

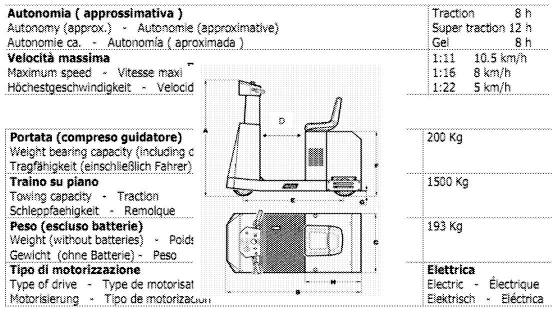


Figure 5 - MR specifications

2 SYSTEM DESIGN

Designing a system with good performance, considering all constrains, is a complex mission. It is necessary to consider space factory limitation, inventory (equipment), schedules, transported volumes, etc., to get optimal and flexible solution.

In this project it is used a complex developed system that integrates the use of AutoCAD and Witness Simulation ("http://www.lanner.com," 2011) through a MS Access database, with a set of built-in functionalities that supports advanced moving methodologies of entities inside the system. This approach made possible to achieve a holistic system design. That tool is named IDS (Integrated System's Design), started in 2005, is now mature, and has been developed by the same research team that signs this paper. For more information about the IDS see following sources - (Vik, Dias and Oliveira, 2007; Vik, Dias and Oliveira., 2008; Vik, Dias, Pereira and Oliveira, 2010).

In this project, the following functions from the IDS were used:

- automatic generation of a simulation model
- milk-run transportation system
- generation of CAD layouts and material flows
- shortest path searching and routes design

Two models with different levels of detail and different insights have been developed: **Model α** studies the number of MR units and **model β** studies the individual detailed behavior of MRs.

These two models need some type of integration, where information and data are shared.

Figure 6 shows data interconnection of the two models (α and β). Together, they both describe completely the behavior of this internal logistic system. Data is shared, between model α, where the CAD **layout** is more important, and model β which

is focused on the **process** production management policy – kanban setup.
This is also one of the general benefits of the IDS – sharing information between models.

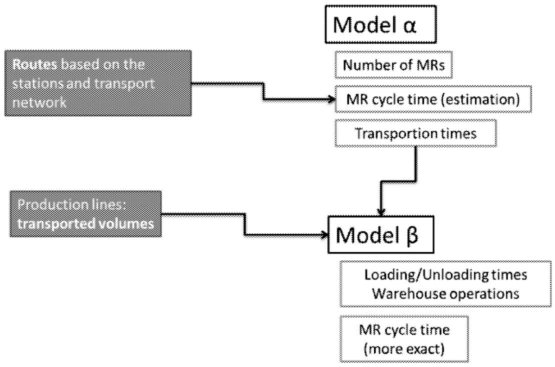


Figure 6 - Joined simulation models

3 MODEL α (FULL SYSTEM – 10 MRS)

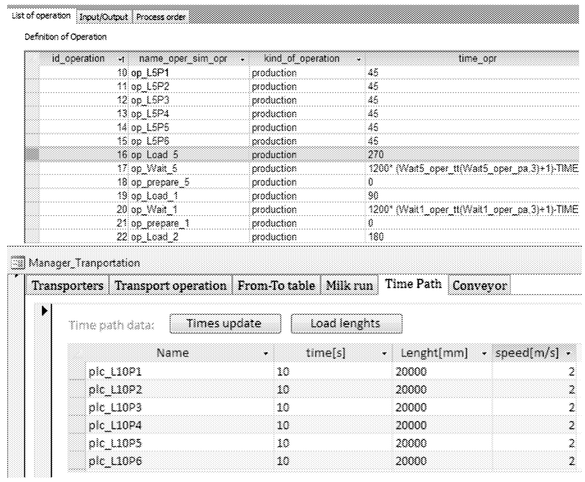
Model α, studies the use of 10 MRs, each supplying one production line. Each MR is represented by an entity, with defined routes (sequence of processes with production times and loading/unloading operation time).
Transportation times are calculated automatically by IDS, based on speed and shortest transportation distances, calculated in the CAD layout.
This model does not include exact loading/ unloading volumes (number of boxes) but only times for these operations (see model β, for detailed times). The main aim is to **design transport routes** and the required MR cycle time.
Expected outputs from model α are:

- Routes design;
- Usage of MRs;
- Traffic accumulated frequency in the main roads (shared by several MRs).

For this purpose, the waiting time of each MR is given as:

```
waiting time=
(MR_cycle_time)*f(number of operation)-TIME
⇔
waiting time = 1200 * P_2_tt(P_2_pa,3)-TIME
```

Where, f(number of operation) represents the total number of performed cycles(laps) including the current run ("P_2_tt(P_2_pa,3)"), and TIME is the simulation time. 1200 is the cycle time in seconds. Supposing that the simulation time (TIME) is exactly 11700, and the MR is ready to go, it must wait 300 seconds, until the end of the 10^{th} cycle. Waiting time = 1200*10-11700.
Figure 7 shows excerpts of data tables from the project database. In the table "List of operation", (see Figure 7, first image), it is specified that **Unloading** items at each station needs 45 seconds (as default value), **warehouse operations** (loading full boxes in the Warehouse) take 45 * number of stations (ex: MR#5 needs 45 x 6 = 270 seconds). In the table "Time Path", (see Figure 7, second image), it defaulted the **Average length** between stations as 20 meters and **default speed** as 2 meters per second.
Based on the DB system specification, IDS generated the first CAD schematic layout and simulation Witness model, both in diagonal shape and with linear material flows. The Figure 8 shows a CAD screenshots with the generated layout based only on the distances (20m) between stations, and including simulation run outcomes, represented by lines. The width of the lines is proportional to the flow (number of movements that was performed through each specific path).


List of operation | Input/Output | Process order

Definition of Operation

id_operation	name_oper_sim_opr	kind_of_operation	time_opr
10	op_L5P1	production	45
11	op_L5P2	production	45
12	op_L5P3	production	45
13	op_L5P4	production	45
14	op_L5P5	production	45
15	op_L5P6	production	45
16	op_Load_5	production	270
17	op_Wait_5	production	1200* (Wait5_oper_tt(Wait5_oper_pa,3)+1)-TIME
18	op_prepare_5	production	0
19	op_Load_1	production	90
20	op_Wait_1	production	1200* (Wait1_oper_tt(Wait1_oper_pa,3)+1)-TIME
21	op_prepare_1	production	0
22	op_Load_2	production	180

Manager_Tranportation

Transporters | Transport operation | From-To table | Milk run | Time Path | Conveyor

Time path data: Times update | Load lenghts

Name	time[s]	Lenght[mm]	speed[m/s]
plc_L10P1	10	20000	2
plc_L10P2	10	20000	2
plc_L10P3	10	20000	2
plc_L10P4	10	20000	2
plc_L10P5	10	20000	2
plc_L10P6	10	20000	2

Figure 7 - Setup of operation times and Transportation

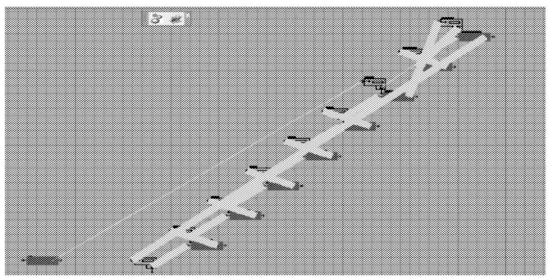


Figure 8 - First generated schematic layout

3.1 Model α – Enhancing the model (realistic transport network distances)

Follows several steps with information enhancements on model α , detailing respective processes.

3.1.1 [A] MR routes definition on CAD layout

The design of the MR system in CAD is processed by generating blocks that represent resources in the factory layout and placing them into positions (stations) next to the production lines. After that, each **route** can be automatically generated by the function "Shortest path" that searches the smallest distances in the transportation network. Figure 9 highlights two milk run routes: Milkrun#5 (red colour) and Milkrun#3 (blue colour). The underlying transportation network, in light orange colour, was designed in CAD over physical layout.

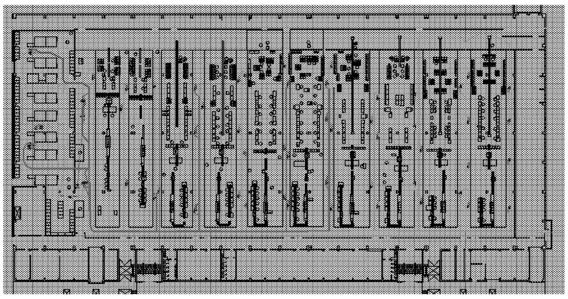
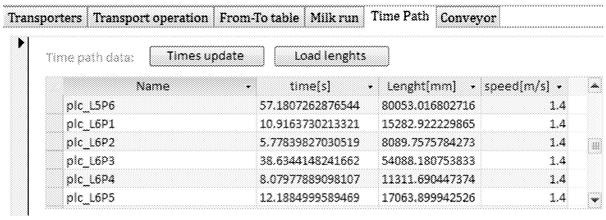


Figure 9 - Actual layout of MR routes

3.1.2 [B] Distances update into DB

Next step is to save **realistic routes information** (length of each transport route section) into the DB and recalculated **transport time** (speed has been also adjusted to 1.4 m per second). Figure 10 shows the "Time Path" DB table with updated distances values and movement times.



Transporters | Transport operation | From-To table | Milk run | Time Path | Conveyor

Time path data: Times update | Load lenghts

Name	time[s]	Lenght[mm]	speed[m/s]
plc_L5P6	57.1807262876544	80053.016802716	1.4
plc_L6P1	10.9163730213321	15282.922229865	1.4
plc_L6P2	5.77839827030519	8089.7575784273	1.4
plc_L6P3	38.6344148241662	54088.180753833	1.4
plc_L6P4	8.07977889098107	11311.690447374	1.4
plc_L6P5	12.1884999589469	17063.899942526	1.4

Figure 10 - Actual distances updated into DB

3.1.3 [C] Transportation movement frequency

The transportation movement frequency, one output of simulation, is a very important information. It gives the number of movements in each section of the transportation network. Figure 11 shows, in purple colour, that the highest values are near to the Warehouse (around 1100 movements per day, considering the sum of both directions, it means that, in average, every 75 sec, some MR crosses this network section).

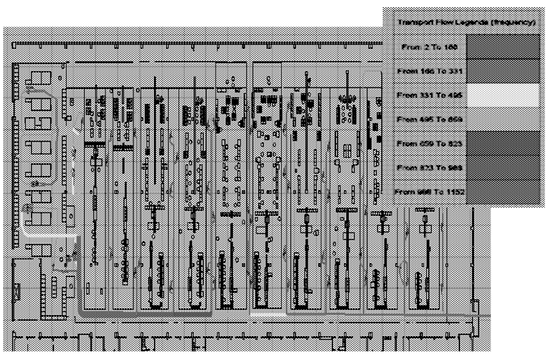


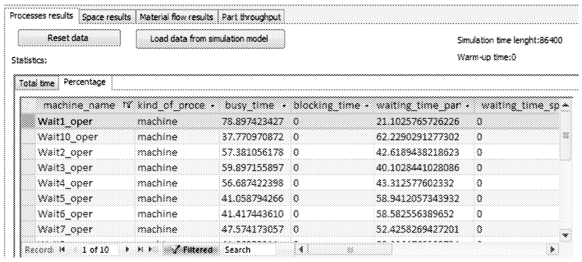
Figure 11 - Transportation movement frequency

3.1.4 [D] Milk Runs usage analysis

Results from simulation model based on updated layout values (distances and actual movement times) are shown in Figure 12. This table shows Milk Runs usage report. For example, it shows that MR#5 is working (transport, load, unload) around 59% of the total time, the remaining is waiting in the warehouse (41%) for next supplying round.

(Note: MR must stay in the Supermarket until the beginning of a new cycle. for instance, the process that makes the MR#5 to wait is named **Wait5_oper**. The value of "busy_time" of that process is percentage of time that the MR#5 is waiting (41%). The value of "waiting time" is the percentage of time that the MR#5, is, actually, moving around its cycle (59%)).

The starting times of MRs are spaced by 120 seconds, to reduce the possibility of congestion, however, the space occupation by MRs while waiting for next cycle was not evaluated and could be critical in case of scarce space in supermarket output area.



Processes results | Space results | Material flow results | Part throughput

Reset data | Load data from simulation model

Simulation time lenght:86400

Statistics:

Warm-up time:0

Total time | Percentage

machine_name	kind_of_proce	busy_time	blocking_time	waiting_time_par	waiting_time_sp
Wait1_oper	machine	78.897423427	0	21.1025765726226	0
Wait10_oper	machine	37.770970872	0	62.2290291277302	0
Wait2_oper	machine	57.381056178	0	42.6189438218623	0
Wait3_oper	machine	59.897155897	0	40.1028441028086	0
Wait4_oper	machine	56.687422398	0	43.312577602332	0
Wait5_oper	machine	41.058794266	0	58.9412057343932	0
Wait6_oper	machine	41.417443610	0	58.582556389652	0
Wait7_oper	machine	47.574173057	0	52.4258269427201	0

Record: 1 of 10 Filtered Search

Figure 12 - MRs usage report

Cycle time of 20 minutes, as a company standard, for each MR seems to be enough, if this time is too short, there appears an error notice during simulation running, explaining that *waiting_time* got a negative value.

The longest way (MR#10) needs 13 minutes – that is absolute minimum time for MR cycle, with 7 minutes of spare time at this stage of model detail.

Some of the MRs need less time once routes are shorter. Anyway, for uniformity purposes, a common time was set up for all of them.

3.2 Model α – Decreasing to 5 MRs

Based on results already shown in Figure 12, MR utilization (moving, loading or unloading) is only around 50%. This waste of time motivated a study to reduce the number of MRs, assuming that one MR can supply two production lines.

The routes layout for of each of the 5 MRs was defined as shown in.Figure 13.

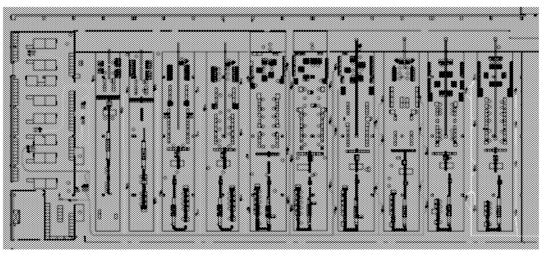


Figure 13 - Routes layout with 5 MRs

Comparing with the actual layout with 10 MRs, where each MR crosses two corridors, making left side and right side of one specific production line,

in this layout with 5 MRs, each MR supplies two consecutive corridors, depicted by the same colour.
For example, the MR#4, represented in green, supplies: left side of line 3, complete line 4 and right side of line 5.
The usage of 5 MRs is reported in Figure 14.

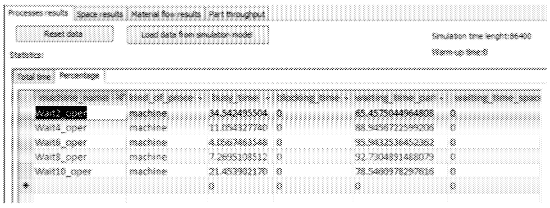


Figure 14 - Usage of 5 MRs

This model shows that it is theoretically possible to reduce the number of MRs from 10 to 5. This reduction represents significant operational costs savings.

4 MODEL β (DETAILED SIMULATION OF ONE MR)

This model contains more details describing MR behavior, as loading/ unloading volumes and different kind of items. It was decided to simulate this detail level to just one MR, once they all have an analogue behavior and they are **independent** of each other. Routes and supermarket are not occupied (low traffic) and there is small risk of some kind of blocking (traffic jam, queuing).

Main aim of this model is to:

- parameterize the **kanban system** (replacing empty boxes by full boxes)
- establish **safety level** of buffers near production lines and buffer sizes
- improve MR cycle time calculation
- calculate transported volumes in each MR cycle and totally
- calculate accurate number of MR wagons necessary

Buffer sizes are dependent of cycle time and affect and are influenced by the layout.
In the previous model, the loading time is an exact value of 45 seconds. In this model, loading times are based on the number of transported items:

```
loading time in warehouse =
number of items x 10 sec (constant value)
```

4.1 Simplified analytical calculation of kanban cards units number

A simplification is used in the simulation model - kanban cards are tags in the transportation boxes (transportation unit) which are changed empty boxes by full boxes in the stations and in the supermarket (full by empty).
The adjustment of their number is a very important system parameter because a low number can stop the production (lack of material) whereas a high number can be a wasting (excessive storage near production links, extra space, etc.). It is influenced by several factors as:

- production specification
- consumption of material within a time period (average vs. maximum)
- number of parts represented by one kanban card (transportation batch)
- reaction time of supplier (external lead time)
- transportation time between supermarket and station
- quality of parts (percentage of unconformity of input parts)
- desired safety level of storages to covers errors in the production, breakdowns, etc.

Gross and McInnis (2003), Krieg (2005) and Satoh (2008) use deterministic calculation to establish kanban units in the system:

$$N = \frac{\mathrm{D.}\,(\mathrm{T}_w + \mathrm{T_p}\,)\,(1 + \mathrm{a})}{\mathrm{c}}$$

where:
D = consumption of parts by time unit [parts per hour]
T_w = waiting time for kanban batch, transportation times [hours]
T_p = time for processing batch kanban parts, e.g. by assembly operation [hours]
a = safety coefficient
c = number of parts in one kanban transportation unit (capacity)

The major problem is to quantify **the safety coefficient** (a) covering many factors as listed above. This value can be from 0.2 up to 9. Due to the difficulty and inaccuracy, the **simulation can be used** to establish the appropriate number of kanban units and to deal with all uncertainties.

4.2 Simulation approach in the calculation of kanban cards units number

With simulation, we can include in the model the system's stochastic nature and milk run behavior.
By the test of several values/scenarios, it is chosen the optimal number of kanban card units.
Several factors influence each other:

- number of kanban units (of each part type)
- MR cycle (Cycle time, route, stop points and stop time)
- consuming speed (of parts in each production point)

One complex simulation model can be split into several experiments to understand more easily **particular results** and to demonstrate **relationships** between all factors, which are: the number of kanban units, storage safety level of items in buffers during milk-run cycle time and different consumption of supplied items.

4.2.1 *Experiments A: Number of kanban units*

This set of experiments shows the influence of kanban units number in the system. In Figure 15 (upper table), there is a set of input data with different number of kanban units in the system for each product (5-10-12, etc).
It was considered a cycle time of 20 minutes and consuming speed of 1 box in every 200 seconds.

Part_name	Crate_name	Sequence_	variant_ordinary	Number_arrived	First_Arrived_tim	Inter_Arrived_tim	Lot_size_in
A1	part		1	5	0	1	1
B1	part		1	10	0	1	1
C1	part		1	12	0	1	1
D1	part		1	13	0	1	1
E1	part		1	15	0	1	1
F1	part		1	20	0	1	1

Total time | Percentage

machine_name	kind_of_proce	busy_time	blocking_time	waiting_time_par	waiting_time_spac
L5P1	machine	41.653137735	0	58.3468622652824	0
L5P2	machine	83.314581838	0	16.6854181624472	0
L5P3	machine	96.464733583	0	3.53526641727975	0
L5P4	machine	99.788993553	0	0.211006446823919	0
L5P5	machine	99.770379850	0	0.229620149855398	0
L5P6	machine	99.756490809	0	0.243509191168238	0

Figure 15 - Model β: number of kanban units

In the lower table of Figure 15, the results show that the aim of 100% working production lines can be achieved with **13** or more boxes.[1]

4.2.2 *Experiments B: Safety level*

Safety level means the minimum number of items that must be in buffers. This number should be **enough to** cover of at least **2 unrealized MR cycles** to avoid that the production stops because of human mistakes, breakdowns, low quality of delivered parts or other unpredictable reasons.

$$safety\ level\ value = \frac{2\ x\ MR\ cycle}{average\ consumption\ time\ for\ 1\ box}$$

In the system, where ***MR cycle*** length is 1200 seconds and has an ***average consumption time of 1 box*** in 200 seconds, the ***safety level value*** is **12** units (2*1200/200). This value (12) is added added to the value established in Experiments A (13), so the number of kanban units in the system is then established to 25 (12+13).
In Figure 16 (upper table), there are different kanban units in the system. For the parts "B1" and "D1", there are 10 kanban units and an established value of 25. The plot graphs in Figure 16 (lower image) display fluctuations in the buffer's occupancy during the simulation: the lower curve is for the buffer without safety level and the upper curve is for the buffer with safety level 12. The number of kanban units does not go under this safety level.
In the beginning of the plot (until time = 2400 seconds), there is a warm-up period of simulation (filling of empty buffers).

[1] The difference to 100% in the last three lines in *busy_time* is due to the lack of warm up period.
Some "noise" is also introduced in the first MR round where more boxes make the first delivery longer.
This model was generated by **IDS** in WCL by VBA from MSAccess database. This automatic generator does not support, yet, an automatic warm up period usage.

Part_name	Crate_name	Sequence_	variant_ordinary	Number_arrived	First_Arrived_tim	Inter_Arrived_tim	Lot_size_in	variant_sto	for
A1	part		1	5	0	1	1		
B1	part		1	10	0	1	1		
C1	part		1	24	0	1	1		
D1	part		1	25	0	1	1		
E1	part		1	27	0	1	1		
F1	part		1	32	0	1	1		
* part_			0	10	0	1	1		

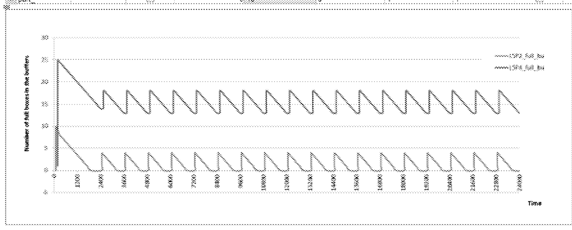


Figure 16 - Number of kanban units in buffers

4.2.3 *Experiments C: buffer size based on a different MR cycle time*

In this set of experiments C, it is analysed the consequences of increasing the cycle time of MR. This experiments assumes the same speed consumption of stations (1 box in 200 seconds).

Table 1 – Model β: Table of experiments C data

Number of inputing "Parts D" Safety level	25 12	28 15	36 24	48 36
Number of kanban units in the buffer ("Parts D")	25	31	48	72
MR cycle time [seconds]	**1200**	**1500**	**2400**	**3600**
Average (after stabilization 2 MR cycles)	15.68	19.64	29.76	44.8
Maximum number of boxes in buffer	18	23	35	53
Maximum number of boxes in MR	35	54	96	162

Minimum value for MR was established in model α = 20 minutes (1200 seconds). Table 1 contains several simulation output values for different MR cycle time, from 1200 to 3600 seconds.

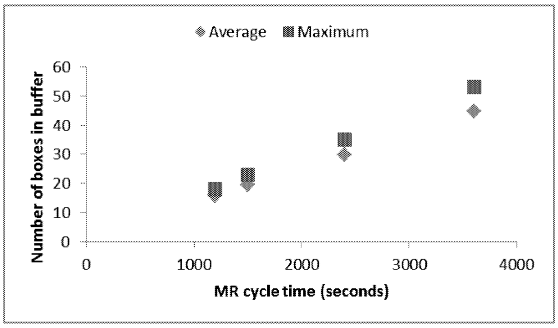


Figure 17 - Plot of buffer size vs. MR cycle times

There is a **linear correlation** (see Figure 17) that describes the buffer size – longer period means requirement for larger buffers, whereas shorter period can cause an inefficient use of MRs (higher percentage of time of useless movements, or even running empty).
The maximum number of boxes in MR defines the load capacity and so the number of wagons. One wagon can carry around 20 boxes. For a MR cycle of 3600 seconds it may be necessary to use 162/20 = 8 wagons in same cases. More than 3 wagons compromises MR maneuverability.

4.2.4 *Experiments D: Different consumption in stations*

This set of experiments evaluates the impact of different levels of time consumption of boxes in the stations.

It is relevant to analyze the **speed consumption of delivered items**, which is influenced by external customer demands. These demands can be very hard to predict. Therefore, several scenarios are tested (different speed consumptions) to assure that the MR configuration is correct.

Base Conditions:
1200 MR cycle time
safety levels – different numbers based on consumption and 100% utilization of production line (each item have different consumption speed, aim is to test whole MR system).

In Figure 18, the set of tables shows input data and experiment results. Different operation times represent different consumption of delivered items. The definition is based on the stochastic function *"Normal(200,40)"*, which means a normal distribution function with a mean = 200 seconds and standard deviation = 40 seconds. Different numbers of kanban units are also defined in the system, based on the average consumption.

Results of the use (Figure 18, third table) display that the number of items ("D1","E1" and "F1") should be increased because there is a lack of items and production line facilities (L5P4, L5P5 and L5P6). Also, they do not work on 100 %. After adding more units, all stations of production line work on 100%, as shown in Figure 19.

List of operation | Input/Output | Process order

Definition of Operation

id_operation	name_oper_sim_opr	kind_of_operation	time_opr	time_min_bu	blocation	set
10	op_p1	production	Normal(200,40)	0		0
11	op_p2	production	Normal(100,20)	0		0
12	op_p3	production	Normal(50,5)	0		0
32	op_p4	production	Normal(400,70)	0		0
33	op_p5	production	Normal(600,100)	0		0
34	op_p6	production	Normal(1000,100)	0		0

Part_name	Crate_name	Sequence_	variant_ordinary	Number_arrived	First_Arrived_tim	Inter_Arrived_tim	Lot_size_in
A1	part		1	24	0	1	1
B1	part		1	70	0	1	1
C1	part		1	100	0	1	1
D1	part		1	12	0	1	1
E1	part		1	8	0	1	1
F1	part		1	4	0	1	1

Total time | Percentage

machine_name	kind_of_proce	busy_time	blocking_time	waiting_time_par	waiting_time_sp
L5P1	machine	97.034524026	0	2.96547597441226	0
L5P2	machine	99.161354210	0	0.838645789540845	0
L5P3	machine	99.764524301	0	0.235475698630954	0
L5P4	machine	95.513134723	0	4.486865276527	0
L5P5	machine	92.642687329	0	7.35731267114816	0
L5P6	machine	82.148357514	0	17.8516424864223	0

Figure 18 Model β: Different consumption of items

Part_name	Crate_name	Sequence_	variant_ordinary	Number_arrived	First_Arrived_tim	Inter_Arrived_tim	Lot_size_in
A1	part		1	24	0	1	1
B1	part		1	70	0	1	1
C1	part		1	100	0	1	1
D1	part		1	15	0	1	1
E1	part		1	10	0	1	1
F1	part		1	6	0	1	1

Total time | Percentage

machine_name	kind_of_proce	busy_time	blocking_time	waiting_time_par	waiting_time_spac
L5P1	machine	99.034524026	0	0.96547597441226	0
L5P2	machine	99.161354210	0	0.838645789540845	0
L5P3	machine	99.764524301	0	0.235475698630954	0
L5P4	machine	99.684826887	0	0.315173113490586	0
L5P5	machine	99.620026089	0	0.379973911439662	0
L5P6	machine	99.652324142	0	0.347675857834904	0

Figure 19 – Increasing kanban units (D1,E1,F1)

4.2.5 *Safety Buffers Levels*

Basically safety levels are established as a number of boxes consumed within two MR cycles (see experiments B). If a time is described stochastically, the mean of the distribution is taken and tested. For example, Normal (100,20), safety level can be = 2*1200/100 = 24. This value is tested by simulation experiment and results are shown (Figure 18 and Figure 19).

In the plot of full boxes in the buffer ("L5P5") (see first plot in Figure 20), it happens to be lower than the safety level (= 2) so there is a risk to achieve zero and a consequent breakdown. Therefore, the number of kanban units must be increased. In the second plot, the safety level is 24 and the minimum number during simulation is 20. This could be a point for discussion, if it is necessary to add more kanban units into the logistic system or accept a more risky behaviour.

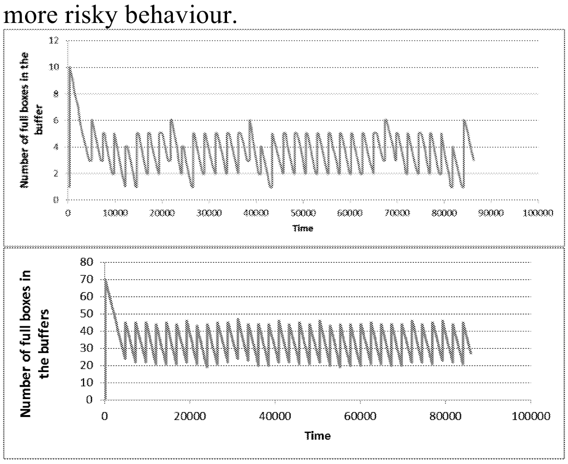


Figure 20 - Model β: Safety level with stochastics

5 CONCLUSIONS

As detailed in the paper body, this project was developed in several stages. Here we resume the outcomes from each of them:

Model α (with full system design) showed us that one MR for each of the 10 lines means an average non-working time around 50% for each MR. This learning motivated the simulation of the system with only 5 MR, which proved to be possible to achieve this big saving.

Model β was developed to include maximum detail in one MR cycle, to extrapolate to the whole system. Here is made the setup of the kanban system for parts supplying.

In Experiments A the number of kanban units was studied, disregarding safety level, and pointed to values above 13 kanban units.

In Experiments B safety level was added, meaning that it must be enough material for 2xMilkRun cycle time in each production point to avoid line stops in case of MR miss/fail. Experiments B raised minimum kanban units in 12, moving the minimum to 13+12=25 kanban units.

Experiments C studied the impact of increasing cycle time, and showed that cycle time over 2000 seconds can compromise the number of MR wagons and thus the MR maneuverability.
Experiments D analyzed the impact of different levels of consuming time of boxes in production line, showing that adjusting kanban values, the MR setup can achieve to make the correct supplying.

In the end of this project we verified that this work could be relevant to some further design of MR systems, once it studied most of the problematic issues on MRs and Kanban supplying.
This study was made on a real system, being forced to include important concerns, not included in the technical model (e.g. strong safety levels)
Using MRs with kanban pushes the model logical complexity to a higher level. IDS helped to deal with it by supporting directly those concepts. The discussion and experiments have been designed in a strategic level, mainly over policy details in decision rather than on simulation model details.

REFERENCES

Costa, B., Dias, L., Oliveira, J.A, & Pereira, G. (2007). Modelos de Simulação do Sistema de Abastecimento de Linhas de Produção. Actas do VIII Congreso Galego de Estatística e Investigación de Operacións (pp. 153-158). doi:ISBN 978-84-690-9136-4

Gross, J. M., & McInnis, K. R. (2003). Kanban made simple: demystifying and applying Toyota's legendary manufacturing process (Vol. 1). New York,: Amacom Books.

Jiang, Z., Huang, Y., & Wang, J. (2010). Routing for the Milk-Run Pickup System in Automobile Parts Supply. Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology (pp. 1267–1275). Springer.

Krieg, G. N. (2005). Kanban-controlled manufacturing systems (p. 236). Springer-Verlag New York Inc.

Satoh, I. (2008). A Formal Approach for Milk-Run Transport Logistics. IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, 91(11), 3261–3268.

Tanchoco, J. M. A. (1994). Material flow systems in manufacturing. Cambridge: Springer.

Taylor, G. (2009). Introduction to logistics engineering. CRC Press.

Vik, P., Dias, Luis, & Oliveira, J. (2007). An Example Of Using 3d Simulation In The Internal Logistic Process. Proceedings of 2nd international conference Výrobní systémy dnes a zítra 2007. Liberec: Technická univerzita v Liberci. doi:ISBN 978-80-7372-295-1

Vik, P., Dias, Luis, & Oliveira., J.A. (2008). Using of 3D Simulation in the Industry. Proceedings of ICTKI – 2nd. International Conference of Technology Knowledge and Information. Ústí nad Labem. doi:ISBN 978-80-7044-969-1

Vik, P., Dias, Luis, Guilherme, P., & Oliveira, J. (2010). Automatic Generation of Computer Models through the Integration of Production Systems Design Software Tools. 3th International Conference on Multidisciplinary Design Optimization and Applications. Paris, France.

Vik, P., Dias, Luis, Pereira, Guilherme, & Oliveira, J. (2010). Improving Production and Internal Logistics Systems - An Integrated Approach Using CAD and Simulation. Proceedings of the 3rd International Conference on Information Systems, Logistics and Supply Chain. Casablanca (Morocco). doi:ISBN 978-2-930294-22-1

http://www.fantozzi.com (2012) Milk Run manufacturer.

http://www.ipaslovakia.sk(2012) Milk Run concept.

http://www.lanner.com (2012) Witness simulation tool.

BIOGRAPHIES

LUÍS S. DIAS was born in 1970 in Vila Nova de Foz Côa, Portugal. He graduated in Computer Science and Systems Engineering at the University of Minho, Portugal. He holds an MSc degree in distributed operative systems, computer communications and advanced architectures, having developed the dissertation work in visual programming languages. He holds a PhD degree in Production and Systems Engineering (Simulation) from the University of Minho, Portugal. He is Assistant Professor at University of Minho. His main research interests are Modelling, Simulation and Optimization.

PAVEL VIK was born in 1982 in Mladá Boleslav, Czech Republic. He studied at the Technical University of Liberec where he obtained his MSc degree in Manufacturing Systems, in 2005. He is making doctoral studies in Manufacturing Systems and Processes and he is attended an ERASMUS-SOCRATES program at the University of Minho. His main research interests are 3D animation and virtual reality in design of production systems and visualization of computer simulation and automatic generating of simulation models.

JOSÉ A. OLIVEIRA was born in 1966 in Matosinhos, Portugal. He studied Mechanical Engineering at the University of Porto, Portugal. He graduated with a Ph.D. in Production and Systems Engineering at the University of Minho, Portugal. His main research interests are Optimization with Heuristic Methods in Systems Engineering.

GUILHERME PEREIRA was born in 1961 in Porto, Portugal. He graduated in Industrial Engineering and Management in the University of Minho, Portugal. He holds an MSc degree in Operational Research and a PhD degree in Manufacturing and Mechanical Engineering from the University of Birmingham, UK. He is Associate Professor at the University of Minho. His main research interests are Operational Research and Simulation.

PRODUCTION LOGISTICS

EVALUATING REPLENISHMENT SYSTEMS UNDER PERTURBED ENVIRONMENTS

S. T. Enns
Chandandeep S. Grewal
Department of Mechanical and Manufacturing Engineering
University of Calgary
Calgary, Alberta, T2N 1N4 Canada
E-mail: enns@ucalgary.ca

KEYWORDS
Simulation-optimization, reorder-point replenishment, performance sensitivity, performance robustness

ABSTRACT

A methodology is introduced to evaluate the expected performance of replenishment systems along with their sensitivity to perturbation of environmental factors. Optimal tradeoff curves of inventory as a function of service levels are generated for a small supply chain using simulation-optimization. The environment is then perturbed and the replenishment systems are compared using an index based on the root-mean-squares of areas under the optimal and perturbed tradeoff curves. Results show that differences in sensitivity, sometimes referred to as robustness, are relatively unimportant when comparing reorder point and Kanban replenishment systems.

INTRODUCTION

Concern with the robustness of supply chains has increased with longer and more complex supply chains, the result of increased globalization. On the other hand, increased competitiveness and the resulting move to lean supply chains has also put pressure on reducing inventory, which can mitigate the effects of disruptions. The recent literature is ambiguous about how to define robustness. Studies can roughly be divided into those dealing with strategic decisions and those dealing with tactical or operational decisions. Since the second category is most relevant to this research, only it will be briefly discussed.

Research dealing with tactical or operational issues is less likely to be concerned with major disruptions. Instead it is concerned with uncertainty in the course of regular operations, such as uncertainty about demand rates, setup times or transportation variability. In these robustness studies the parameters of probability distributions, such as the mean demand rate or variability of demand, are unknown. It is usually of interest to assess the performance of a production or replenishment system under conditions that are different from those anticipated. Since changes in the environment may not be immediately obvious, it is common to evaluate performance without assuming there is any intervention, such as changing the decision variables.

The definition of robustness in such studies relates mostly to the sensitivity of performance to changes in the supply chain environment (Jodlbaur and Huber, 2008, Vieira and Lemos, 2009). Tactical and operational robustness is also closely related to Taguchi's definition of robustness used extensively in product and process development and improvement (Taguchi, *et al*., 1999). In this case the objective is to have the product or process performance to be as insensitive to variation in the environment as possible. Furthermore, Taguchi's signal-to-noise (S/N) ratios consider both the mean performance and the variation in performance due to environmental uncertainty, making these measures useful for overall comparisons.

Studies of robustness in the context of tactical and operational decisions are amenable to the use of structured experiments and building relationships between environmental uncertainty, or variation, and performance. Often simulation is used to examine the effects of perturbed environments on supply chain performance. Tee and Rossetti (2002) present a simulation study with one warehouse and a number of retailers. This is an un-capacitated system with continuous-review reorder point replenishment. Overall, the model indicated that the best decision variables for a known, stationary demand environment may not be the best for an environment subject to unknown, perturbed demand. Shang, *et al*. (2004) present a simulation study with a supply chain consisting of three suppliers, one manufacturer and four retailers. The objective is to maximize performance, given the uncertainty. Taguchi orthogonal arrays and response surface methods are used to find decision variables that will best mitigate the effects of the environmental factors. Jodlbauer and Huber (2008) use of simulation and optimization to create performance tradeoff curves for a six station flowline under MRP, Kanban, CONWIP and Drum-Buffer-Rope (DBR) replenishment. Results indicated that the relative robustness ranking of the replenishment systems is sometimes dependent on the inventory levels chosen.

The research in this paper also considers tactical and operational decision making. It deals with environments that have been perturbed but not to the extent structural changes to the supply chain or any other type of intervention is required. This research is particularly focused on quantitatively evaluating the sensitivity of performance, or robustness, relative to the expected performance in an average environment. The methodology developed is an extension of that found in Grewal, *et al*. (2010), in which optimal tradeoff curves between inventory and customer service performance are generated using simulation-optimization. In that paper the methodology was applied to the comparison of replenishment systems, namely the continuous-review reorder point and single-card Kanban systems.

METHODOLOGY FOR COMPARISON

The methodology relies on simulation and simulation-optimization experimental procedures. The assumption is also made that the most important dimensions of performance are inventory and customer service levels. The methodology is based on three main ideas. First, any comparison of replenishment systems should consider the performance when each system is being run optimally under the expected environmental conditions. Second, comparisons should be made over a range of inventory and customer service level tradeoffs if they are to be generalizable. This can be done by generating optimal performance tradeoff curves. Third, the environment can be perturbed from that expected to create additional performance tradeoff curves, which can then be used to measure the sensitivity of performance to environmental changes. As this research shows, it is then possible to incorporate sensitivity to perturbation into performance comparisons as well.

Generation of optimal tradeoff curves

A single point on an optimal tradeoff curve can be obtained by optimizing decision variables to minimize one performance measure while constraining a second performance measure to a fixed value. If the optimization procedure is repeated with incrementally higher settings of the second performance measure constraint, an optimal tradeoff curve can be generated. In this research the replenishment system decision variables were optimized, using simulation-optimization, to minimize the total inventory levels subject to various customer service level constraints. Figure 1 illustrates conceptual optimal tradeoff curves for two different replenishment systems, 'a' and 'b'.

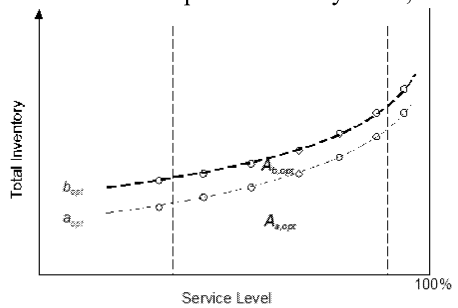


Figure 1: Optimal Tradeoff Curves

Comparison using optimal areas

It is obvious that in Figure 1 replenishment system 'a' performs better than 'b' since inventory is lower for any given service level. The difference in performance can be quantified by comparing the areas under the tradeoff curves. The vertical dashed lines indicate service level bounds that facilitate comparing areas over some comparable portion of both tradeoff curves. The areas associated with the curves for the two replenishment systems are labeled as $A_{a,opt}$ and $A_{b,opt}$. The ratio of the areas may be termed the Mean Performance Index, *MPI*.

$$MPI = \frac{A_{a,opt}}{A_{b,opt}} \tag{1}$$

For example, assume $A_{a,opt}$ = 9.8 and $A_{b,opt}$ = 10.0. *MPI* would then be 0.98, indicating 'a' performs better than 'b'.

Generation of perturbed tradeoff curves

Perturbed tradeoff curves are defined as those generated by changing some environmental factor, while continuing to use the optimal decision variables obtained prior to the perturbation. For example, changing the demand rate would shift, or perturb, a tradeoff curve. Figure 2 illustrates perturbed tradeoff curves where the environmental factor has been changed to perturb the curves both upwards, increasing the area under the curves, and downward, decreasing the areas under the curves. These curves can be easily generated using simulation alone, without re-optimization, by incrementally changing the service level constraint. The amount of shift due to perturbation is an indicator of performance sensitivity to changes in the environmental factor. In Figure 2, system 'a' is shown to be very sensitive with respect to upward shifts of the tradeoff curve. Therefore system 'a' may not necessarily be preferable to system 'b' even though it has better performance under the expected environmental conditions used to optimize the decision variables.

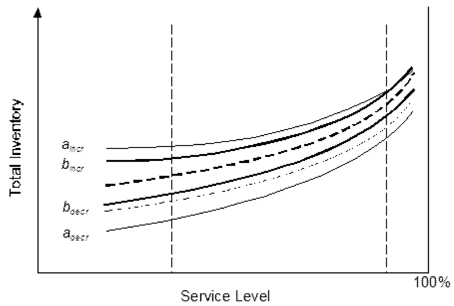


Figure 2: Perturbed Tradeoff Curves

For replenishment system 'a', the difference between the areas under the perturbed tradeoff curve, $A_{a,pert}$, and the optimal tradeoff curve is given by the following.

$$\Delta_{a,pert} = A_{a,pert} - A_{a,opt} \tag{2}$$

If the perturbation results in an increased area, $A_{a,pert}$ is designated as $A_{a,incr}$, whereas if it results in a decrease it is designated as $A_{a,decr}$. For example, if it is assumed that $A_{a,incr}$ = 10.7, then $\Delta_{a,incr}$ = 10.7 – 9.8 = 0.9.

Comparison using perturbed areas

A major purpose of this research is to investigate the relative performance of replenishment systems when the supply chain environment deviates from that predicted. While one system could have better performance under optimal conditions, it might have poorer performance when the environment is perturbed. In this research, root mean squares are used to incorporate performance variation in the measure used for replenishment system comparison. If for replenishment system 'a' the perturbed tradeoff curve areas are designated as $A_{a,decr}$ and $A_{a,incr}$, the Root Mean Square of the Areas, $RMSA_a$, is defined as follows.

$$RMSA_a = \sqrt{\frac{\left(A_{a,decr}^2 + A_{a,incr}^2\right)}{2}} \tag{3}$$

Now consider that the same is done with replenishment system 'b'. The ratio of the root mean squares of the areas can then be used as a measure of comparison. This is referred to as the Total Performance Index, *TPI*:

$$TPI = \frac{RMSA_a}{RMSA_b} \tag{4}$$

This measure incorporates both the expected, or mean, performance and the sensitivity due to uncertainty in the supply chain environment. For example, assume $A_{a,incr}$ and $A_{a,decr}$ are 10.7 and 9.7 respectively, and that $A_{b,incr}$ and $A_{b,decr}$ are 10.1 and 9.9 respectively. The *TPI* would be 1.02, indicating system *'b'* has the best performance. However, this is opposite the previous conclusion where an *MPI* of 0.98 indicated system *'a'* was better. In this example, perturbation has a much larger effect on the performance of system '*a*' than on system '*b*'. This negatively affects the relative performance as measured by *TPI*.

THE EXPERIMENTAL SCENARIO

The experimental supply chain scenario used in this research is represented by Figure 3, although not all details are shown. The customer demand for each product type was assumed to be Poisson, which means the interarrival times between customers are negative exponential. The customer arrival rate for each product type was 20 items per hour in the base case. Each customer was assumed to order only one item of a specific product type. If there was finished goods inventory of the required product type in stock, the customer's order was filled immediately and the finished goods inventory, *FG*, was reduced. Otherwise the customer backorders, *BO*, were incremented. There were assumed to be no lost sales.

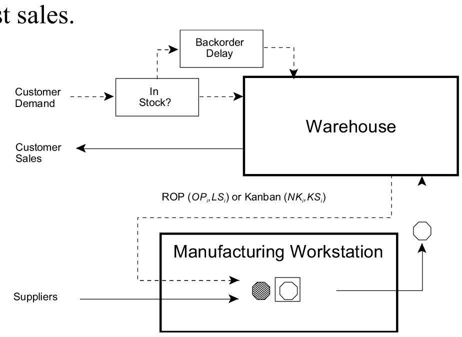


Figure 3: A Capacitated Supply Chain Scenari**o**

Two types of warehouse replenishment systems were considered. These were a reorder point (ROP) and a Kanban (Kbn) system. The reorder point system was a continuous-review system with decision variables for each product type *i* being the reorder point, OP_i, and lot size, LS_i. A new order was released to the manufacturer whenever the inventory position for any product type fell to its reorder point. The inventory position included the current finished goods inventory plus any released orders from the warehouse not yet filled, minus the customer backorders. The Kanban system was assumed to be a single card system. The decision variables for each product type *i* were the number of Kanban cards, NK_i, and the Kanban quantity or size, KS_i. In this research the number of Kanban cards was considered to be equivalent to the number of Kanban containers in circulation.

Once an order was released, it was sent to the manufacturing workstation. In the case of the Kanban system, the order had to wait for a transporter, assumed to be released at fixed intervals of 4 hours. In the case of the reorder point system, there was a uniformly distributed order processing delay with parameters (0, 4) hours. The upstream order transit time was assumed to follow a triangular distribution with parameters (4, 8, 12) hours for both systems.

Once the order arrived at the manufacturer it immediately joined a queue at the workstation. The orders in queue were processed in first-come-first-served (FCFS) priority. The setup time was assumed to follow a gamma distribution with a mean of 0.25 and standard deviation of 0.125 hours. Units in the lot were processed sequentially with each unit having a deterministic processing time of 0.015 hours. These values were the same for both product types. The total time at the workstation was therefore the setup time plus the processing time per unit multiplied by the lot size.

Once a complete lot size was processed it had to wait for shipment to the warehouse. Transporters were released at fixed intervals of 4 hours. The transit time to the warehouse followed a triangular distribution with parameters (4, 8, 12) hours. Once the shipment was received at the warehouse the finished goods inventory, *FG*, was adjusted, taking into account immediately filling any existing backorders, *BO*.

The optimization problem was to find decision variables to minimize the total inventory, *TI*, for each replenishment system, *j*, where *j* = 1 for the reorder point (ROP) system and *j* = 2 for the Kanban (Kbn) system. This minimization was subject to the constraint that the actual customer service level for each product type *i*, SL_i, must be greater than or equal to a target customer service level, β_p.

$$\text{Min. } TI_j\left(OP_{i,j=1}, LS_{i,j=1}, NK_{i,j=2}, KS_{i,j=2}\right) = \frac{1}{(t_e - t_s)}\sum_{i=1}^{n}\left\{\int_{t=t_s}^{t_e}\left(WIP_{i,t} + FG_{i,t}\right)dt\right\} \quad (5)$$

subject to:

$$SL_i\left(OP_{i,j=1}, LS_{i,j=1}, NK_{i,j=2}, KS_{i,j=2}\right) \geq SL_p$$

$OP_{i,j=1}, LS_{i,j=1}, NK_{i,j=2}, KS_{i,j=2}$ are integers

where,

- i - product type
- j - replenishment system
- n - number of product types
- t - current time
- TI - total time-averaged inventory
- $WIP_{i,t}$ - work-in-process of product type *i* at *t*
- $FG_{i,t}$ - finished-goods for product type *i* at *t*
- t_s - start of simulation data collection
- t_e - end of simulation data collection
- SL_i - actual service level for product type *i*
- SL_p - target service level
- OP_i - reorder point for product type *i* when *j* = 1
- LS_i - lot size for product type *i* when *j* = 1
- NK_i - Kanbans for product type *i* when *j* = 2
- KS_i - Kanban size for product type *i* when *j* = 2

The performance measure, *TI*, was the total inventory count in the replenishment loop. Inventory was added to the count at the time an order from the warehouse joined the queue at the manufacturing workstation. The customer service levels, SL_i, were measured in terms of the proportion of customer orders not backordered for each product type *i*. This is also the proportion of orders filled immediately from stock, or the fill rate.

Decision variables to minimize inventory were found using simulation-optimization. A general-purpose heuristic optimization engine, *OptQuest*®, was interfaced with a general-purpose simulation engine, *Arena*® (Kelton, *et al.* (2007). The simulation component provides performance outputs for a given set of input decision variables. The optimization component uses the outputs from previous simulation runs to determine a new set of simulation input decision variables. This iterative simulation and optimization process will search for decision variables that maximize or minimize the objective function. The procedure *OptQuest*®

uses relies on a combination of scatter search, tabu search and neural networks. Further details on the simulation-optimization procedure are available in Grewal, *et al*. (2010).

THE EXPERIMENTAL DESIGNS

The first stage of experimentation required the generation of the optimal tradeoff curves. Separate sets of experiments were run for the reorder point and Kanban systems. For each replenishment system, simulation-optimization runs were used to determine optimal decision variables at target service levels, SL_p, of 75%, 80%, 85%, 90%, 95% and 99% customer orders filled from stock. A full factorial set of simulation-optimization experiments results in 12 combinations of settings. Each simulation-optimization experiment was run until the optimal, or approximately optimal, decision variables were found. Simulation runs within each experiment used different combinations of decision variables, as determined by *OptQuest®*. The simulation runs within each experiment used a warm-up period of 8784 hours, equivalent to one seasonal cycle, to reach steady-state conditions. Data in each simulation run was then collected over a period of 87840 hours, equivalent of ten seasonal cycles. Each simulation run within the simulation optimization experiment was replicated three times. The number of simulation runs used to find the optimal decision variables in each experiment was set to 2500 in *OptQuest®*.

The second stage of experimentation required the generation of the perturbed tradeoff curves. Two environmental factors were selected for perturbation. The first perturbation factor, downstream transit time variability, was chosen to be one which would change the amount of variability in the replenishment system. A triangular distribution with parameters (4, 8, 12) hours was used as the base case. The range of this distribution was changed by one of the six levels shown in Table 1. For example, with a -60% change in the range, the parameters were adjusted to (6.4, 8, 9.6) hours. The second perturbation factor was the deviation in the mean demand rate from the expected value. This factor was chosen as one which would change the load on the system, as opposed to the variability. The demand rate for the base case was 20 items per hour for each product type. The perturbed demand rate adjustments are also shown in Table 1. For example a setting of -3% means a perturbed demand rate of 19.4 items per hour for each product.

Table 1: Factors for Perturbed Experiments

Factor	Levels
Replenishment system	Kanban, ROP
Service level, *SL*	75%, 80%, 85%, 90%, 95%, 99%
Transit time variability	-60%, -40%, -20%, +20%, +40%, +60%
Demand rate deviation	-3%, -2%, -1%, +1%, +2%, +3%

All 12 combinations of the replenishment system and service levels were run against each perturbed setting of the other two factors. This resulted in 144 combinations of settings. The decision variables used in these experiments were the optimal ones determined without perturbation, for the given replenishment system and service level. The simulation data collection procedures were the same as those discussed for the simulation-optimization experiments. Five replications were run for each of the 144 combinations of experimental settings. Within-group variance was very low. In creating perturbed tradeoff curves, the results of the five replications were averaged. Since six service levels, *SL*, are used to develop one tradeoff curve, the total number of perturbed tradeoff curves represented by this experimental design is 24.

THE EXPERIMENTAL RESULTS AND ANALYSIS

The optimal tradeoff curves are plotted in Figure 4. The values beside the points along the curve are the average observed service levels, as a percent, and the average total inventory counts in the system. The areas under these tradeoff curves were calculated using lower and upper service level, *SL*, bounds of 87% and 97%. The areas were 6070.59 and 6791.68 for the reorder point (ROP) and Kanban (Kbn) systems respectively. The Mean Performance Index (*MPI*), as given in Equation (1), is as follows.

$$MPI = \frac{A_{ROP,opt}}{A_{Kbn,opt}} = 0.8938 \qquad (6)$$

The performance of the reorder point system is better than that of the Kanban system under optimal decision variable settings. This is consistent with pervious findings and is largely due to the ability of reorder point systems to consider backorder information in the replenishment logic.

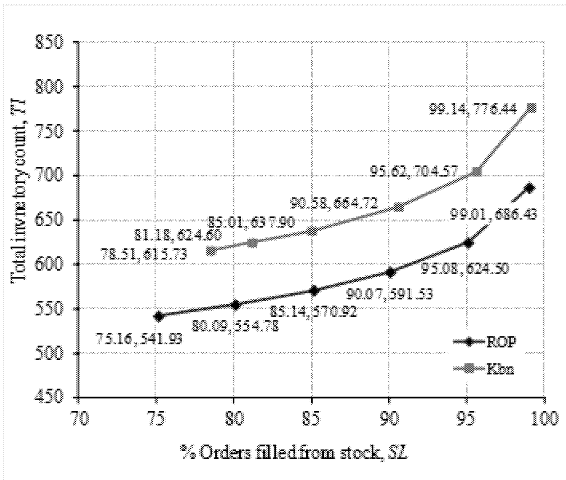


Figure 4: Optimal Tradeoff Curves

The areas under the perturbed tradeoff curves, calculated between the 87% and 97% service levels, are shown in Table 2. The columns for the mean areas are the averages across the five replications for each tradeoff curve. The standard deviations of the areas across the five replications were less than 0.2% in all cases. The columns marked 'Δ_{opt} Area' show the difference in the areas under the tradeoff curves when compared to the optimal areas before perturbation.

Table 2: Areas under Optimal Tradeoff Curves

Factor	Perturbation	ROP Area		Kanban Area	
		Mean	Δ_{pert} Area	Mean	Δ_{pert} Area
Transit Time Variability	Hi_decr (-60%)	5796.30	-274.29	6550.65	-241.03
	Med_decr (-40%)	5872.97	-197.62	6616.45	-175.23
	Lo_decr (-20%)	5969.67	-100.92	6701.73	-89.95
	Lo_incr (+20%)	6167.53	96.94	6888.71	97.03
	Med_incr (+40%)	6260.09	189.50	6990.68	199.00
	Hi_incr (+60%)	6346.21	275.62	7079.61	287.93
Demand Rate	Hi_decr (-3%)	5858.97	-211.62	6539.31	-252.37
	Med_decr (-2%)	5925.06	-145.53	6619.60	-172.08
	Lo_decr (-1%)	5992.91	-77.68	6702.63	-89.05
	Lo_incr (+1%)	6149.88	79.29	6888.36	96.68
	Med_incr (+2%)	6231.03	160.44	6990.99	199.31
	Hi_incr (+3%)	6311.06	240.47	7085.20	293.34

The root mean squares of the areas, *RMSA*, were calculated by pairing low, medium or high percent decreases in transit time variability or demand rates with the equivalent percent increases. For example, when the transit time variability, TTV, is adjusted down and up by 20% the *RMSA* values, using Equation (3), were as follows.

$$RMSA_{ROP,TTV\_Lo} = \sqrt{\frac{\left(A_{Lo\_decr}^2 + A_{Lo\_incr}^2\right)}{2}} = \sqrt{\frac{(5969.67^2 + 6167.53^2)}{2}} = 6069.41 \quad (7)$$

Similar calculations for the Kanban system resulted in $RMSA_{Kbn,TTV\_Lo}$ = 6795.86. Therefore the Total Performance Index, *TPI*, comparing these systems under low transit time variability deviations can be stated as follows:

$$TPI_{TTV_{Lo}} = \frac{RMSA_{ROP,TTV\_Lo}}{RMSA_{Kbn,TTV\_Lo}} = \frac{6069.41}{6795.86} = 0.8931 \quad (8)$$

Similar *RMSA* and *TPI* values were also calculated for the other perturbed tradeoff curve combinations and are given in Table 3.

Table 3: Total Performance Indices, *TPI*

	Perturbation	$RMSA_{ROP}$	$RMSA_{Kbn}$	*TPI*
Transit Time Variability	Lo (-20%, +20%)	6069.41	6795.86	.8931
	Med (-40%, +40%)	6078.73	6806.14	.8941
	Hi (-60%, +60%)	6077.48	6820.26	.8911
Demand Rate	Lo (-1%, +2%)	6017.90	6796.13	.8934
	Med (-2%, +3%)	6079.97	6807.83	.8931
	Hi (-4%, +4%)	6089.21	6871.72	.8931

The results show there is almost no difference in the *TPI* indices for different levels of perturbation. The *TPI* values are also almost the same for transit time variability perturbations as for demand rate perturbations. Furthermore, the ratios are very close to the *MPI* value of 0.8938 computed using the areas under the optimal tradeoff curves.

These results show the relative performance of these replenishment systems is almost the same whether or not the supply chain environment is perturbed. In other words, neither system shows lower performance sensitivity to perturbation and therefore cannot be considered more robust. This is important in that robustness need not be major concern when considering the relative performance of reorder point and Kanban replenishment systems.

If replenishment decision variables are set optimally for the expected environmental conditions, this research indicates comparing replenishment systems on expected performance under these conditions is adequate. However, it is not clear these conclusions can be generalized if the decision variables used are not near optimal for the environmental conditions anticipated. In such cases perturbation may lead to different performance behavior.

CONCLUSIONS

This research has explored the problem of measuring and comparing replenishment systems while taking into account both expected performance and sensitivity to uncertainty in the environment. A methodology based on measuring the effect of perturbation away from an optimal performance tradeoff curve has been developed. A Total Performance Index, *TPI*, works well for comparison. This metric is based on measuring the root mean square of areas under perturbed tradeoff curves.

Results comparing continuous-review reorder point and single-card Kanban replenishment systems indicate there is little difference in the sensitivity of these two systems to changes in the supply chain environment. Perturbation of transit time variability, which affects replenishment time variability, and customer demand rates, which affects system loads, both result in much the same change in behavior for both replenishment systems. The practical implication is that differences in the "robustness" performance of these systems are relatively minor compared with differences in performance under expected, or average, conditions.

Sensitivity of performance to changes in environmental conditions has still received relatively little attention in the supply chain literature. Alternative metrics and methodologies need to be further explored and the scope of supply chain environments studied needs to be broadened. A particularly difficult issue is that of optimizing decision variables given that there is future uncertainty in the environment. In this research optimization was based on the expected conditions, not a range of possible future environments.

REFERENCES

Grewal, C.; P. Rogers; and S.T. Enns. 2010. "Performance evaluation of inventory replenishment strategies in a capacitated supply chain under optimal parameter settings." *International Journal of Value Chain Management*, 4(3), 195-212.

Kelton, W.D.; R.P. Sadowski; and D.T. Sturrock. 2007. *Simulation with Arena*. 4th edition, McGraw-Hill, NY.

Jodlbauer, H. and A. Huber. 2008. "Service-level performance of MRP, Kanban, CONWIP and DBR due to parameter stability and environmental robustness." *International Journal of Production Research*, 46(8), 2179-2195.

Shang, J.S; S. Li; and P. Tadikamalla. 2004. "Operational design of a supply chain system using Taguchi method, response surface methodology, simulation, and optimization." *Int. Journal of Production Research*, 42(18), 3823-3849.

Taguchi, G.; S. Chowdhury and S. Taguchi. 1999. *Robust Engineering*, McGraw-Hill, New York, NY.

Tee, Y.S. and M.D. Rossetti. 2002. "A robustness study of a multi-echelon inventory model via simulation." *Int. Journal of Production Economics*, 80, 265-277.

Vieira, G.E. and R. Lemos. 2009. "Understanding supply chain robustness." *Proceedings of IEEE/INFORMS International Conference on Service Operations and Logistics and Informatics (SOLI)*, Chicago, 157-162.

BIOGRAPHIES

Silvanus T. Enns is an Associate Professor in the Schulich School of Engineering at the University of Calgary.

Chandandeep S. Grewal received his PhD in 2012 and is now a supply chain analyst with SMART Technologies.

THE REDESIGN OF THE PROCESS CONTROL OF CONCRETE SYSTEMS LTD PTY FOR A MORE EFFECTIVE AND EFFICIENT PRODUCTION PROCESS

M.P.W. van der Velden, H.P.M. Veeke, G. Lodewijks
Faculty of Mechanical, Maritime and Materials Engineering
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
E-mail: G.Lodewijks@tudelft.nl

KEYWORDS
Process control, PROPER model, discrete simulation

ABSTRACT

This article describes a research project concerning a case study performed at Concrets Systems Ltd. Pty. (CS) Analysis of the problem situation at CS showed that the day planning of the production process was never realized, the production capacity was not used efficiently and a high amount of waste concrete was created per day. The research goal was therefore to redesign the process control in order to gain a controlled production process, such that the production capacity is used efficiently and that products are used effectively. The redesign should be able to realize the day planning and to minimize waste concrete. The redesign was defined according to the steps of the innovation model as a design process [In't Veld. 2002] and was validated with a simulation model. Experiments with the simulation model showed that the redesign will lead to a better controlled production process and is able to fulfill the research goal.

INTRODUCTION

Concrete Systems Ltd. Pty. (CS) is a manufacturer of customer specific prefab concrete systems. The variety of products are high and the volumes of products are low. Everything at CS is made project based. The wish of CS to use the production capacity efficiently and the products effectively is not being fulfilled. It was noticed that agreed production data, milestones and sometimes even delivery times are not being realized. It was also noticed that production capacity is not being used proportional, which gives large fluctuations in capacity usage. Besides CS has a high amount of waste concrete.

Analysis of the problem situation using the Black box method [In't Veld. 2002] of the systems approach has found the following problems at CS:

- Projects are not controlled on time and within budget;
- The production planning is not reached;
- Waste concrete occurs.

Projects are not controlled on time and within budget because standards are neglected, which leads to the absence of a work budget and project planning. Neglecting the standards will currently not result in an immediate consequence for the employee. The production planning is not reached because of disturbances. Also the function structure of the production process does not allow for dealing with disturbances. Besides the entrance check of materials is not functioning well. Currently, the efficiency of the employees in the production process is on average 66%. The average waste concrete is currently 3,6% of the mixed concrete. This is caused by a lack of tuning between the production tasks and by ordering the wrong or too much concrete.

To gain a controlled production process the process control at the product flow must be redesigned. Therefore the research goal of the project was to gain a controlled production process, such that the production capacity is used efficiently and products are used effectively in order to realize the day planning and to minimize waste concrete.

REDESIGN OF THE PROCESS CONTROL

The redesign was defined according to the steps of the innovation model for a design process [In't Veld. 2002]:

1. Determine the objectives of the redesign;
2. Define several alternatives;
3. Confront and tune;
4. Develop and organize;

Objectives Of The Redesign

The redesign consists of the functions planning of the day production, doing an entrance check, executing the various production tasks, mixing concrete, determining the pouring place and using the personnel. Several alternatives are defined for these functions. These alternatives are compared to each other using the following criteria:

- Minimizing the amount of waste concrete;
- Realizing the day planning;
- Using employees efficiently;
- Using products effectively;
- Having a low disturbance sensitivity;
- Being possible to implement.

The priorities of these criteria are determined according to the priority theory of Saaty [Lootsma. 1978].

Functional Structure Of The Redesign

The redesign was made for the product flow. The redesign checks all incoming products for quality and quantity. In case of a quality or quantity disturbance in the incoming goods feed forward is used to intervene in the production planning. Using this type of control should allow to measure a disturbance and to determine the compensatory intervention [In't Veld. 2002]. In the redesign the production is planned if all needed products are in stock. Figure 1 shows the functional structure of the redesign in a function model.

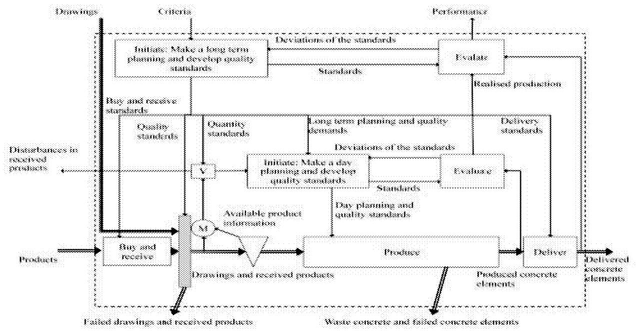


Figure 1. Functional structure of the product flow in the redesign at CS

Within the production process tuning between the production tasks is gathered by using a dynamic sequence of production tasks. This is done by using the following steps:

1. Predetermine the production tasks and the sequence of these tasks for producing an concrete element;
2. Predetermine how the throughput times;
3. Predetermine the earliest and latest start and finish times;
4. Execute these tasks and control the production process using these times. [Wild. 2003]

The earliest start and finish times and the latest start and finish times can be used to calculate the probability of successfully meeting the schedule [Kerzner. 2000]. The earliest start time of an activity (EST_n) equals the earliest finish time of the previous activity (EFT_n-1), such that:

$$EST_n = EET_{n-1}. \quad (1)$$

The earliest finish time of an activity (EFT_n) is calculated from the earliest start time (ESTn) and the throughput time of the activity(t_n), as is

$$EST_n = EST_n + t_n. \quad (2)$$

The latest start time of an activity (LST_n) is calculated from the latest finish time (LFT_n) and the throughput time (t_n), as in

$$LST_n = LFT_n - t_n \quad (3)$$

The latest finish time of an activity (LFT_n) equals the latest start time of the next activity (LST_{n+1}), such that:

$$LFT_n = LST_{n+1}. \quad (4)$$

Disturbances can be compensated by using feed forward [In't Veld. 2002]. This is shown in a steady state model in figure 2.

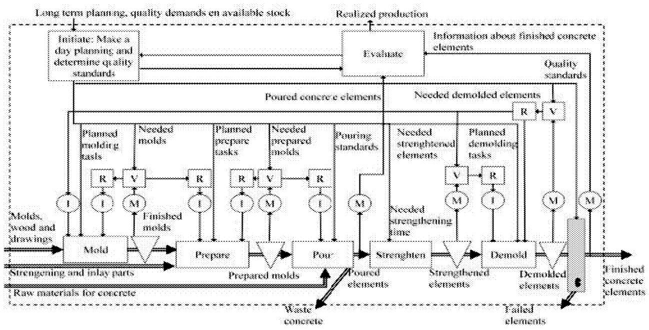


Figure 2. Steady state model of the redesign of the production process at CS

If a disturbance occurs then the execution time can ke shortend by using an extra employee for that task or to stop the task and to reschedule it. Repetition of the disturbance can be prevented by using feedback [In't Veld. 2002].

Personnel Structure Of The Redesign

The redesign uses employee groups. These groups are defined by the dependency of the tasks, which should be done by its members. Within these groups the internal control capacity should increase. This should lead to a better tuning between the tasks [Sitter et al. 2000].

In the case of CS the redesign uses two kinds of employee groups, a production team and a pouring team. The production team consists of carpenters and production employees. A pouring team consists of a concrete driver and a pouring employee. Figure 3 gives the results of the redesign for the personnel structure.

Production teams are responsible for molding, preparing and demolding of the elements. A production team exists of carpenters and production employees which are lead by a team leader. The production chef is responsible for external tuning between the production teams, the pouring teams and the planned production [Sitter et al. 2000]. Team leaders are responsible for realizing the planning by internal tuning of the activities. The team leader evaluates the daily planning with the production chef and the production planner. Based on the realized production, the available products and the long term planning, a planning is made for the next day. The team leader determines the earliest and latest start and finish times for the tasks in the day planning and controls with these times the production. If a disturbance occurs, the team leader should take some actions. A pouring team is responsible for transporting the concrete, pouring and finishing the elements. A pouring team exists of concrete drivers and pouring employees and are lead by the mixing plant.

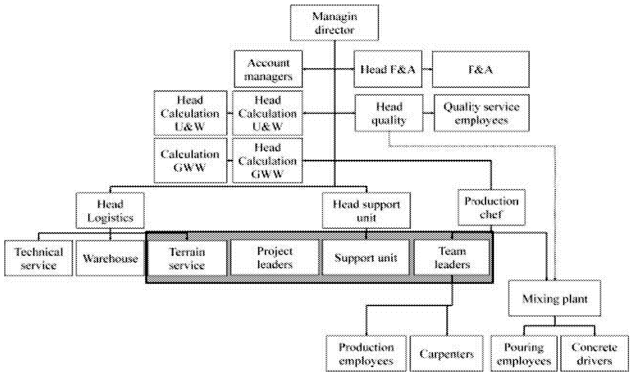


Figure 3. Personnel structure of the redesign at CS

A pouring team is responsible for transporting the concrete, pouring and finishing the elements. A pouring team exists of concrete drivers and pouring employees and are lead by the mixing plant. The mixing plant is responsible for mixing and delivery of concrete. Therefore the mixing plant determines the amount of the mixed concrete and the mixtures of the concrete. The mixing plant should mix all concrete of the same mixture after each other as much as possible. The mixing plant must have a list of the mixtures and the amount of concrete per element to determine the amount of the concrete, the mixtures of the concrete and the sequence in which it is mixed.

The production chef and the production planner discuss weekly the production planning three weeks in front of the real production. During this meeting it is determined which capacity is needed and if the planning can be realized. Using the results of this meeting the production chef must plan its capacity three weeks in front. Besides, the production chef must evaluate the realized production daily with the team leaders and the planner. The production chef must also discuss daily with the team leaders and the mixing plant the sequence in which the elements are poured.

The planner is responsible for making a long term planning and determining the day planning. The production planner receives from the work planner a sequence in which the elements of a project must be produced. The production is planned from this sequence. The planner checks daily the available stock and evaluates with the team leaders and the head production the realized production. Based on the available stock and the realized production the production planner makes a new day planning an discusses this with the team leaders and the head production. 68

VALIDATION METHOD OF THE REDESIGN

The redesign is validated by a simulation model in Delphi using Tomas, which is a tool for object-oriented modeling and simulation [Veeke & Ottjes. 2000]. The performance of the redesign is tested to determine whether the redesign is suitable to realize the day planning, to use production capacity efficiently and to use the products effectively even if disturbances occur.

Simulation Model

The simulation model consists of the whole production process and is shown in the PROPER model [Veeke et al. 2008] in figure 4. The simulation model uses carpenters, production employees, the mixing plant, concrete drivers, pouring employees and nine production halls with a varying amount of production position. Orders are created using the OrderGenerator and planned in a long term planning. Products are then created via the PartOrderProcess and employees are created and planned via the EmployeePlanner. Tasks are released by the JobScheduler. Jobs are consisting of the tasks: molding, preparing, pouring, strengthening and demolding. Finished elements and waste concrete leave the model. The simulation model needs actual order information, which consists of :

- The planned amount of orders over a long term;
- The planned amount of man hour per task;
- The amount of jobs of the planned orders at the same position;
- The delivery dates of the planned job;
- The needed amount of concrete per job;
- The color of the jobs.
- The product type of the order;
- The hall in which the jobs of the orders are produced.

Since all these data varies per job these are delivered to the simulation model as distributions. Disturbances will be added to the simulation to test the robustness of the system. Hereby the following performances are measured:

- Effectiveness: The percentage waste concrete in relation to the amount of mixed concrete;
- Effectiveness: The realized production in relation to the planned production;
- Efficiency: The needed amount of man hour in

relation to the hired amount of man hour.

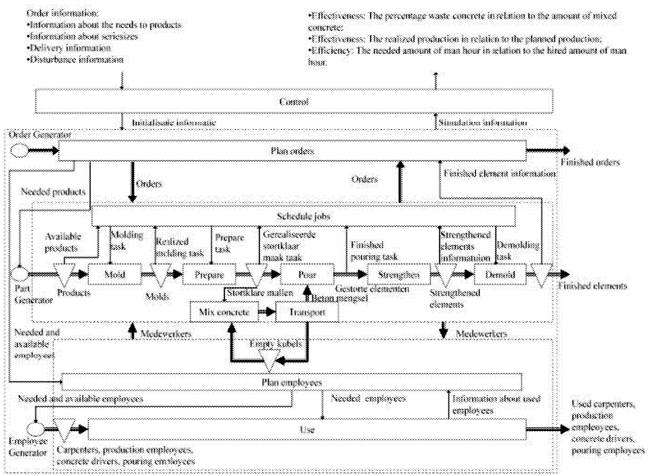


Figure 4: Simulation model

Experiments

A reference model is made to compare the current situation with the redesign. The capacity is determined by a run with infinite capacity. Using the simulation model the performance of the redesign is tested with respect to the reference model. Hereby the following experiment have been executed:

• No disturbances;
• Disturbances in execution times;
• Disturbances in the delivery of bought products.

To get a reliable result all these experiments are done for the runtimes of 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900 and 2000 hour. The execution times are disturbed by sampling a disturbance from a normal distribution and add the disturbance tot the planned execution time. Hereby two experiments are done. In the first experiment a normal distribution is used with a mean (μ) of μ = 0 and a variance (σ) of σ = 1 hour. In the second experiment a normal distribution is used with μ = 0 and σ = 1 hour. The delivery of the bought products is also disturbed by sampling a disturbance from a normal distribution and add it to the delivery time of the bought products. Hereby two experiments are done. In the first experiment a normal distribution is used with μ = 0 and σ = 2,5 days. In the second experiment a normal distribution is used with μ = 0 and σ = 5 days.

RESULTS

This section shows the results of the simulation experiments. First the results with respect to the amount of waste concrete will be shown. Then the results with respect to the efficiency will be shown. Finally the results with respect to delayed products will be shown.

Results With Respect To The Amount Of Waste Concrete

The results with respect to the amount of waste concrete are shown in a box plot in figure 5. The results of the simulation shows that the percentage waste concrete for the redesign is less than the percentage waste concrete for the theoretical current situation.

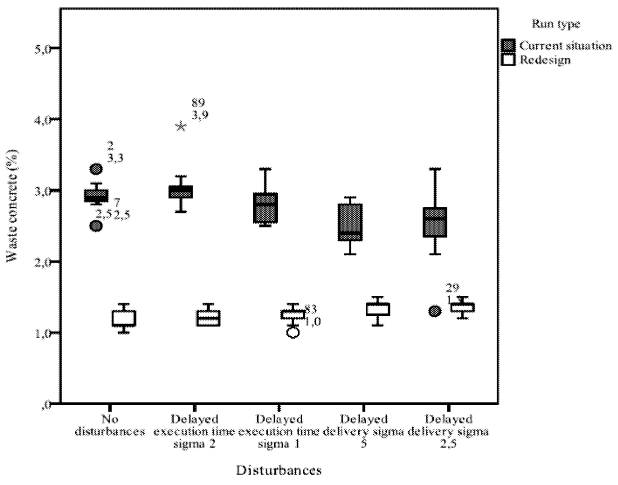


Figure 5. Results waste concrete

Results With Respect To The Efficiency

The results of the simulation for the efficiency of the carpenters are shown in a box plot in figure 6. The efficiency for the carpenters in the redesign is higher than the efficiency of the carpenters in the theoretical model if the delivery of the products is delayed. To determine the difference of the other tests an ANOVA Test is performed. If no disturbances occur there is a difference between the efficiency of the carpenter in the redesign and the efficiency of the carpenters in the current situation with a significance of P = 0.004. If the execution times are delayed by σ = 1 there is no significant difference with P = 0.537. If the execution times are delayed by σ = 2 there is a difference between the efficiency of the carpenter in the redesign and the efficiency of the carpenters in the current situation with a significance of P = 0.027. Figure 7 shows the results of the simulation for efficiency of the production employees in a box plot. This figure shows that the efficiency of the production employees is higher in the redesign than in the theoretical model of the current situation for all executed experiments. Figure 8. shows the results of the simulation for the efficiency of the concrete drivers. This figure shows that the efficiency of the concrete drivers is higher in the redesign than in the theoretical model of the current situation for all executed experiments.

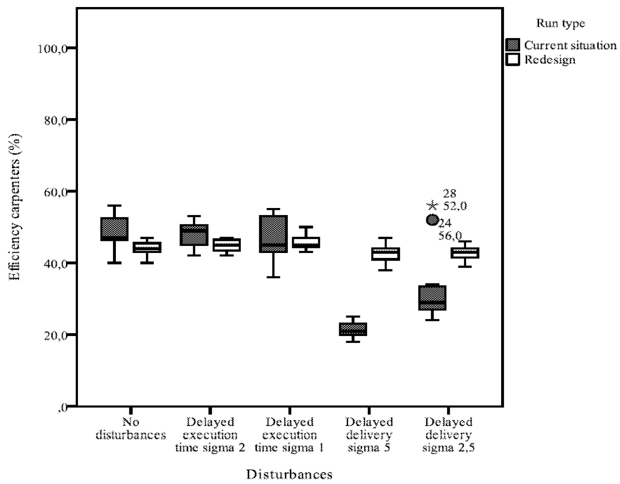


Figure 6. Results efficiency carpenters

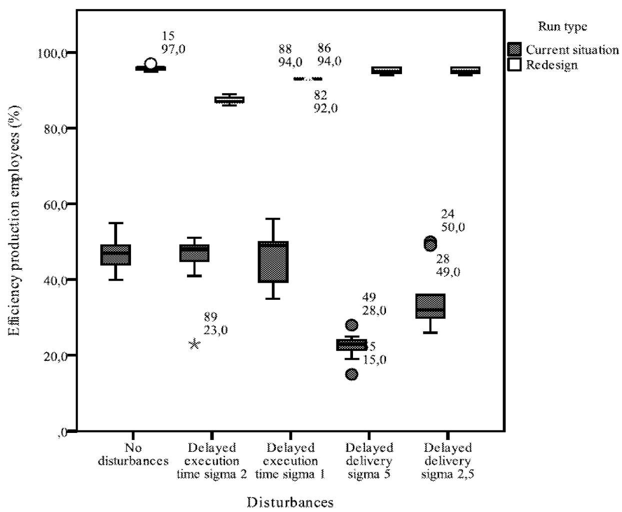


Figure 7. Results efficiency production employees

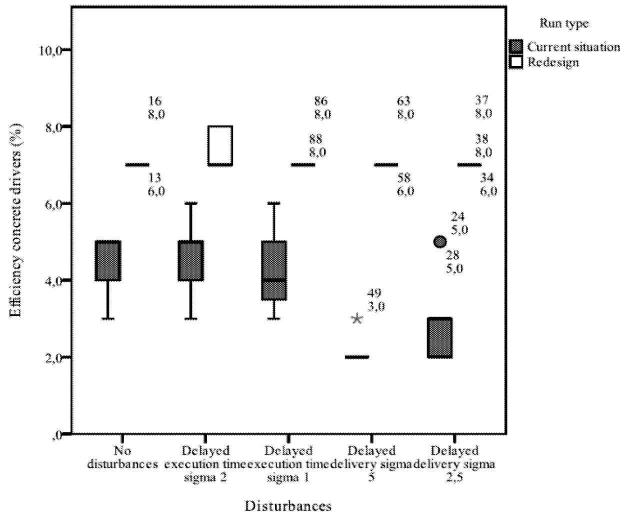


Figure 8. Results efficiency concrete drivers

Results With Respect To Delays

Figure 9 shows the results of the simulation for the percentage delayed element with respect to the original planning. This figure shows that the percentage of delayed elements is lower in the redesign than in the theoretical model of the current situation for all executed experiments.

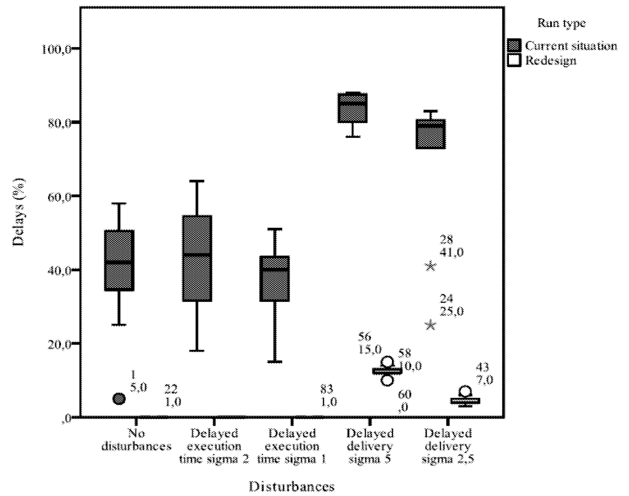


Figure 9. Results delays

DISCUSSION

The results of the simulation show that applying the redesign will result in a decrease of waste concrete by almost 50% with respect to the theoretical model. Besides, the efficiency of the production employees and the concrete drivers in the redesign will increase with respect to the theoretical model. In case of the production employees efficiencies between the 80% and 97% were found, which are high efficiencies. Although the efficiency of the concrete drivers increases by implementing the redesign, they are still low. This can be explained by the fact that concrete drivers have a lot of waiting time while the concrete is mixed and that they have cleaning tasks, which are not taken into account in the simulation.

The efficiencies of the carpenters in the redesign are higher than the efficiencies of the carpenters in the theoretical model of the current situation if the delivery of the products is delayed, while they are lower if the execution times are heavily disturbed or if there are no disturbances. Besides the percentage of delays decreases for all experiments. This implicates an overcapacity of carpenters in the redesign. The redesign will probably show higher efficiencies if less carpenters are used. The redesign realizes the day planning better than the theoretical model of the current situation. The production process of the redesign is therefore better controlled than the production process of the current situation. Only delayed product delivery will lead to a delay in the production. This problem can be solved by adding some extra slack between the planned received date and the start production date [Slack et al. 2007].

CONCLUSIONS

The research goal was to gain a controlled production process at CS, such that production capacity is used efficiently and products are used effectively in order to realize the day planning and to minimize waste concrete. The redesign gives a better controlled production process and is able to fulfill the research goal. Implementing the redesign will lead to:

- A decrease in waste concrete of 50%;
- An increase in efficiency of the production employees and the concrete drivers;
- The need of less carpenters;
- A decrease in delayed elements.

The methods used during the design process can also be used to solve planning and control problems in other project based companies.

REFERENCES

Kerzner, H., *Project Management*, 10th ed. New Jersey, Verenigde Staten : John Wiley & Sons, Inc., 2000.

Lootsma, F.A., *Saaty's priority theory and the nomination of a Senior Professor in Operations Research*, 21 December 1978.

Sitter, L.U. de Sitter, Naber, J.L.G., and Verschuur, F.O., *Synergetisch produceren human resources mobilisation in de productie: een inleiding in structuurbouw*, 3rd ed. Assen: Van Gorcum, 2000.

Slack, N., Chambers, S., and JohnSton, R., *Operations Management*, 5th ed. Harlow: Pearson Education Limited, 2007.

Veeke, H.P.M., Ottjes, J.A., *Tomas: Tool for object-oriented modeling and simulation*, in Proceedings of the Business and Industry Simulation Symposiom, Washington D.C., 2000.

Veeke, H.P.M., Ottjes, J.A., and Lodewijks, G., *The Delft Systems Approach Analysis and Design of Industrial Systems*. Londen: Springer- Verlag London Limited, 2008.

Veld, J. In 't, *Analyse van organisatieproblemen een toepassing van denken in systemen*. Houten: Wolters-Noordhoff bv, 2002.

Wild, R., *Operations Management*, 6th ed. Londen: Thomson Learning, 2003.

AN IMPROVED LARGEST GAP ROUTING HEURISTIC FOR ORDER PICKING

Jinping Liu
Transportation Management College
Dalian Maritime University
1 Linghai Road, Dalian City, 116026 China
liujp@dlmu.edu.cn

Veronique Limère
Hendrik Van Landeghem
Department of Industrial Management
Ghent University
Technologiepark 903, 9052 Zwijnaarde,
Belgium

KEYWORDS
Improved Largest Gap Routing, Largest Gap Routing, Order Picking, Simulation.

ABSTRACT

The largest-gap policy is a routing heuristic for order picking systems. In this paper we develop an improved largest gap routing method. A simulation approach is used to demonstrate the superior performance of the improved largest gap routing over traditional largest gap. Moreover, this paper tests the performance impact of storage assignment rules on largest gap routings. Several scenarios with various order sizes and different item popularity proportions are tested. Monte-Carlo simulation is used to carry out the experiments. The numerical results from the computational analysis show that our improved largest gap routing always outperforms the traditional largest gap routing, i.e. for all order sizes. The effect is the most distinct when the order size is smaller. Finally the study demonstrates that the optimal storage assignment rule to be combined with largest gap routing is within-aisle storage.

INTRODUCTION

Order picking is a highly labour-intensive and costly operation. Of all warehouse operation costs, order picking costs account for 65% (Coyle et al. 2002). Order picking activities are diverse, ranging from administration activities to preparing, searching, extracting and packing goods and walking. In a manual order picking system, this latter activity – walking – amounts to 50% of the total order picking operation time (Tompkins et al., 2003). Thus, the reduction of walking times can help improve picking efficiency.

In order to reduce walking times, it is important to carefully select a good routing strategy. Routing strategies deal with the route of a picker for a picking tour. More specifically, through the use of well-defined rules, the exact sequence in which items are to be picked during a picking tour is determined (Petersen 1997). Although, for a rectangular warehouse, an optimal routing algorithm exists (Ratliff and Rosenthal 1983), in practice heuristic strategies are often applied because of their simple use. Largest gap routing is one type of heuristic strategy. Although largest gap routing is not as easy as “S” routing, which is based on a transversal strategy, it is still used in some warehouses due to the shorter walking distances that are often obtained.
A large number of studies exist that focus on order picking routing policies (Petersen1997, Petersen 1999, Petersen and Schmenner 1999, Petersen and Aase 2004, Petersen et al. 2005, Caron et al. 1998, Roodbergen and De Koster 2001).

Aside from routing, a good storage assignment is another aspect that has an influence on picking efficiency, put-away productivity and space utilization. Two main storage policies are available: random and class-based. By a class-based storage rule, items are classified into several groups based on a specific criteria – e.g. volume, popularity, product group, weight – and each group is then assigned to a dedicated area of the warehouse. Volume-based storage means that items are classified into several categories, according to expected cubic movement during a period (Petersen, 1999), and high-volume items are assigned to storage locations close to the I/O point in order to reduce pickers’ travel. Petersen (1999) and Petersen and Schmenner (1999) studied volume-based storage and showed significant savings over random storage. Edward (2001) indicates that the number of requests for an item during a period is the true measurement of popularity in a warehouse. Popularity can be translated into the number of times a picker must visit a storage location for a given item. Different from the volume-based criteria, this indicator counts how many times an item is requested by customers rather than the cubic volume or units of the item that are demanded during a period.

Petersen and Aase (2004) gave a deeper study on the interaction of routing methods and storage rules. However, largest gap routing was left out. Moreover, little research studies the combination of routing and class-based storage strategies (De Koster et al., 2007). Petersen (1999) studied the impact of routing and storage strategies on warehouse efficiency but without consideration of popularity-based storage. Petersen and Aase (2004) gave a deeper study on the interaction of routing methods and storage rules. They considered popularity-based storage rules but largest gap routing was left out. Petersen et al. (2005) also found that popularity-based storage performs better than volume-based storage, but this research again did not consider largest gap routing.

It is clear that largest gap routing as well as popularity-based storage should be studied more. In this research, an improved largest gap strategy (ILG) is developed. A simulation approach is applied to compare ILG routing and original largest gap routing, taking into consideration two different storages rules, i.e. random storage and popularity-based storage.

IMPROVED LARGEST GAP ROUTING

Under a largest gap routing strategy, the largest gap in a visited aisle is the distance between either any two adjacent picks, or between the first pick and the front cross-aisle, or between the last pick and the back cross-aisle. If the largest gap is the distance between two adjacent picks, the picker should perform a return as shown in Figure 1.

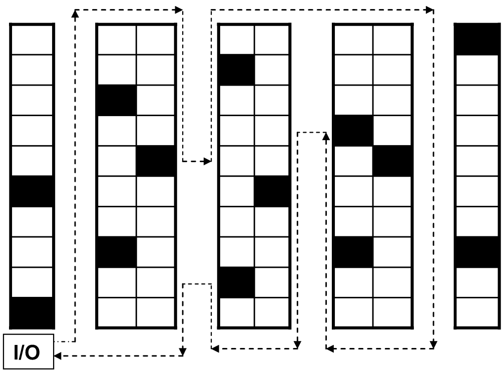


Figure 1: Largest Gap Routing

Under the principle of the largest gap strategy, when the quantity of visited aisles is one, a return is performed (return routing); when the quantity of visited aisles is two, a picker always traverses two aisles (also called "S" routing); when there are more than two aisles visited, the first and last aisle to be visited must be traversed as well, and the aisles in between will be entered as far as the largest gap within the aisle (largest gap rule).

This means that even when all the visited locations fall into the front half section of the warehouse, a picker should still traverse the first and last visited aisle. In this case, the walking distance could be reduced if the picker would perform return routes rather than traversing any aisles. Therefore, some modifications can be made to improve the performance of the largest gap strategy.

Figure 2 presents the logic of improved largest gap (ILG) routing. Each location is represented by a two-dimensional coordinate (x, y) and I/O is (0,0). The parameter max(y^i) is the vertical coordinate value of the furthest location visited from the front cross-aisle. The quantity of aisles visited during a pick tour is denoted by *n*. L^λ is the length of one picking aisle denoted by the number of stock locations, and $0.5L^\lambda$ is half of the length of the picking aisle. The function max(y^i) $\leq 0.5L^\lambda$ then means that all locations to be visited fall in the front half section of the picking area.

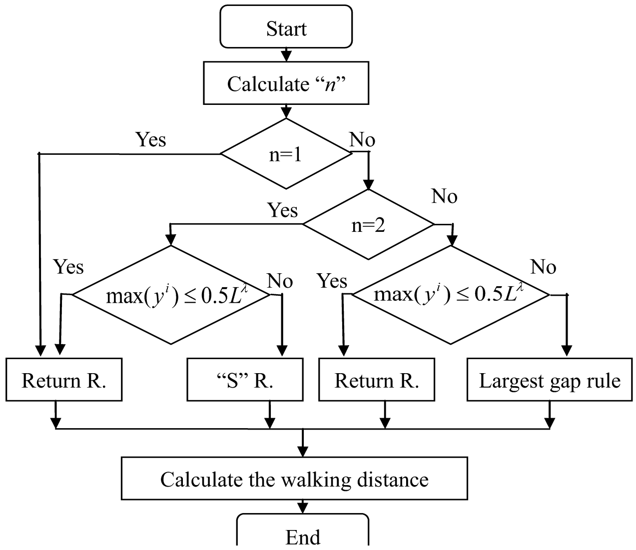
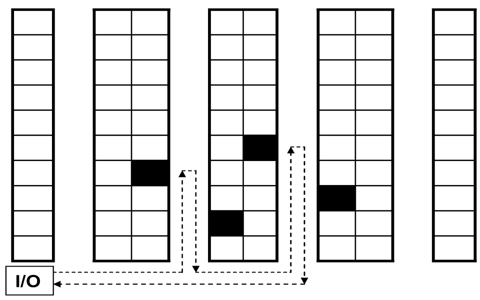
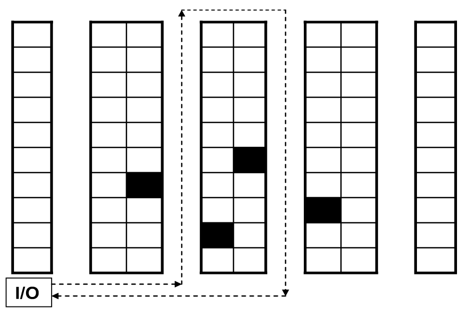


Figure 2: Decision Framework for ILG Routing

The improvements of ILG routing over largest gap routing stem from two aspects. The first aspect is illustrated in Figure 3, where there are two aisles visited during a pick tour. If the furthest location(s) visited from the front aisle fall in the front half section of the picking area, a return routing should be conducted under ILG routing as shown in Figure 3a; on the contrary, under traditional largest gap routing "S" routing would be performed as shown in Figure 3b.



a. ILG Routing



b. Largest Gap Routing

Figure 3: Difference of ILG and Largest Gap Routing

The second aspect of improvement is seen when there are more than two aisles visited during a picking tour. If all visited storage locations are distributed in the front half section, a return routing is applied again; on the contrary, under the traditional largest gap routing the first and last aisle would be traversed entirely.

CASE STUDY

This research is based on a broken-case cosmetics order picking system where single order picking, random storage and "S" routing are currently used. This scenario will be a base-case to discuss largest gap routing and ILG routing.

The warehouse layout has five picking aisles with front and back cross-aisles. The picking aisles are two-sided and wide enough for two-way travel. It is assumed that each item is assigned to one storage location and every location has the same size. An item is a single unique type of product; it is also called SKU (stock keeping unit) or product line.

Single order picking is applied. It is a manual picking environment where the picking cart is used to move the items picked back and forth to the I/O where each picking tour begins and ends. The I/O is located in the lower left corner of the warehouse. Bin-shelving is used and 800 items are stored. The height of the storage racks is four in terms of the number of stock locations. Each storage location is 1 meter by 0.3 meter. The width and length of the picking aisles are 2m and 6m respectively. The width of the cross-aisle is 3m.

EXPERIMENTAL DESIGN

This section presents the warehouse simulation model used for analysis and the experimental design. The purpose of this research is to identify the advantages of ILG routing in comparison to largest gap routing under several storages policies. The base-case is "S" routing and random storage. Two additional routing policies are examined: largest gap routing and ILG routing. The popularity-based storage policy is examined with four variations. This design of experiments results in 15 combinations.

Two proportions of popularity will be considered: on the one hand 20/80, which means that 20% of items contribute to 80% of all item requests, while the remaining 80% of items only provide 20% of all item requests; on the other hand 20/60 which indicates that 20% of the items provide 60% of all item requests and the remaining 80% of items provide 40% of all item requests. According to their popularity, all items will be classified into two groups: A and B.

The class-based storage rule also must consider the categorization of stock locations. Two principles are often used in practice to classify storage locations: within-aisle (W) and cross-aisle (C). Within-aisle storage means that the "A" items will be assigned to the aisle closest to the I/O point as shown in Figure 4a. According to a cross-aisle rule, the "A" items should be stored in the most accessible locations closest to the front cross-aisle, as shown in Figure 4b. In this study, storage assignment within each area is at random.

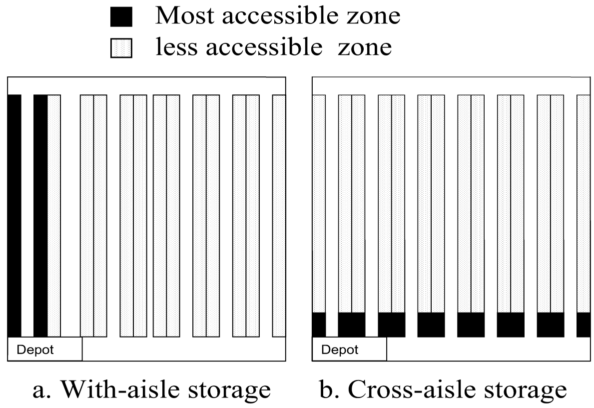


Figure 4: Location Categories

The 15 treatments from the 3×5 factorial design are evaluated using data sets generated by Monte Carlo simulation. A sensitivity analysis is conducted to evaluate different combinations of routing and storage rules. The analysis is carried out for different order sizes. More specifically, seven levels of order size are chosen: 3, 5, 10, 15, 20, 25 and 30 items. Among them, 3 and 5 items represent small orders; 10 and 15 items correspond to medium orders, and the rest are considered as large orders. Thus, for every given order size 15 combinations of routing and storages rules are tested. This results in 105 cells. Table 1 summarizes the design of experiment. Each cell is run for 1000 randomly generated orders and this results in a total of 105000 observations. The performance measurement for this experiment is the walking distance per picking tour.

Table 1: Experiment Factors

Factors	Explanation	Levels
Routing policies	"S" routing (S), largest gap routing (LG), improved largest gap routing (ILG)	3
Storage rules	Random(R), W(20/80), W(20/60), C(20/80), C(20/60)	5
Order size	3,5,10,15,20,25,30	7

RESULTS AND ANALYSIS

The analysis part has three sections. The first section explores the efficiency of ILG routing in comparison to LG routing. The second section examines the best performance of largest gap routings in combination with three types of storage policies. The final section investigates the effect of the popularity proportion on the performance of routing and storage policies.

Routing Performance

Figure 5 shows the savings percentage in average walking distance of a picking tour under random storage, for largest gap and ILG routings relative to "S" routing. The ILG routing is clearly a better heuristic than largest gap routing when the order size is smaller, for instance 3 items per order. Largest gap routing performs the same or sometimes even worse than "S" routing when the order size is three or less. This indicates that ILG routing can improve the performance of largest gap when order sizes are very small.

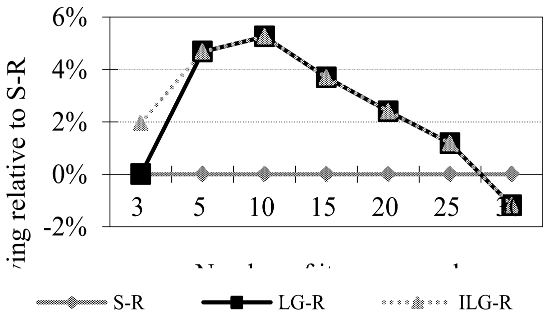


Figure 5: Percentage of Savings of ILG and Largest Gap Routing over Base-case under Random Storage

There is no significant difference between ILG and traditional largest gap routing when order sizes become larger than 3. Both largest gap routings achieve the biggest percentage of savings relative to "S" routing, 5.2%, when the order size is 10. As the order size increases, the advantages of largest gap routings are gradually lost. Both of the largest gap routings generate longer distance than "S" routing when the order size becomes 30. This indicates that largest gap routing is ineffective for large orders.

The Effect of the Storage Policy

Figure 6 compares the percentage of savings for different scenarios, i.e. different combinations of routing and storage policies, relatively to the base-case of "S" routing and random storage. Within-aisle and cross-aisle storage rules are applied under a 20/80 popularity distribution.

Firstly, we note that all three routing policies perform much better when they are operated with within-aisle storage, and the percentage of savings relative to the base case is between 13% and 35%. The ILG routing and traditional largest gap routing perform identically when order sizes ranges from 5 to 30 but when the order size is 3, ILG routing saves more than largest gap routing. Secondly, under the cross-aisle storage rule, ILG routing achieves savings relative to the base-case of nearly 5% more than largest gap routing, while both of them are becoming identical as order sizes increase to 15. Finally, in general, the scenario of ILG-W storage achieves the best performance for all order sizes in the experiment.

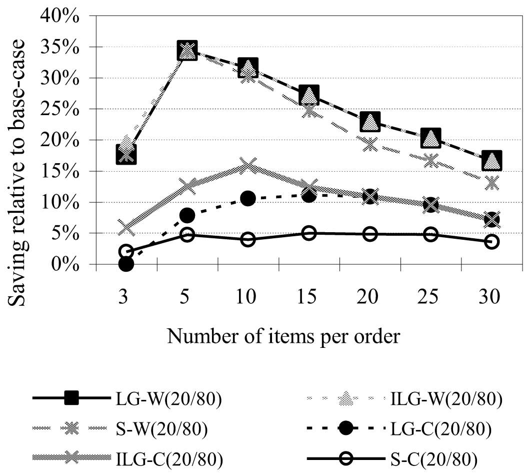


Figure 6: Effect of Storage Rules on Routings (20/80 Popularity Proportion)

The Effect of Popularity Proportion

The difference between the two types of popularity proportion can be found by comparing Figures 6 and 7. With an item popularity distribution of 20/80, the combinations of ILG-W and LG-W provide the largest savings (between 16% and 35% as order sizes vary) and S-W has an acceptable performance. Under a 20/60 proportion, the scenarios of ILG-W and LG-W still give the shortest distances. However, the savings percentage only ranges from 6% to 20%. Under both popularity proportions, ILG-W provides a little more savings than LG-W, when the order size is 3 items.

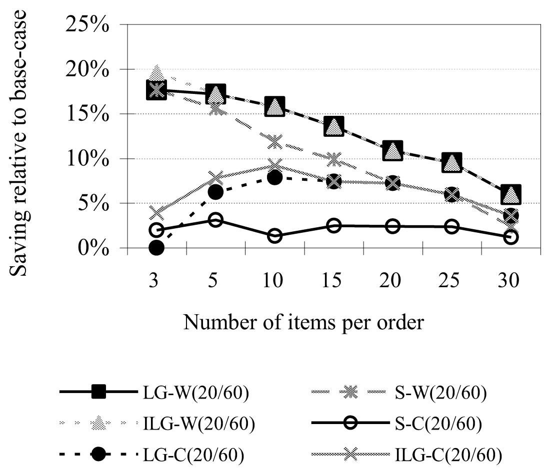


Figure 7: Effect of Storage Rules on Routings (20/60 Popularity Proportion)

In Figure 7, the relative savings achieved by S-W show a sharp decline as the order sizes increase. As a consequence, as long as the order sizes are smaller than 25 items, S-W is the third best policy, after ILG-W and LG-W; but when the order size is 30 items S-W is outperformed by LG-C. This is different to the scenario of 20/80 where S-W is always the third performer behind ILG-W and LG-W. ILG-C and LG-C acquire less than 10% savings for all order sizes.

In some circumstances, order sizes may vary. We now assume the situation where there is a balanced mix of orders, i.e. the proportion of small, medium and large orders are equal. The average percentage of savings for various order sizes is calculated for each combination of routing and storage rules. Figure 8 and 9 show the rankings of each combination for respectively the 20/80 and the 20/60 scenario.

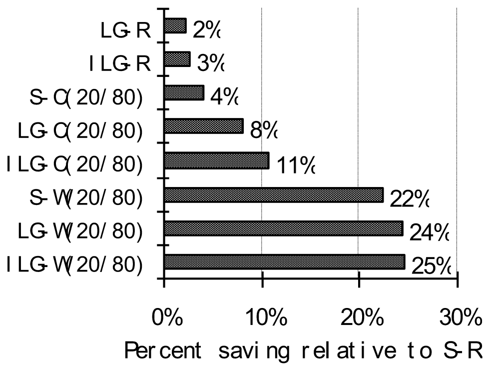


Figure 8: Ranking of Average Percentage of Savings under 20/80 Proportion

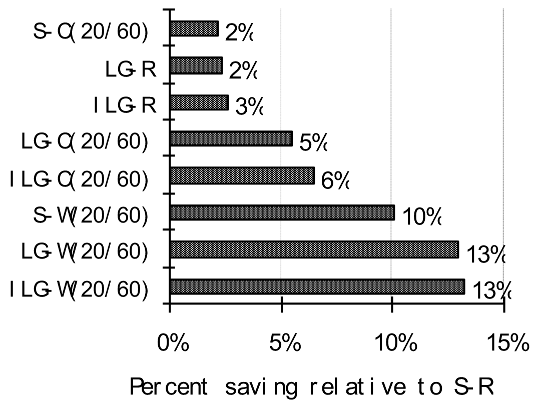


Figure 9: Ranking of Average Percentage of Savings under 20/60 Proportion

The change in popularity proportions has a negligible effect on the ranking of the heuristics combined with within-aisle storage. ILG-W, LG-W and S-W, are listed first, second and third respectively for both proportions. However, the change in popularity proportions has an effect on the ranking of S-C. The case of S-C (20/80) performs better than ILG-R and LG-R, while S-C is ranked behind ILG-R and LG-R with a 20/60 popularity.

CONCLUSION AND FURTHER RESEARCH

To achieve a more efficient order picking operation, it is necessary to reduce picking routes. Heuristics routings are commonly used due to their simplicity. A literature study has shown that little research focuses on largest gap routing and popularity-based storage. This paper evaluates an improved largest gap (ILG) routing policy in comparison with the original largest gap routing policy in a manual bin-shelving warehouse. It is concluded that the ILG routing policy can improve the performance of traditional largest gap, especially when the order size is smaller. Moreover, we investigated the effects of storage rules, order size and item popularity proportions on the performance of largest gap routings. When considering to switch from "S" routing to (improved) largest gap routing, an analysis by order size and by item demand popularity should be conducted. This is essential in order to gain insight in the potential returns of using largest gap routings. The results of this paper will provide decision-making support for order picking system designers and managers.

REFERENCES

Caron, F.; Marchet, G.; and Perego, A. 1998. "Routing Policies and COI-based Storage Policies in Picker-to-part Systems." *International Journal of Production Research* 36, No.3,713-732.

Coyle, J. J.; Bardi E.J.; and Langley C.J. 2002. *The Management of Business Logistics* (7th editon). South-Western College,Publishing, St. Paul.

de Koster, R.; Le-Duc, T.; and Roodbergen, K. J. 2007. "Design and Control of Warehouse Order Picking: A Literature Review." *European Journal of Operational Research* 182 ,No.2, 481-501.

Edward, H. F. 2001. *Supply Chain Strategy.* McGraw-Hill, 224-241.

Petersen, C. G. 1997. "An Evaluation of Order Picking Routing Policies." *International Journal of Operations & Production Management* 17, No.11,1098–1111.

Petersen, C. G. and Schmenner, R. W. 1999. "An Evaluation of Routing and Volume-based Storage Policies in An Picking Operation." *Decision Sciences* 30,No.2,481-501.

Petersen, C.G. 1999. "The Impact of Routing and Storage Policies on Warehouse Efficiency." *International Journal of Operations and Production Management* 19,No. 9-10, 1053-1064.

Petersen, C. G. and Aase, G. 2004. "A Comparison of Picking, Storage, and Routing Policies in A Manual Order Picking." *International Journal of Production Economics* 92, No.1, 11-19.

Petersen, C. G.; Aase, G. R. and Heiser, D. R. 2005. " Improving Order Picking Performance Utilizing Slotting and Golden Zone Storage. *International of Operatioins and Production Management* 25, No.9-10, 997-1012.

Ratliff, H.D. and Rosenthal, A. S. 1983. "Order-picking in A Rectangular Warehouse: A solvable Case of the Ttraveling Salesman Problem." *Operations Research* 31, No.3, 507-521.

Roodbergen, K. J., De Koster, R. 2001. "Routing Methods for Warehouses with Multiple Cross Aisles." *International Journal of Production Research* 39, No.9,1865–1883.

Tompkins, J. A.; White, J.A.; Bozer, Y.A.; and Tanchoco, J.M.A. 2003. Facilities Planning(3rd editon). Wiley John, New York , 447-457.

ACKNOWLEDGEMENT

This research is funded by the Fundamental Research Funds for the Chinese Central Universities and while Dr. Jinping Liu is a visiting researcher at the Department of Industrial Management, Gent University.

MANUFACTURING ROBOT CONTROL SIMULATION

OPTIMIZING BLENDED PTP MOVEMENTS FOR INDUSTRIAL ROBOTS BY MEANS OF METAHEURISTICS

Juergen Rossmann
Institute for Man-Machine Interaction
RWTH Aachen University
Ahornstr. 55
52074 Aachen, Germany
rossmann@mmi.rwth-aachen.de

Sven Severin, Oliver Stern
RIF e.V.
Department Robot Technology
Joseph-von-Fraunhofer-Str. 20
44227 Dortmund, Germany
{sven.severin|oliver.stern}@rt.rif-ev.de

KEYWORDS
Industrial robot, PTP movement, optimization, metaheuristic

ABSTRACT

Economy of time is fundamental to the efficiency of industrial production processes. Particularly in robot work cells, small time savings per cycle will accumulate and thus reduce production costs. In this context, the paper discusses the automatic optimization of blended PTP (point-to-point) movements that are commonly found in robot programs. It shows how the firefly algorithm can be adapted to the given problem and how it compares to other metaheuristics.

INTRODUCTION

A robot position can be described as a vector of all robot joints. A synchronous PTP movement drives the robot joints in a way that acceleration and deceleration times for all joints are equal. The slowest joint, which is determined by the distance to move and the motor specifications of the joints, determines the speed and acceleration of all the other joints. It is called the leading axis. PTP movement is by definition the fastest possible movement between two positions.

In cases where a PTP movement would lead to a collision between the robot and the environment, additional positions are necessary. These intermediate positions may or may not be reached, according to a given blend parameter for each intermediate position.

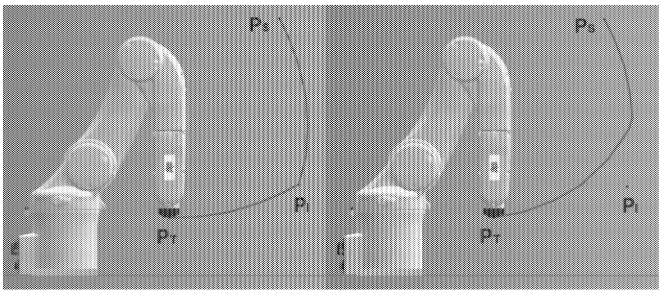


Figure 1: PTP Motion without and with Blending

Figure 1 illustrates a robot moving along a PTP trajectory. On the left side no blending occurs, on the right side the position P_I is blended. The blend parameter b describes the position on the trajectory from point P_S to the intermediate point P_I, where the trajectory is modified and blended into the movement to point P_T. Parameter b can be specified as the percentage of the distance the leading axis has to travel. A value of 0% means no blending, a value of 100% means that only half the distance to point P_I is travelled.

Since the robot does not stop at the intermediate point, the blended movement is faster. The challenge is to find the optimal location of the intermediate position, minimizing the duration of the movement while avoiding collisions with the environment.

STATE OF THE ART

Many offline programming systems for industrial robots include modules to plan and/or optimize robot trajectories for different applications like polishing, cutting, milling, or gluing. These modules need the knowledge of a process expert to specify the robot motion and are not fitted to optimize a general movement. The path planning system GLEAM (Blume and Jakob, 2009) uses genetic algorithms to produce paths for industrial robots. Although it is theoretically qualified to generate optimized PTP movements, the focus of GLEAM is on the planning of collision free paths, not on the optimization of single movements. In (Davidor, 1990), a method is shown that uses genetic algorithms to generate a trajectory for an industrial robot to follow a given path. Although it shows the superiority of genetic algorithms over hill-climbing and random search, the author demonstrates his method only on a kinematic device with three degrees of freedom. In (Tangpattanakul and Artrit, 2009) is demonstrated how to use the harmony search algorithm to optimize a given path according to the movement duration. This method does not consider a standard movement type like PTP and therefore needs a special control unit for the robot. So, currently, there exists no method to help robot programmers to optimize PTP movements for industrial robots in real world application scenarios.

The family of metaheuristic algorithms includes the genetic algorithms, evolutionary algorithms and swarm based algorithms (Weise 2009). Swarm based algorithms have in common that they evaluate a small set of solutions at a time and then build new and hopefully improved solutions. The method how these solutions are built is determined by the algorithm.

The particle swarm optimizer (Kennedy and Eberhart 1995) models the social behavior of collectives in swarms to find new solutions. Many improvements to this algorithm were suggested. Examples are an inertia weight factor (Eberhart and Shi 1998), or the guaranteed convergent particle swarm optimizer (van den Bergh and Engelbrecht 2002). Various algorithms for different problem classes resemble the behavior of specific animals, for example the ant algorithm (Dorigo et. al. 1996) or the honey bee algorithm (Nakrani and Tovey 2004). While these are specifically well fitted to optimize discrete problems, the firefly algorithm (Yang 2008) was designed to solve continuous optimization problems. Since its introduction in 2008 a number of enhancements were suggested, for example the lévy firefly algorithm (Yang 2010), the chaotic firefly algorithm (Coelho et. al. 2011) or the adaptive firefly algorithm (Farahani et. al. 2011).

PROBLEM ANALYSIS

To describe this optimization problem, a kinematic is needed which is capable of executing PTP motions, as well as a start and a target position for the movement, and a description of the surrounding geometry that prevents the kinematic from moving directly from start to end without any collisions.

A possible solution to the optimization problem is given by a sequence of intermediate positions and associated blend parameters. A solution may be written as a solution vector, where the length of the vector depends on the type of kinematic and the number of intermediate positions.

A solution vector for a kinematic with n degrees of freedom and p intermediate positions is given by:

$$sol = ((g_{1,1}, g_{2,1}, \dots, g_{n,1}, b_1), \dots, (g_{1,p}, g_{2,p}, \dots, g_{n,p}, b_p))$$

Where g are joint value vectors and b blend parameters. The performance of such a solution depends on three factors:

- The movement has to be free of collisions.
- The elapsed time should be minimal.
- There should be no unnecessary movements.

It is required to interpolate the blended PTP motion to quantify any values for these criteria. In cases where the actual interpolation algorithm for a robot is not known, it may be sufficient to approximate it by a general method described in (Severin 2011). It is assumed that any algorithm used for interpolation depends on an interpolation interval (IPO). Changes in speed may only occur between intervals, never within them. The IPO cycle for industrial robots ranges from approximately 1ms to 12ms.

It is sufficient to check whether a collision occurs or not. Time, duration, position, or involved bodies of a collision are not necessary to be known for performance evaluation.

The elapsed time can be taken as movement time of the leading axis.

The usefulness of a movement cannot be directly measured but compared between two solutions. So, if the durations of two collision free movements are equal, the one with more overall motion is considered to be less useful. To compare this, the combined motion of all axes can be measured as

$$w = \sum_N \sum_I |g_{n,i} - g_{n,i-1}|$$

Where N is the number of joints, I the number of interpolation steps, and $g_{n,i}$ the n-th element of the joint value vector for interpolation step i. There are several reasons for limiting the number of intermediate positions. First of all, every intermediate position adds to the movement time. In general, it is less time consuming for a PTP movement, to move faster along a longer trajectory then to brake for multiple near-stops on more intermediate positions. Secondly, the resulting program is probably maintained by a human programmer. The acceptance of computer optimized positions is much better if they are easy to comprehend. To solve the optimization problem different metaheuristics were reviewed. This paper will concentrate on the standard firefly algorithm as described in (Yang 2008).

FIREFLY ALGORITHM

A firefly has the ability to emit light to communicate with other fireflies. It is assumed that the luminosity and rhythm of this light impacts the mating behavior of these insects. The luminosity decreases with the distance between two fireflies. Therefore, a given individual cannot determine the brightest mate in its population but the brightest mate in its neighborhood. The firefly algorithm as described by (Yang 2010) was inspired by this behavior. Every firefly in a swarm represents a single solution to an optimization problem. The performance of every solution has to be evaluated by an appointed fitness function *F*. This performance rating corresponds to the luminosity and causes the fireflies to move towards better performing mates. When moving towards other fireflies, the solution vector will be changed accordingly. The following simplifications have been made:

- Every firefly may be attracted by any other firefly.
- Attractiveness is proportional to luminosity.
- Luminosity is derived from the solution vector that a single firefly represents.

The firefly algorithm can be written in pseudo code as follows:

```
initialize population of fireflies x_i (i=1,2,...,n)
compute luminosity I_i for position x_i by F(x_i)
define absorption coefficient γ
DO
    FOR i = 1 : n
        FOR j = 1 : i
            vary attractiveness by distance and γ
            IF( I_i< I_j)
                move firefly i towards firefly j
            ENDIF
        ENDFOR
    ENDFOR
    update all luminosities I_i
    replace 'global best position' if possible
UNTIL termination criterion
'global best position' contains optimization result
```

Each firefly occupies a position, which is represented by a solution vector *sol* for the problem under optimization. The movement of this position towards another firefly is done element by element as follows:

$$x_{i,n} = x_{i,n} + \beta_0{}^{-\gamma \cdot r^2} \cdot (x_{j,n} - x_{i,n}) + \alpha(\text{rand} - \frac{1}{2})$$

Where α is the leverage for the random factor, β_0 is the influence of attractiveness at minimum distance, γ the luminosity absorption coefficient, and r is the distance between the fireflies. *Rand* generates a random number in the interval [0, 1].

FITNESS FUNCTION

The fitness function is a mapping $F: \overrightarrow{sol} \rightarrow \mathbb{R}_+$. It evaluates the performance of a solution and creates a simple to

compare value for a solution vector. A higher performance is indicated by a lower value for F.
The following explanations will be illustrated by diagrams of the kinematic's configuration space. For demonstration purposes, the kinematic has only two planar rotational degrees of freedom. The values in the diagrams were obtained by simulating each corresponding solution in the simulation system CIROS (Roßmann et al. 2010).

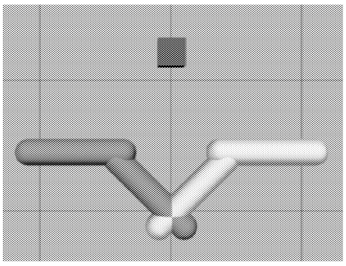


Figure 2: Demonstration Robot with Obstacle

Figure 2 shows the robot at its starting position (white) and at its target position (black). A box shaped obstacle is located above the kinematics origin.

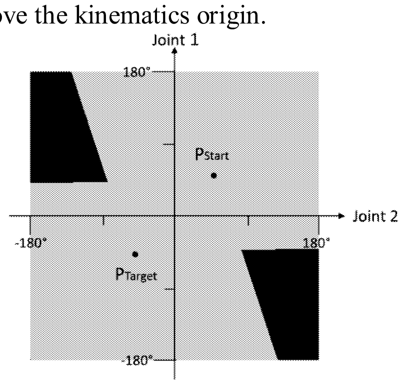


Figure 3: Configuration Space with Collision State

Figure 3 shows the collision state of the resulting path for every intermediate position described by a point in the configuration space. The blend parameter for this demonstration is 0%. Black areas represent collision free solutions. The diagram illustrates how much space may be denied by a single obstacle.

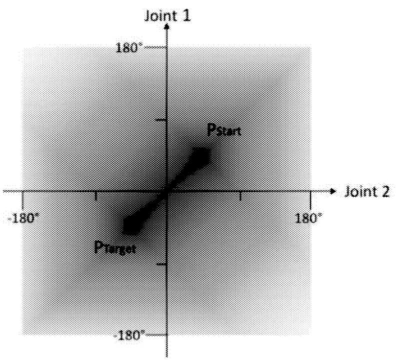


Figure 4: Configuration Space with Duration

Figure 4 shows the duration of the motion along the resulting path for every intermediate position described by a point in the configuration space. The values are normalized to the available levels of grey. Small durations are darker. The diagram shows how different intermediate positions affect the duration of the overall path. The cross-shaped pattern emerges from the composition of the PTP movement.

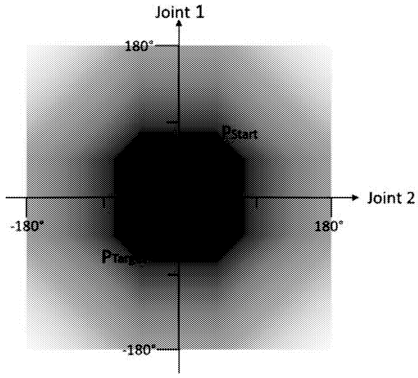


Figure 5: Configuration Space with combined Axis Motion

Figure 5 shows the cumulated movement length of all axes along the resulting path for every intermediate position described by a point in the configuration space. Intermediate points in a rectangular shape between P_{Start} and P_{Target} do not change the combined length (but maybe the duration). To combine the different criteria into one fitness function, the values have to be weighted according to their importance. This is, for PTP trajectories:

- Collision avoidance
- Minimal time
- No unnecessary motion

Additionally, it is desirable to create a smooth surface for the resulting fitness function, because the optimizing algorithm will work better on smooth surfaces. The measured time will be used as the base value. Collisions and unnecessary motion can be added as penalty times and extend the duration of the movement. Each trajectory containing a collision should be rated worse than any collision-free trajectory. If two solutions result in movements of equal duration, the one with less unnecessary motion should be rated better. To achieve this, we utilize that the difference in duration between two solutions is either 0 ms or at least one IPO cycle. The measured, accumulated way of all axes w will be normalized on an interval [0, IPO). To achieve this, an upper bound for w has to be found. Then, penalty for colliding trajectories is equal to the maximum duration of a movement for a given problem. The fitness function will then evaluate to lesser values for higher performance.
The maximum duration for a trajectory depends on the kinematic structure and the number of intermediate positions. With g_- as minimum joint value vector, g_+ as maximum joint value vector, and n_{IP} as number if intermediate positions, the maximum time t_{kin} and motion w_{kin} between g_- and g_+ can be derived from the interpolation between these positions. These have to be multiplied by the desired number of intermediate positions.

$$t_{max} = t_{kin} \cdot (n_{IP} + 1)$$
$$w_{max} = w_{kin} \cdot (n_{IP} + 1)$$

These values are only valid for an appointed optimization problem and the fitness values for different problems (different kinematics, start, or target positions) are not comparable. The duration t_{sol} and combined motion w_{sol} of an actual solution can only be obtained by interpolating the blended motion. They are incorporated into the fitness function as follows:

$$F = \begin{cases} t_{sol} + \frac{w_{sol}}{w_{max}} \cdot IPO & \text{, if not colliding} \\ 2t_{max} - (t_{sol} + \frac{w_{sol}}{w_{max}} \cdot IPO) & \text{, else} \end{cases}$$

The penalty is not a single t_{max} but, based on the idea that the path has to be longer to avoid collisions, a term that produces a downward slope headed towards solutions generating longer movements. However, it is warranted that every colliding solution is rated worse than any collision-free solution.

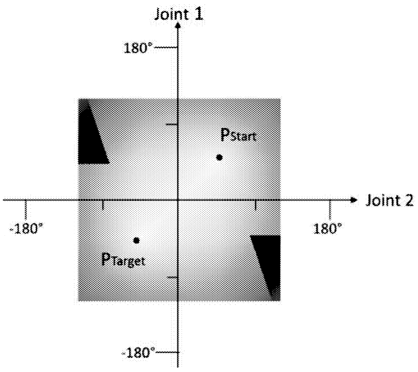


Figure 6: Configuration Space with Fitness Values

Figure 6 shows the resulting fitness values of the example movement along the resulting path for every intermediate position described by a point in configuration space. The function is fitted to avoid a particular class of collisions, namely the ones where the kinematic is forced to follow a longer trajectory. These obstacles are typically located between the start and the target position. Other obstacles restrict the movement by colliding with e.g. the shoulder of the robot, like a wall or the work cell enclosure. In many cases these obstacles force the robot to follow a shorter trajectory. The downward slope of the fitness function should point in the opposite direction.

MODIFIED FITNESS FUNCTION

Based on the analogy to the wildlife that is expressed in the firefly algorithm, each swarm element can be seen as an individual member of a swarm. As individuals, they can have their own idea on where to search for a mating partner or food, if nothing is in sight. Applied to the given problem, this means that the whole swarm may be divided into two groups of individuals, which, if they all represent solutions leading to collisions, follow different strategies to find better solutions. One group may try to extend the trajectory length, while the other tries to shorten it. The following formula shows a fitness function that implements the two strategies for different individuals.

$$F_i = \begin{cases} t_{sol} + \frac{w_{sol}}{w_{max}} \cdot IPO & \text{,if not colliding} \\ 2t_{max} - (t_{sol} + \frac{w_{sol}}{w_{max}} \cdot IPO) & \text{,coll. and i}\in 2Z \\ t_{max} + (t_{sol} + \frac{w_{sol}}{w_{max}} \cdot IPO) & \text{,coll. and i}\in 2Z+1 \end{cases}$$

This is equivalent to two different fitness functions which, in the collision free case, behave identically.

TEST SCENARIOS

To identify the correct parameter settings for the firefly algorithm and to test the optimization itself, we developed several scenarios. Each scenario consists of a robot, a start and a target position, and geometry that prevent the robot from moving directly between the two positions. The goal for the optimization algorithm is to find a fast trajectory from the start to the target position without colliding with the obstacles. The scenarios were chosen with the intention to resemble the most common use cases where PTP movements take place.

The scenarios include the predominant industrial robot types. These are the standard six-axis kinematic, a palletizer and a SCARA. They are assigned different roles: pick-and-place, handling, or machine to machine transfer. Several difficulties were added in the form of narrow operating space, big gripper geometries and configuration changes during the movement.

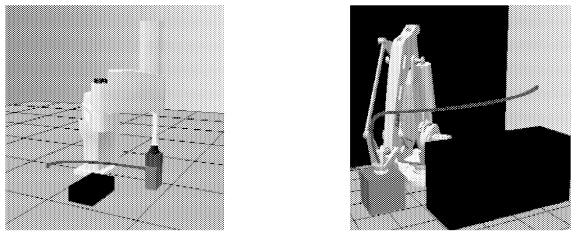


Figure 7: Scenarios A and B with special Robot Types

Figure 7 depicts scenario A containing a SCARA, optimized for pick-and-place tasks and scenario B containing a palletizing robot. In both cases, the task is to move the work piece across the obstacle to a target position, without colliding with it or the wall. These scenarios were used to ensure the applicability of the algorithm to different kinematic devices.

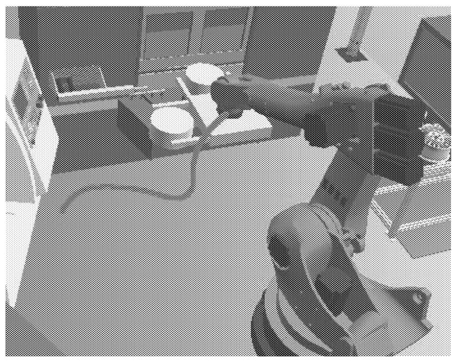


Figure 8: Machine to Machine Transfer in Scenario C

Scenario C that is shown in Figure 8 is an example of machine-to-machine transfer. The work piece is gripped by a double gripper and has to be moved to the machine on the left side.

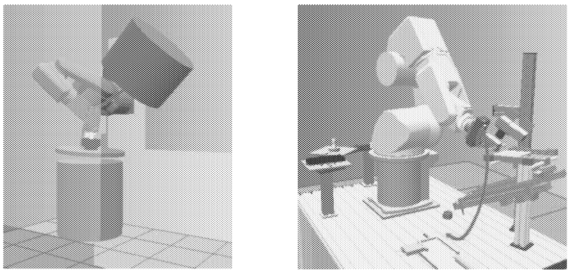


Figure 9: Handling Tasks featured in Scenario D and E

Scenario D features a robot within a very narrow operating space. The task is to transfer a work piece from the magazine and put it onto the lower cylinder. Scenario E shows a table mounted workstation. The task of moving the work piece from the magazine to the deposition requires a configuration change during movement. Both scenarios are shown in Figure 9. The scenarios C and D stem from real world industrial work cells, while scenario E originates from an educational training set.

COMPARISON CRITERIA

A criterion is necessary to measure the success of an appointed parameterization or optimization. The fitness value of the found solution is inadequate as quantification, because it depends on the kinematic structure and the scenario. We introduce a measurement called the *utilization of the optimization potential*. For this, two guesses are necessary. The minimum potential F_0 is a fitness value that gives a figure for what can be definitely achieved. The maximum value F_{100} shows what is possible to achieve. In the present case, the minimum values are manually optimized solutions for all the test cases. The maximum values were taken from the solutions of the 280 first parameterization tests.

Table 1: Optimization Potential

Scenario	F_0	F_{50}	F_{90}	F_{100}
A	0.936100	0.878447	0.832324	0.820793
B	1.631200	1.421198	1.253196	1.211195
C	1.730500	1.465456	1.253420	1.200411
D	1.161200	0.916086	0.719995	0.670972
E	0.916100	0.843617	0.785630	0.771133

Table 1 shows the manually obtained fitness value (F_0) and the maximum (F_{100}) fitness values. The potential utilization of any given fitness value for one of these scenarios can be calculated as:

$$P = 1 - \frac{(F - F_{100})}{(F_0 - F_{100})}$$

Where P is the utilized potential and F is the fitness value of the solution

ADAPTED FIREFLY ALGORITHM

A complete factorial experimental design as specified in (Kleppmann 2003) was constructed to configure the firefly algorithm. The required parameters are:

- α the factor for randomness
- β_0 the influence of luminosity at distance 0
- γ the light reduction factor
- n the population size
- i the number of iterations

Values for these parameters were suggested in (Yang 2008 and 2010) and further extended (Severin 2011). After evaluating the results of the experimental design, the following parameters have been determined:

- $\alpha = 1$
- $\beta_0 = 0.5$
- $\gamma = 10^{-5}$
- $n = 32$
- $i = 625$ (≈ 20.000 function evaluations)

Since the configuration space is limited, but solutions lying outside may be generated by the optimizer, these invalid solutions have to be repaired as described in (Paqet and Engelbrecht 2003). The element of the solution vector that is exceeding its upper limit will be changed to a value equal to the difference between its former value and the upper limit, added to the lower limit and vice versa. The advantage of this method is that swarm elements will less likely get caught in an 'edge' of the configuration space.

COMPARISON AND RESULTS

The obtained parameters produce the following results for all scenarios, averaged for 25 test runs per scenario.
Table 2 shows the utilization of the optimization potential by the firefly algorithm for each scenario.

Table 2: Utilized Potential for the Firefly Algorithm

Scenario	P
A	95.05%
B	99.14%
C	98.70%
D	92.26%
E	93.10%
Mean value	95.65%

In any case, the utilization easily reaches values above 90%. In order to find a fitting metaheuristic not only the firefly algorithm (FA) was reviewed but also the particle swarm optimization (PSO) as described in (Kennedy and Eberhart 1995) and the harmony search (HS) as described in (Geem et al. 2001). The exact way in which the algorithms were adapted to the problem under optimization can be found in (Severin 2010).

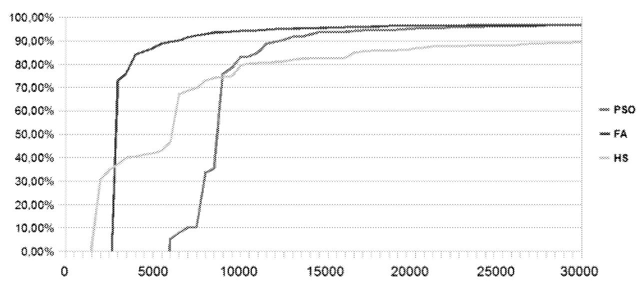


Figure 10: Comparison of Algorithms

Figure 10 shows a comparison of the three reviewed algorithms. The x axis shows the number of function evaluations that are necessary to get a result as shown on the y axis. The blue line shows the performance of the firefly algorithm, the red line shows the performance of the particle swarm algorithm, and the yellow line shows the performance of the harmony search. The firefly algorithm outperforms all other algorithms.

Table 3: Number of Function Evaluations necessary to reach Potential Utilization F_X

Algorithm	F_{30}	F_{50}	F_{75}	F_{90}	F_{95}
PSO	8500	9500	9500	13000	20500
FA	3500	3500	4000	7000	13500
HS	2500	7000	10500	-	-

Table 3 lists the approximate number of function evaluations needed to reach an appointed utilization barrier. Although the harmony search reached a 30% optimization after 2500 function evaluations, it is quickly outperformed by the particle swarm optimization algorithm and the firefly algorithm. Note that the firefly algorithm always needs fewer function evaluations than the particle swarm optimizer. The harmony search failed to exceed the 90% threshold even after 30000 function evaluations and reached approximately 89.5%.

Also, several modifications including the adaptive firefly algorithm (Farahani et al. 2011), the guaranteed convergent particle swarm optimizer (van den Bergh and Engelbrecht 2002), and the harmony memory tuning (Geem 2010) were studied. None of these modifications has led to a significantly higher performance, but most of them failed at least at one single scenario.

The impact of the automated optimization can be illustrated by an example. The robots movement in the machine-to-machine transfer scenario C took 1.76s in the manually optimized version. After optimizing the movement with the standard firefly algorithm it took 1.5s, which is ~15 % less.

CONCLUSION

The firefly algorithm is capable to optimize the examined problem. Especially with the chosen parameterization and fitness function, it needs less than 15000 function evaluations to utilize up to 95% of the existing potential.

All scenarios were optimized to at least 90%, which indicates a good general applicability.

The described algorithm is capable of finding and optimizing intermediate positions for PTP movements and may be used to support robot programmers at their work. The found solution is compatible with all industrial robot controllers that support PTP movement and needs no further hard- or software.

REFERENCES

van den Bergh, F. and Engelbrecht, A. P. 2002. “A New Locally Convergent Particle Swarm Optimiser“, In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, Vol. 3, 6 pp.

Blume, C. and Jakob, W. 2009. “GLEAM - General Learning Evolutionary Algorithm and Method : ein evolutionärer Algorithmus und seine Anwendungen”, KIT Scientific Publishing, Karlsruhe [in German]

Coelho, L., Bernert, D., Mariani, V. 2011. “A Chaotic Firefly Algorithm Applied To Reliability-Redundancy Optimization“, In *IEEE Congress on Evolutionary Computation*, 517-521

Davidor, Y. 1990. “Robot programming with a genetic algorithm” In *Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering*, 186-191

Dorigo M., Maniezzo, V., Colorni, A. 1996. “The Ant System: Optimization by a colony of cooperating agents“, In *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, Vol. 26, No. 1, 1-13

Farahani, S., Abshouri, A., Nasiri, B., Meybodi, M. 2011. “An Improved Firefly Algorithm with Directed Movement“, In *Proceedings of 4th IEEE International Conference on Computer Science and Information Technology*, 248-251, Chengdu, China

Eberhart, R. and Shi, Y. 1998. “Parameter Selection in Particle Swarm Optimisation“, In *Evolutionary Programming VII*, 591-600, Springer

Geem, Z. W., Kim, J. H., Loganathan, G. V. 2001. “A New Heuristic Optimization Algorithm: Harmony Search“, In *SIMULATION*, Vol. 76, 60-68

Geem, Z. W. 2010, “State-of-the-Art in the Structure of Harmony Search Algorithm“, In *Studies in Computational Intelligence*, Vol. 270, 1-10

Kennedy, J. and Eberhart, R. 1995. "Particle Swarm Optimization", In *Proceedings of the IEEE International Conference on Neural Networks IV*, 1942–1948

Kleppmann, W. 2003. “Taschenbuch Versuchsplanung“, 3. erweiterte Auflage, Carl Hanser Verlag, München [in German]

Nakrani, S. and Tovey, C. 2004. “On honey bees and dynamic server allocation in internet hosting centers“, In *Adaptive Behavior*, vol. 12, no. 3-4, 223-240

Paquet, U. and Engelbrecht, A. 2003. “A New Particle Swarm Optimiser for Linearly Constrained Optimisation“, In *The 2003 Congress on Evolutionary Computation*, Vol. 1, 227-233

Roßmann, J., Wischnewski, R., Stern, O. 2010. “A Comprehensive 3-D Simulation System for the Virtual Production“, In *Proceedings of the 8th International Industrial Simulation Conference (ISC)*, Budapest, 109-116

Severin S. 2011. “Entwicklung eines Verfahrens zur Optimierung von PTP-Bahnen für Industrieroboter mit Hilfe von Metaheuristiken“, Master thesis, FernUniversität Hagen [in German]

Tangpattanakul, P. and Artrit, P. 2009. “Minimum-time trajectory of robot manipulator using Harmony Search algorithm” In *6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 354-357

Weise, T. 2009. “Global Optimization Algorithms - Theory and Application“, eBook, www.it-weise.de

Yang, X. S. 2008. “Nature-Inspired Metaheuristic Algorithms“, Luniver Press, Frome

Yang, X. S. 2010. “Firefly Algorithm, Stochastic Test Functions and Design Optimisation“, In *International Journal on Bio-Inspired Computation*, Vol. 2, No. 2, 78-84

AN ADAPTIVE HARMONY SEARCH ALGORITHM TO SOLVE A ROBOTIC ASSEMBLY LINE BALANCING PROBLEM

Slim Daoud
Université de Technologie de Troyes,
Institut Charles Delaunay (STMR UMR CNRS 6279), LOSI,
Aries packaging, Rosière-près-Troyes, 10430,
France

Farouk Yalaoui
Lionel Amodeo
Hicham Chehade
Université de Technologie de Troyes,
Institut Charles Delaunay (STMR UMR CNRS 6279), LOSI,
France

Philippe Duperray
Aries packaging, Rosière-près-Troyes, 10430,
France

Email : (slim.daoud, farouk.yalaoui, lionel.amodeo, hicham.chehade) @utt.fr
Email: philipe.duperray@mwv.com

KEYWORDS
Robotic assembly line balancing, mixed integer linear program, optimization, adaptive harmony search.

ABSTRACT

This paper presents a method based on an adaptive harmony search algorithm to solve a robotic assembly line balancing problem. This robotic assembly system, which covers an industrial line, consists of seizing products on a moving conveyor and placing them on different location points. The goal is to optimize the efficiency of the line by assigning the suitable tasks and components to each robot. As in our industrial application we are bounded by the execution time, we suggest an adaptive harmony search (AHS) to solve the above problem. A mixed integer linear program was developed in order to assess the quality of our method (AHS). As the exact method deduced by Cplex requires a large execution time, we are unable to apply it for the industrial application. This is the main advantage of our method compared to the exact method. Numerical results show that our method performs efficiently for the tested instances in a reasonable computational time.

INTRODUCTION

Using robots, the flexibility and the automation of assembly lines can be enhanced. In this work, we are interested in a robotic system which realizes pick and place operations from a product moving conveyor to different location points. The goal is to have a real time management of the robots in order to improve the efficiency and the performances of the line. For that purpose, our goal is to define the suitable pick and place operation (task) and components to assign to each robot of the line. Therefore, the optimization of the gripping performances by maximizing the number of seized products and the lines balancing are considered in this paper. Since our output results must be produced within a bounded time, we propose an adaptive harmony search. In literature, different works have been interested in assembly line balancing problems (ALBP). Several versions of the assembly lines balancing problems (ALBP) deal with the modification of the objective function as described by (Scholl and Becker, 2006) and (Chehade *et al.*, 2010). (Kilincci, 2011) have developed a new heuristic algorithm based on a Petri net approach for SALBP 1. This algorithm provides an order of firing sequence of transitions from Petri net model of precedence diagram. (Blum and Miralles, 2011) have proposed an algorithm based on beam search for solving the assembly line worker assignment and balancing problem which is considered as a generalization of SALB-2. For type–E which is similar to our problem, (Nai-chieh and Chao, 2011) have proposed a type E simple assembly line balancing problem (SALBP-E) that combines models SALBP-1 and SALBP-2. Another type of ALBP was described by (Delorme *et al.*, 2012) who have dealt with a new line balancing problem called transfer line balancing problem (TLBP). They have proposed a combinatorial design to minimize the cost transfer line. In a later work, (Battaïa and Dolgui, 2012) have proposed pre-processing methods which can reduce the size of the initial problem in order to shorten the solution time required.

In our work, we are interested in an automated assembly line which uses robots to perform all tasks. This problem in literature is called robotic assembling line balancing problem. (Rubinovitz and Bukchin, 1991) were the first to introduce the robotic assembly line balancing (RALB). (Kouja *et al.*, 2000) have developed two-stage methodology to perform robotic assembly cells. (Nicosia *et al.*, 2002) have proposed a dynamic programming algorithm in order to minimize the cost of the workstation. Their problem aims to assign all tasks to non-identical workstations subject to precedence constraints. (Yoosefelahi *et al.*, 2011) have proposed a new mixed-integer linear program which aims to minimize the cycle time, robot setup costs and robot costs. They have developed three versions of multi-objective evolution strategies.

Since in our industrial application, we are bounded by the execution time, we try to apply the adaptive harmony search (AHS). This method is able to give outputs results within a small bounded time. (Afkousi-Paqaleh *et al.*, 2010) have presented a harmony search algorithm to find the optimal scheduling while the total operating costs can be minimized in unit commitment problem. (Zou *et al.*, 2010) have proposed a novel global harmony search algorithm which is inspired by the particle swarm to solve a task assignment problem. (Degertekin, 2012) have developed two improved harmony search algorithms called efficient harmony search (EHS) and self adaptive harmony search algorithm (SAHS) for sizing optimization of truss structures.

We note the gap analysis regarding the optimization of robotic assembly line balancing efficiency in the literature. The contribution of this study is to develop an adaptive harmony search since we are bounded by the industrial execution time. At our knowledge, this method has never been adapted in this field. For our study, the efficiency is defined as the number of seized components of products by each robot.

The remainder of this paper is organized as follows. The second section presents the problem description which is the type E of robotic assembly line balancing problems and the problem formulation. The resolution methods are presented in the third section. Computational experiments and numerical results are presented in the fourth section before ending the paper by a conclusion and perspectives for future works.

PROBLEM DESCRIPTION AND FORMULATION

Problem description

We are interested in developing a method which satisfies the industrial constraints. This method is based on a real time system which is able to manage the performances of robotic lines.

The main objective of this work is to balance all tasks for each robot and to maximize the line efficiency *E* by maximizing the gripping products. This robotic assembly line is illustrated in figure 1 where robots seize the components (on the left side of the figure) from the moving conveyor belt and place them in specific locations (right side of the figure) in order to get the final assembly product. To produce this latter, a certain number of assembled tasks are needed. These operations are the robot motion and represent the moving of *K* components from *J* to the location points. The final assembled product is composed of *n* layers and for each layer we have K_α location points. The order in which the tasks have to be performed is ensured by the precedence constraints which are also represented as a cyclic precedence graph as illustrated in figure 2. In this latter, all the tasks are presented like nodes. Our robotic assembly line consists of *I* serial robots. As mentioned before, we are interested to maximize the line efficiency E (productive part of the line capacity) which is in our case the maximization of the seized components by means of robots.

The following assumptions are mentioned by (Levitin *et al.*, 2006) and (Gao *et al.*, 2009) and we have added some extra ones:

(1) This precedence relationships is due to technological assembly constraints

(2) The duration of assembling task is deterministic and cannot be subdivided

(3) The duration of an assembling task depends on the assigned robot and the assigned located points.

(4) The assignment of an assembling task for each robot is limited by the throughput rate which manages the robot activity to perform the assembling task.

(5) A single robot is assigned to each station

(6) One type of product is considered in this problem.

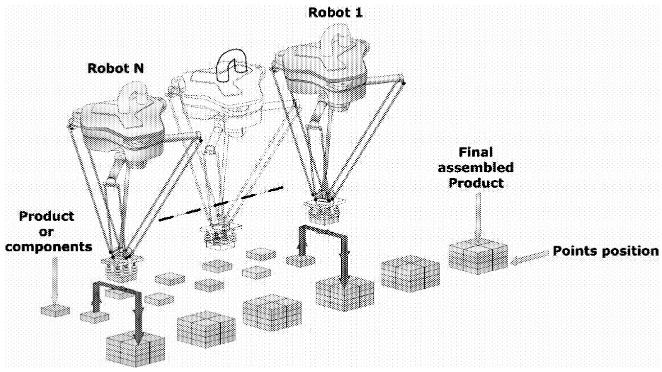


Figure 1: Robotic assembly systems description

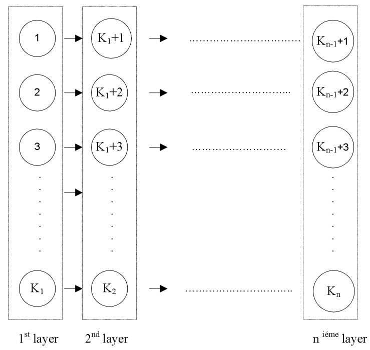


Figure 2: Precedence constraints of assembly tasks to assign on a robotic assembly line

Mathematical formulation

This section presents the mathematical formulation as described by (Daoud *et al.*, 2012) . For that, we first start by presenting the index and the parameters of the model.

Index:

i: index of robots, $i = 1, 2,. . ., I$
j: index of components $j = 1,2,..J$
k: index of location points, $k=1,2,\ldots,K$
k_α: index of location points in the α^{th} layer of components
m: index of final assembly products, $m=1,2,\ldots,M$

Notations:

C_{max}: the required time to assemble the J components
C_{moy}: the average workload of different robots
δ: The proportion between the average workloads of robots
n: the total number of layers of components
t_{ijkm}: the processing time of robot i to seize component j and placed it in the location points k of the final product m
K_α: the total number of components in the α^{th} layer
I: the total number of robots
J: the total number of components
K: the total number of location points
M: the total number of final assembled products

Decision variable:

$$x_{ijkm} = \begin{cases} 1 & \text{if robot } i \text{ seizes component } j \text{ and places it in location point } k \text{ of the final assembly product } m \\ 0 & \text{otherwise} \end{cases}$$

Our model is strictly linear. Thus, the CPLEX solver can be used here to solve the problem. The RALB-E problem is then formulated as described by (Daoud *et al.*, 2012):

$$Maximize\ (E) = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{m=1}^{M} x_{ijkm} . \quad (1)$$

The objective function (1) consists of maximizing the line efficiency E (productive part of the line capacity) which is the number of seizing components by the robots.

Subject to

$$\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ijkm} \le 1, \forall k = 1,..., K, \forall m = 1,..., M . \quad (2)$$

Constraint (2) shows that each component has to be placed in one location points.

$$\sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{m=1}^{M} x_{ijkm} . \le 1, \forall j = 1,..., J \quad (3)$$

Constraint (3) means that one robot seizes one component and placed it on one location point.

$$\sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{m=1}^{M} t_{ijkm} x_{ijkm} \le C_{\max} . \forall i = 1,..., I \quad (4)$$

Constraint (4) represents the required time to assemble the J components for each robot to process the assigned tasks to complete the assembly of all the final assembly products. For that, the assigned tasks for each robot must be less than the maximum elementary throughput rate.

$$C_{moy} = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{m=1}^{M} t_{ijkm} x_{ijkm}}{I} \quad (5)$$

Constraint (5) shows the average of workload of different robots.

$$\sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{m=1}^{M} x_{ijkm} t_{ijkm} - \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{m=1}^{M} x_{sjkm} t_{sjkm} \le \delta\ C_{moy} \quad (6)$$
$$\forall i = 1,..., I, \forall s = 1,..., I, i \ne s .$$

$$\sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{m=1}^{M} x_{ijkm} t_{ijkm} - \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{m=1}^{M} x_{sjkm} t_{sjkm} \ge -\delta\ C_{moy} \quad (7)$$
$$\forall i = 1,..., I, \forall s = 1,..., I, i \ne s .$$

Constraint (6) and (7) ensure that the total task processing time assigned to each robot does not exceed the proportion of the average of the workload of different robots. This constraint ensures the balancing between all robots.

$$\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ijk_1m} - \sum_{i=1}^{I} \sum_{j=1}^{J} x_{ijk_2m} \ge 0, \forall m = 1,..., M, , \forall k_1 = 1,..., K_1, \quad (8)$$
$$\forall k_2 = K_1 + 1,..., K_2 .$$

Constraint (8) imposes the precedence conditions among the assembly tasks and components in the first and the second layer in the same final assembled product. For that, a robot cannot perform a layer of components if the previous layer is not full.

$$\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ijk_\alpha m} - \sum_{i=1}^{I} \sum_{j=1}^{J} x_{ijk_{(\alpha+1)}m} \ge 0, \forall m = 1,...,M, \forall \alpha = 2,...,n-1, \quad (9)$$
$$\forall k_\alpha = K_{(\alpha-1)} + 1,..., K_\alpha, \forall k_{(\alpha+1)} = K_\alpha + 1,..., K_{(\alpha+1)} .$$

Constraint (9) generalizes the precedence constraints in the same final assembled product if the number of layers in the box exceeds two layers.

$$\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ijk_2m} - \sum_{i=1}^{I} \sum_{j=1}^{J} x_{ijk_1(m+1)} \ge 0, \forall m = 1,..., M, \forall k_1 = 1,..., K_1, \quad (10)$$
$$\forall k_2 = K_1 + 1,..., K_2 .$$

Constraint (10) imposes the precedence conditions among two successive final products. As a result, the last layer of the final assembled product m must be processed before the first layer of final assembly product $m+1$.

$$\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ijk_{(\alpha+1)}m} - \sum_{i=1}^{I} \sum_{j=1}^{J} x_{ijk_\alpha(m+1)} \ge 0, \forall m = 1,...,M, \forall \alpha = 2,...,n-1, \quad (11)$$
$$\forall k_\alpha = K_{\alpha-1} + 1,..., K_\alpha, \forall k_{(\alpha+1)} = K_\alpha + 1,..., K_{\alpha+1} .$$

Constraint (11) generalizes the precedence constraint for two successive final products if the number of layers is greater than 2.

$$x_{ijkm} = 0\ \ or\ \ 1, \forall i = 1,..., I, \forall j = j,..., J, \forall k = 1,..., K, \forall m = 1,..., M \quad (12)$$

Constraint (12) means that the decision variables have two values 1 (if task i for the component k is assigned to robot l and placed it in final assembly product m) or 0 (otherwise).

RESOLUTION METHOD

This section contains the adaptive harmony search developed to solve the robotic assembly line balancing. The idea is to test the ability of this method to manage the functioning of pick and place robotic assembly line which requests a short execution time. For this reason, we are unable to apply an exact method that requires a large execution time.

The adaptive harmony search

Harmony search (HS) algorithm has been developed by (Lee and Geen, 2005) on the basis of natural musical performances of searching for a perfect state of harmony as described by (Degertekin, 2012).

Solution encoding:

In order to apply the algorithm to our problem, we have adopted the encoding presented below for the robotic assembly line with *KM* location points, *J* components and *I* robots. We represent each harmony with a matrix of location points, robot number and component number. Table 1 shows an illustrative example of the r^{th} harmony H^r, which describes the pick and place operation (task) assigned to each robot. This pick and place operation is defined by the component number and the location points in the final assembled product. So each harmony is the whole of different triplet (location points, robot number and component number).

For example, $H_r^1 = \{1, 1..I, 1..J\}$ means that the first location points in the assembled product is attributed to robot number between 1 and *I* and to component between 1 and *J*.

Table 1: An example for the r^{th} harmony

The r^{th} harmony H^r	Location points	1	2	…	*KM*
	Robots number	1..I	1..I	…	1..I
	components	1..J	1..J	…	1..J

The procedure of the adaptive harmony search:

The flowchart of the proposed algorithm is shown in figure 3 and explained in the following steps as described by (Afkousi-Paqaleh *et al.*, 2010) and (Dgetekin, 2012) :

Step 1: Initializing the adaptive harmony search parameters

These parameters are: the number of design vectors in harmony memory (*HMS*), harmony memory consideration rate (*HMCR*), pitch adjusting rate (*PAR*) and the stopping criterion.

Step 2: Initializing harmony memory

The size of the harmony memory (HM) matrix is HMS. In this step, the harmony memory matrix is filled with many random vectors (harmony) as shown in figure 4. In this latter, (H^1, H^2,…, H^{HMS-1}, H^{HMS}) and ($E(H^1)$, $E(H^2)$,…$E(H^{HMS-1})$, $E(H^{HMS})$) are the harmony vectors and the corresponding objective function values respectively.

Step 3: Improvising a new harmony

Generating a new harmony is called improvisation. The new harmony $H^{new} = (H^{new}_1, H^{new}_2,…, H^{new}_{KM})$ is generating based on harmony memory consideration and pitch adjustment as described on the equation (13) and (14). Therefore, taking into consideration that each value of the new harmony is deduced from the historical values of the harmony matrix with HMCR probability $H^{new}_t \in \{ H^1_t, H^2_t,…, H^{HMS-1}_t, H^{HMS}_t\}$ $\forall t \in \{1,2,...,KM\}$ or different value of harmony memory . While, the (1-HMCR) is the probability of choosing randomly a value from the all possible solution H_S.

$$\begin{cases} H^{new}_r \in \{H^1_r, H^2_r,..,H^{HMS-1}_r, H^{HMS}_r\} \text{ with probability HMCR} \\ H^{new}_r \in H_S \text{ with probability (1-HMCR)} \end{cases} \quad (13)$$

The pitch adjusting process, for the new harmony, is performed only after a value is chosen from the harmony memory. In this step, we perform the neighborhood search with probability PAR (pitch adjustment rate) eq (14) in order to enhance the solution.

$$\text{Pitch adjusting decision for } H^{new}_r \longleftarrow \begin{cases} \text{Yes with probability PAR} \\ \text{No with probability (1-PAR)} \end{cases} \quad (14)$$

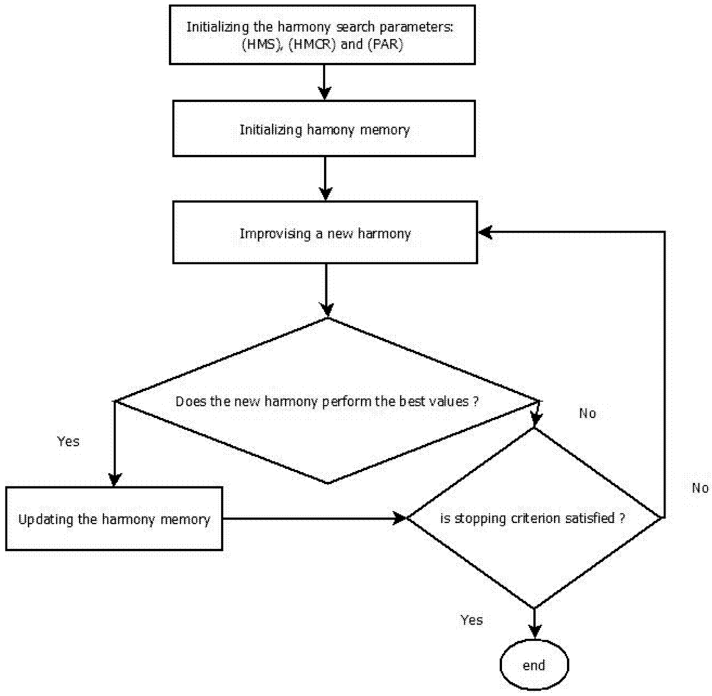


Figure 3: Flowchart of the immune algorithm, (Afkousi-Paqaleh *et al.*, 2010)

$$HM = \begin{bmatrix} H^1_1 & \cdots & H^1_{KM-1} & H^1_{KM} \\ H^2_1 & \cdots & H^2_{KM-1} & H^2_{KM} \\ & \vdots & & \\ H^{HMS}_1 & \cdots & H^{HMS}_{KM-1} & H^{HMS}_{KM} \end{bmatrix} \begin{matrix} : E(H^1) \\ : E(H^2) \\ \\ : E(H^{HMS}) \end{matrix}$$

Figure 4: Harmony memory matrix

We may notice, that the probability to choose a neighborhood value is PARxHMCR.

Step 4: Updating the harmony memory

We update the harmony memory, if the new harmony is better than the worst.

Step 5: termination criterion

The stopping criterion is the number of generations N_g.

The adaptive harmony search algorithm:

The adaptive harmony search algorithm is represented below

Structure of AHS

Initialize the adaptive harmony search HMS, HMCR and PAR
Generate randomly the initial harmony memory HM.
Check the precedence constraint of assembled task[k]
If assembled task[k] = Robot i then assembled task [k+1] = {Robot i or Robot j} ∇ *k*$\in$ *pred(h) and* ∇ *(i,j)* $\in$ *{1,…,I}*
For** each iteration t **do
Evaluate the fitness of each individual
Is stopping criteria satisfied (t=Ng?)
If** t=Ng then **stop
Else
For** each k* $\in$ *[1..K]* ***do

Improvising a new harmony
***If** rand() ≤ HMCR then* $H^{new}_i \in \{ H^1_i, H^2_i, \ldots, H^{HMS-1}_i, H^{HMS}_i \}$
Else
$H^{new}_i \in H_S$ %(all solutions)
***If** rand() ≤ PAR then* Pitch adjusting decision is YES
Do neighborhood search $H^{new}_r = H^{new}_{r+q}$ where m is the neighboring index q ∈ *{…-2, -1, 1, 2, …}*
Else
Pitch adjusting decision is NO
Insert the new harmony in next harmony memory
Sorting the next population such as selected the have which have the best fitness
Update the population
End for
End for
End If
End If
End If

In this paper, we use a variable PAR as described by (Degertekin., 2012). The PAR changes dynamically with the generation number in order to manage the intensification and diversification of the algorithm. In fact, the search space has to be better explorerd for this reason we use PAR deduced by the equation (15). After that, it should be limited to the search space within a neighborhood of the optimal solution. So, we use the equation 16.

$$PAR_1(t) = PAR_{\min} + \frac{(PAR_{\max} - PAR_{\min})}{T} \times t \qquad (15)$$

$$PAR_2(t) = PAR_{\max} - \frac{(PAR_{\max} - PAR_{\min})}{T} \times t \qquad (16)$$

Where, PAR: pitch adjusting rate for each generation, PAR_{min}: minimum pitch adjusting rate, $PAR_{\max}$: maximum pitch adjusting rate t : generation number and T: the total generation number.
In order to show the advantage of dynamic PAR, we compare it to fixed PAR which is used in traditional harmony search.

COMPUTATIONAL EXPERIMENTS

We realize computational experiments by studying robotic assembly systems structures which are different by the number *i* of robots, *k* the number of location points and *n* the number of layers of the final assembly product.. For each structure, we have tested four instances by changing at each time the number of location points and the number of layers. The precedence diagram of our problem which presents the precedence relations among the final assembled product is shown in figure 2. The first structure (S1) represents a system with 2 robots, 4 or 8 location points and 2 or 4 layers whereas (S2), (S3) and (S4) refer respectively to systems with 4, 6 and 8 robots with always the same number of location points and layers. The processing times of the assembly task are generated according to a uniform distribution and dependent on the layer. In our study, the processing time for the first layer is between [1, 1.2] because the robot takes more time to deposit the components in the location points of the assembled product and [0.8, 1] for the rest. The required time to assemble *J* components C_{max} depends on the structure, for (S1) and (S2) C_{max} = 33.6 (s) and for (S3) and (S4) C_{max} = 16 (s). The number of final assembled products to simulate the performance of the line is fixed at 80. This number is taken from an industrial case at the starting phase of a robotic line of dairy food. The industrial application of our method requires an execution time of less than **600 ms**.

Table 2 presents the parameters of the pick and place robots for all the tested structures.

Table 2: Robot parameters

Robot	Average cycle time	0,8 mm/s
	Coverage area length	1050 mm
	Distance between two robots	200 mm

Based on work of (Degertekin, 2012), the AHS parameters values are as follows: the harmony memory (*HMS=20*), the harmony memory consideration rate (*HMCR=0.9*), the minimum pitch adjusting rate (PAR_{min}= *0.2*), the maximum pitch adjusting rate (PAR_{max}= *0.8*) and the stopping criterion N_g= 100.
Table 3-4 show the comparison results and the execution times for the exact method and our method with the different tested instances. Table 3 presents the numerical results and the execution time with a fixed pitch adjusting rate (PAR= 0.35, 0,65, 1) as described by (Afkousi-Paqaleh *et al.*, 2010). Table 4 shows the results and the execution time with dynamic PAR (eq.16).
The AHS with fixed pitch adjusting rate (PAR) has succeeded for the first 11 instances to reach the optimal solutions that have been obtained with the exact method. The optimal solutions are showed in bold. For the AHS with dynamic PAR, it has succeded to obtain 13 instances. For the rest of the instances of a larger size, the difference is very small. As result the worst gap (3.86%) is obtained by AHS with fixed PAR. It corresponds to structure (S4) with 8 robot, 8 location points and 4 layers.
As to the computational results, the average of the execution times obtained with Cplex is 11.24(s) against 0,3(s) for AHS with PAR fixe and 0.34(s) for AHS with dynamic PAR. The difference of both algorithms is due to dynamic PAR procedure. The execution time for each method which respects the industrial constraint is highlighted in table 3-4. We may notice that for the execution time of Cplex are not acceptable for the industrial case since are greater than 600 ms. This fact represents a great interest because our method is able to manage in real time the functioning of an industrial robotic system.

Table 3: Numerical results about the number of seized products and the execution times by a fixed pitch adjusting rate (PAR)

	Robots	Location points	Number of layers	Number of products (E)		Execution time	
				Cplex	AHS	Cplex	AHS
(S1)	2	4	2	**66**	**66**	1.83	**0.121**
			4	**65**	**65**	1.21	
		8	2	**66**	**66**	2.13	
			4	**65**	**65**	4.12	
(S2)	4	4	2	**80**	**80**	5.98	**0.236**
			4	**80**	**80**	6.76	
		8	2	**80**	**80**	3.54	
			4	**80**	**80**	6.81	
(S3)	6	4	2	**80**	**80**	12.45	**0.386**
			4	**79**	**79**	10.86	
		8	2	**80**	**80**	11.78	
			4	**80**	79	13.7	
(S4)	8	4	2	**80**	78	24.82	**0.471**
			4	**78**	77	21.28	
		8	2	**79**	77	25.91	
			4	**78**	75	26.78	
Mean						11.24	**0.30**

Table 4: Numerical results about the number of seized products and the execution times by a dynamic pitch adjusting rate (PAR)

	Robots	Location points	Number of layers	Number of products (E)		Execution time	
				Cplex	AHS	Cplex	AHS
(S1)	2	4	2	**66**	**66**	1.83	**0.131**
			4	**65**	**65**	1.21	
		8	2	**66**	**66**	2.13	
			4	**65**	**65**	4.12	
(S2)	4	4	2	**80**	**80**	5.98	**0. 243**
			4	**80**	**80**	6.76	
		8	2	**80**	**80**	3.54	
			4	**80**	**80**	6.81	
(S3)	6	4	2	**80**	**80**	12.45	**0.398**
			4	**79**	**79**	10.86	
		8	2	**80**	**80**	11.78	
			4	**80**	**80**	13.7	
(S4)	8	4	2	**80**	**80**	24.82	**0.493**
			4	**78**	77	21.28	
		8	2	**79**	78	25.91	
			4	**78**	76	26.78	
Mean						11.24	**0.34**

CONCLUSION

This paper has introduced a method based on an adaptive harmony search algorithm for a robotic assembly line balancing problem. The objective of our study is to prove that our method is able to manage the functioning of an industrial application in order to maximize the efficiency. The numerical results show the performance of AHS with dynamic PAR in order to improve the convergence behavior. The perspectives of our wor are first to apply the fuzzy logic controller in order to tune the pitch adjusting rate. The application of the local search may also be interested such to improve the selection of optimal solutions.

REFERENCES

Afkousi-Paqaleh. M, Rashidinejad. M, Pourakbari-Kasmaei. M. 2010, “An implementation of harmony search algorithm to unit commitment problem”, Electrical Engineering, (92), 215-225.

Battaïa. O, Dolgui. A. 2012, “ Reduction approaches for a generalized line balancing problem”, Computers & Operations Research, (39), 2337-2345.

Bluma. C, Miralles. C. 2011, “On solving the assembly line worker assignment and balancing problem via beam search”, Computers & Operations Research, (38), 328-339.

Chehade. H., Amodeo. L., Yalaoui,F., “A new efficient hybrid method for selecting machines and sizing buffers in assembly lines”, Journal of Operations and Logistics, (16), 2, III.1-III.22.

Daoud. S, Amodeo. L, Yalaoui. F, Chehade. H, Duperray. P. 2012, “New mathematical model to solve robotic assembly lines balancing”, In Proceedings of INCOM-IFAC conference, (Buccarest, Romanie).

Degertekin. S.O. 2012, “ Improved harmony search algorithms for sizing optimization of truss structures”, Computers & Structures, (92-93), 229-241.

Gao. J, Sun. L, Wang. L, Gen. M. 2009, “An efficient approach for type II robotic assembly line balancing problems”, Computers & Industrial Engineering, (56), 1065-1080.

Khouja. M, Booth. DE, Suh. M, Mahaney. JK. 2000, “Statistical procedures for task assignment and robot selection in assembly cells”, International Journal of Computer Integrated Manufacturing, (13), 95-106.

Kilincci. O. 2011, “Firing sequences backward algorithm for simple assembly line balancing problem of type 1”, Computers & Industrial Engineering, (60), 830-839.

Lee. K.S, Geem. Z. W. 2005, “A new metaheuristic algorithm for continuous engineering optimization: harmony search theory and practice”. Comput. Methods Appl. Mech . Eng, (194), 3902-3933.

Levitin. G, Jacob. Rubinovitz, and Boris. Schnits. 2006, “ A genetic algorithm for robotic assembly line balancing”. European Journal of Operational Research, (168), 811-825.

Nai-Chieh. W I-Ming Chao. 2011, “A solution procedure for type E simple assembly line balancing problem”, Computers&IndustrialEngineering, (61), 824-830.

Nicosia. G, Paccarelli. D, Pacifici. A. 2002, “ Optimally balancing assembly lines with different workstations”, Discrete Apllied Mathematics, (118), 99-113.

Rubinovitz. J and J. Bukchin. 1991, “Design and balancing of robotic assembly lines”. In Proceedings of the fourth world conference on on robotics research”, Pittsburgh, PA.

Scholl. A, Becker. C. 2006, “State-of-the-art exact and heuristic solution procedures for simple assembly line balancing”, European Journal of Operational Research, (168), 666–693.

Yoosefelahi. A, Aminnayeri. M, Mosadegh. H, Davari Ardakani. H. 2011, “Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model”, Journal of Manufacturing Systems, doi:10.1016/j.jmsy.2011.10.002.

Zou. D, Gao. L, Li. S, Wu. J, Wang. X. 2010, “A novel global harmony search algorithm for task assignment problem”, Journal of Systems and Software , (83), 1678-1688.

BIOGRAPHIES

Slim Daoud is a PhD student in the University of Technology of Troyes in France and a research engineer in Aries Packaging in France. His PhD thesis concerns the optimization of robotic assembly lines. He received his engineering degree from the National School of Engineering of Tunis (Tunisie), followed by his master's degree in Industrial Engineering and Management in 2009 from Polytechnics Institute of Grenoble.

Hicham Chehade received his engineering degree in Industrial Systems Engineering and his master's degree in Systems Optimization and safety in 2005, followed by his Ph.D. degree in Systems Optimization and safety from the University of Technology of Troyes (UTT, France) in 2009. He is currently an assistant professor in the Industrial Systems Optimization Laboratory (ICD - LOSI, UMR CNRS 6279) in the University of Technology of Troyes in France. His research fields concerns mainly the optimization of production, assembly and robotic lines design. He has also other research topic such as scheduling problems, operations research, modeling and simulation.

Farouk Yalaoui obtained his degree in Industrial Engineering from the Polytechnics School of Algiers in 1995, his master's degree in Industrial System Engineering from Polytechnics Institute of Lorraine in 1997, his Ph.D. degree in Production Management from the Troyes University of Technology (UTT) in 2000 and followed by a Habilitation à diriger les recherches from Compiegne University of Technology in 2006. He is currently a full Professor at UTT, where he is head of Master program and head of Production management and Filed 2 of OSI-ICD team. His research topic focuses on scheduling problems, system design, operations research, modeling, analysis and optimization of logistic and production systems, reliability and maintenance optimization. He is author of more than 180 publications or communications with one book, 7 book chapters and 38 articles in journals. He also published more than 110 papers in conference proceedings. He is a member of several international conference committees and International journal Boards.

Lionel Amodeo received the Ph.D. degree in automatic and production management from the University of Technology of Belfort-Montbeliard, France, in 1999. Since 2000, he has been with the Technology University of Troyes where he is recently a full Professor at the Industrial Systems Optimization Department. He is currently in Head of the Engineer Degree in Industrial Systems. His research interests include logistic and production systems optimization, scheduling and facility layout problems. He has published one book, 7 book chapters and over 140 papers in technical journals and conference proceedings. He was finalist of the Kayamori best paper award of the 2002 IEEE International Conference on Robotics and Automation, Washington. He organized and chaired over 10 sessions/tracks for many conferences including IEEE and he served as Program Committee member in several International Conferences including IEEE.

A NEW FORMULATION OF A BUFFERED TWO-WORKSTATION PRODUCTION LINE WITH PARALLEL UNRELIABLE MACHINES

Yassine Ouazene
Alice Yalaoui
Hicham Chehade
Farouk Yalaoui
Institut Charles Delaunay - Laboratoire d'Optimisation des Systèmes Industriels
UMR-STMR-CNRS 6279, Université de Technologie de Troyes
12 rue Marie Curie,10010 Troyes Cedex, France.
email: {yassine.ouazene, alice.yalaoui, hicham.chehade, farouk.yalaoui}@utt.fr

KEYWORDS

Manufacturing systems, Throughput calculation ,Unreliable parallel machines, Replacing techniques, Queuing theory

ABSTRACT

This paper presents a new mathematical formulation to calculate the throughput of a buffered two-workstation manufacturing system with unreliable parallel machines on each stage. The parallel machines of each workstation are not necessarily identical. Both failure and repair times of the parallel machines are assumed to be exponentially distributed. Based on the technique of replacing a parallel-machine workstation by a single equivalent machine, we obtain a simple two-machine one-buffer line. Two existing replacing techniques are considered and discussed. Many analytical formulations for the simple two-machine one-buffer have been presented in literature. So, in this paper, we consider an analytical formulation based on the construction and analysis of an equivalent machine using a birth-death markov process. The obtained mathematical model is independent from the adopted replacing technique and can be easily solved. Numerical experiments based on a comparison using the benchmark presented by Diamantidis and Papadopoulos (2009) prove the applicability and the effectiveness of the presented formulation.

INTRODUCTION

The performance of a production line is often expressed according to its global production rate or throughput. Liang (2009) makes a light distinction between these two concepts. He has defined the system production rate as the average number of parts produced by the last machine per cycle of time in the steady state of system operation and the throughput as the average number of parts per unit of time. Patchong and Willaeys (2001) have presented the three means to increase the throughput of a manufacturing system which are: increase the intermediate storage capacity; improve the availability of machines and increase the processing capacity of machines. The effects of these possible alternatives on the system throughput are limited.

The parallel manufacturing structure is one of the solutions of increasing both availability and processing capacity of a workstation. So, the production line throughput may be increased by adding parallel machines on workstations. Comparing the vast literature on the analysis of production lines with a single machine on each workstation, the publications considering series-parallel configuration are far and few between.

Some series-parallel models exist in literature. Bergeron et al. (2010) have presented a state of art of manufacturing system models with unreliable machines and buffers especially in series-parallel context. The analytical methods to evaluate the throughput of parallel systems are limited (Li (2004)). One of the earliest researches cited in literature is that of Ignall and Silver (1977). The authors have studied the output of a series-parallel model where each stage consists of identical parallel machines. They proposed an approximate method based on replacing the identical machines on each stage by a single equivalent machine with a production capacity equal to the sum of the individual capacities of the parallel machines. Ancelin and Semery (1987), Burman (1995) and Patchong and Willaeys (2001) have also studied the series-parallel production lines. They have replaced the multiple machines on each stage by an equivalent machine. In the first paper, the authors have proposed a discrete state model. In the second, Burman (1995) has developed a continuous operations-dependent-failure model, he has approximated the discrete flow of parts by a continuous flow. His equivalent workstation has a maximum processing rate equals to the sum of the processing rates of the parallel machines. Failure and repair rates are calculated assuming that all parallel machines operate independently. Patchong and Willaeys (2001) have proposed a similar technique easier to implement than Burman's one. They have also stated that their replacing approach is more realistic in some identified cases than Ancelin and Semery's method. Li (2004) has considered parallel systems with intermediate buffers. He has presented a recursive aggregation method which decomposes the parallel system into overlapping serial lines to approximate the performance of such system. The author has proved analytically the convergence of the procedure and the uniqueness of the solution.

The two-machine line models are often used as a building block to analyze and develop approximation methods for wider and more complex production systems. Therefore, various simple two-machine models have been presented in literature. Li et al. (2006) have presented a comparative study with eight different models. The authors have presented the major differences and their impacts among these models, how they differ in system performance and their common nature. Tan and Gershwin (2009) have presented a general Markovian model for a buffered two-stage continuous-flow production line. They have modelled the system as a continuous time, continuous-discrete state space stochastic process. The system production rate and the buffer level performances are determined based on the steady state distribution. Ouazene et al. (2011) have also presented two-machine-one-buffer line model considering an operation-dependent-failure context. The system throughput is calculated after modeling the buffer behavior as a birth-death process.

Diamantidis et al. (2007) have studied a serial flow line with identical reliable parallel machines on each workstation. Initially, the authors have solved analytically a two-station one-buffer model by developing a recursive algorithm for generating the buffer states transition matrix. Then, they used this model as a decomposition block for solving larger lines. The authors have validated their model by comparison with a simulation approach. Diamantidis and Papadopoulos (2009) have presented an exact Markovian analysis of a two-workstation one-buffer with unreliable parallel machines manufacturing system. The machines states are described using a binary variables (0 and 1) corresponding to down and up states. The transition equation of the Markovain model are solved analytically and the steady state probabilities are calculated using the LU decomposition technique. Liu et al. (2012) have proposed a similar study with considering a three-state model to characterize the asynchronous independent-operations with multiple parallel machines. They have stated on the importance of using more than two states (down and up) to model parallel unreliable machines production line. The system state probability and throughput were calculated based on solving a generic Quasi-Birth-Death process. This model was used by Liu and Wu (2011) as a building block for an approximate decomposition method for the analysis of production lines with multiple stations which consist of parallel unreliable machines.

The rest of this paper is organized as follows: the next section presents the problem formulation; the adopted assumptions and notations are described. Section 3 presents the two replacing techniques considered. Section 4 presents the study of the simple two-machine one-buffer model obtained by the replacing techniques. Section 5 presents the numerical experiments and the discussion of the obtained results. Finally, section 6 summarizes the contribution of this paper and presents some perspectives.

PROBLEM FORMULATION

In this paper, we consider a production system with two workstations separated by an intermediate buffer. Each workstation consists of multiple unreliable parallel machines as illustrated in figure 1. The system is subject to unavailability due to the finite buffer and/or to the the machines failure and repair times. The processing times (or machining times) of products on the machines are deterministic. Both failure and repair times are exponentially distributed and statistically independent.

The following assumptions are also considered:

- Failures are state dependent. A machine cannot fail when it is starved or blocked.
- The first station can not be starved and the second one can not be blocked.
- The buffer does not fail and transition times are equal to zero.

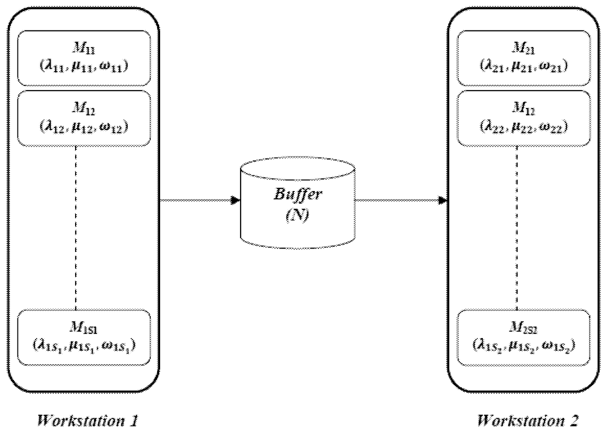


Figure 1: A two-machines one-buffer production line with parallel machines

We, also, adopt the following notations:

N	Capacity of the buffer
j	Stage index
i	Machine index

S_j	Number of machines on the stage j
ω_i	Processing rate of machine M_i
λ_i	Average failure rate of machine M_i
μ_i	Average repair rate of machine M_i
ω_{S_j}	Equivalent production rate of workstation j
λ_{S_j}	Equivalent average failure rate of workstation j
μ_{S_j}	Equivalent average repair rate of workstation j
γ_{S_j}	Efficiency ratio of workstation j

EQUIVALENT MACHINE REPLACING TECHNIQUES

To calculate the system throughput, parallel machines on each stage are replaced by a single equivalent machine. This technique has been introduced for the first time by Ancelin and Semery (1987). Burman (1995) has also presented this equivalent machine replacing technique. He has used a method based on three different equations to estimate production rate, failure and repair rates of the equivalent machine of each workstation. The first equation matches the variability of the output of the equivalent machine with that of the original set of parallel machines. The second equation matches excepted throughput and the third matches the maximum production rate. These two replacing techniques are reported in Patchong and Willaeys (2001). The authors have presented also a novel replacing method.

Ancelin and Semery's replacing method (ASRM)

The equivalent processing rate of a set of parallel machines is defined as the sum of the processing rates of all the machines:

$$\omega_{S_j} = \sum_{i=1}^{S_j} \omega_i \tag{1}$$

The equivalent failure rate of a stage j is given by the following equation:

$$\lambda_{S_j} = \sum_{i=1}^{S_j} (\lambda_i \times \prod_{k=1,k\neq i}^{S_j} \frac{\mu_k}{\lambda_k + \mu_k}) \tag{2}$$

The equivalent repair rate of a stage j is derived from equations 2 and 3 as follows:

$$\frac{\mu_{S_j}}{\lambda_{S_j} + \mu_{S_j}} = \frac{\sum_{i=1}^{S_j} \omega_i \times \frac{\mu_i}{\lambda_i+\mu_i}}{\sum_{i=1}^{S_j} \omega_i} \tag{3}$$

After some simplifications, we obtain:

$$\mu_{S_j} = \frac{\sum_{i=1}^{S_j} \omega_i \times \frac{\mu_i}{\lambda_i+\mu_i}}{\sum_{i=1}^{S_j} \omega_i \times \frac{\lambda_i}{\lambda_i+\mu_i}} \times \sum_{i=1}^{S_j} (\lambda_i \times \prod_{k=1,k\neq i}^{S_j} \frac{\mu_k}{\lambda_k + \mu_k}) \tag{4}$$

Patchong and Willaeys 's replacing method (PWRM)

Patchong and Willaeys (2001) have stated that the equivalent parameters yielded by the method cited above are often unrealistic and meaningless. So, they have presented another method based on the idleness and efficiency states characterization. Therefore, it is assumed that the equivalent machine is considered to be down if at least one of the parallel machines is down. This assumption is introduced to take into account the slowing down of processing rate of the equivalent machine if a subsequent failure of a parallel machine on the stage occurs.
The behavior of the equivalent machine is characterized by a three-state Markov chain: up, down and idle. Considering the balance equations in the operation-dependent-failure case we obtain the following relationship:

$$\lambda_{S_j} \times P_{w_j} = \mu_{S_j} \times P_{d_j} \tag{5}$$

Such as:

$$P_{w_j} = \frac{\sum_{i=1}^{S_j} \omega_i \times \frac{\mu_i}{\lambda_i+\mu_i}}{\sum_{i=1}^{S_j} \omega_i} \tag{6}$$

$$P_{d_j} = \frac{\sum_{i=1}^{S_j} \omega_i \times \frac{\lambda_i}{\lambda_i+\mu_i}}{\sum_{i=1}^{S_j} \omega_i} \tag{7}$$

P_{w_j} : is the probability that the equivalent machine of the stage j is up.
P_{d_j} : is the probability that the equivalent machine of the stage j is down.

The equivalent processing rate of a set of parallel machines on stage j is defined as the sum of the processing rates of all the machines when they are working normally (without failure):

$$\omega_{S_j} = \sum_{i=1}^{S_j} \omega_i \tag{8}$$

The equivalent failure rate λ_{S_j} is given by the following equation:
$\lambda_{S_j} = \frac{1}{P_{w_j}} \times \sum_{i=1}^{S_j} (\lambda_i \times \frac{\mu_i}{\lambda_i+\mu_i} \times \prod_{k=1,k\neq i}^{S_j} \frac{\mu_k}{\lambda_k+\mu_k})$

$$= \frac{\sum_{i=1}^{S_j} \omega_i}{\sum_{i=1}^{S_j} \omega_i \times \frac{\mu_i}{\lambda_i+\mu_i}} \times \sum_{i=1}^{S_j} (\lambda_i \times \frac{\mu_i}{\lambda_i + \mu_i} \times \prod_{k=1,k\neq i}^{S_j} \frac{\mu_k}{\lambda_k + \mu_k}) \tag{9}$$

The equivalent repair rate μ_{S_j} can be derived from equations 5 and 9 as follows:
$\mu_{S_j} = \lambda_{S_j} \times \frac{P_{w_j}}{P_{d_j}}$

$$= \frac{\sum_{i=1}^{S_j} \omega_i}{\sum_{i=1}^{S_j} \omega_i \times \frac{\lambda_i}{\lambda_i+\mu_i}} \times \sum_{i=1}^{S_j} (\lambda_i \times \frac{\mu_i}{\lambda_i + \mu_i} \times \prod_{k=1,k\neq i}^{S_j} \frac{\mu_k}{\lambda_k + \mu_k}) \tag{10}$$

ANALYSIS OF THE EQUIVALENT TWO-MACHINE ONE-BUFFER MODEL

Based on the replacing methods described in section 3, we obtain a simple two-machines one-buffer model(see figure 2). The throughput of the original system is equivalent to that of this simple model calculated using the reliability parameters of the equivalent machines $(\omega_{S_1}, \lambda_{S_1}, \mu_{S_1})$ and $(\omega_{S_2}, \lambda_{S_2}, \mu_{S_2})$.

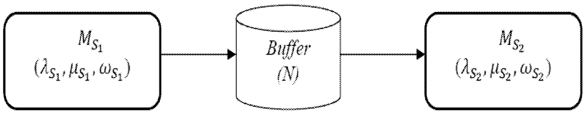


Figure 2: Equivalent two-machine one-buffer simple model

In this paper, we present a mathematical formulation based on the construction of two equivalent machines. Considering the buffer as an isolated system between two reliable machines, its states have been analyzed by a Markov process. To do so, we consider a birth-death process with $(N + 1)$ states $\{0, 1, ..., N\}$ and ω_{S_1}, ω_{S_2} are respectively the birth and the death transition rates (see figure 3). In order to calculate the effective production rate of each machine, Songgui and Yao (1992) and Songgui and Changjun (1997) have defined for each one an equivalent machine without starvation, blockage and free from failures. The main idea is to replace each machine by an equivalent one that has only up and down states. The blockage and starvation states are integrated on the up state.

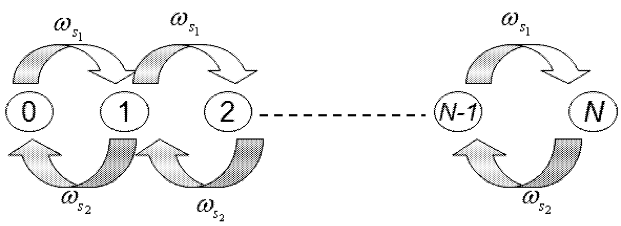


Figure 3: Buffer states transitions

The differential equations for the probability that the system is in state j at time t are:

$$\begin{cases} P_0^{'}(t) = -\omega_{S_1} P_0(t) + \omega_{S_2} P_1(t) \\ P_j^{'}(t) = \omega_{S_1} P_{j-1}(t) - (\omega_{S_1} + \omega_2) P_j(t) + \omega_{S_2} P_{j+1}(t) \\ P_N^{'}(t) = \omega_{S_1} P_{N-1}(t) + \omega_{S_2} P_N(t) \end{cases} \tag{11}$$

For the steady state of system operation, the probability to be in the jth state of the buffer is given by equation (12). It is obtained by simplifying the system of equations above and considering that $\sum_{j=1}^{N} P_j = 1$.

$$P_j = \begin{cases} \frac{\alpha^j \times (1-\alpha)}{1-\alpha^{N+1}} & \text{if } \alpha \neq 1 \\ \frac{1}{N+1} & \text{if } \alpha = 1 \end{cases} \tag{12}$$

$\alpha = \frac{\omega_{S_1}}{\omega_{S_2}}$ is the processing rates ratio.

The construction and analysis of the equivalent machine is based on the analysis of the possible states of the buffer, especially the empty and full buffer states. The system throughput ψ is defined as the bottleneck between the effective production rates of the two machines(Ouazene et al. (2011)).

$$\rho_{S_1} = \begin{cases} \frac{\omega_{S_1} \times \mu_{S_1} \times \frac{1-\alpha^N}{1-\alpha^{N+1}}}{\mu_1 + \lambda_{S_1} \times \frac{1-\alpha^N}{1-\alpha^{N+1}}} & \text{if } \alpha \neq 1 \\ \frac{\omega \times \mu_{S_1} \times \frac{N}{N+1}}{\mu_{S_1} + \lambda_{S_1} \times \frac{N}{N+1}} & \text{if } \alpha = 1 \end{cases} \tag{13}$$

$$\rho_{S_2} = \begin{cases} \frac{\omega_{S_2} \times \mu_{S_2} \times \frac{\alpha(1-\alpha^N)}{1-\alpha^{N+1}}}{\mu_{S_2} + \lambda_{S_2} \times \frac{\alpha \times (1-\alpha^N)}{1-\alpha^{N+1}}} & \text{if } \alpha \neq 1 \\ \frac{\omega \times \mu_{S_2} \times \frac{N}{N+1}}{\mu_{S_2} + \lambda_{S_2} \times \frac{N}{N+1}} & \text{if } \alpha = 1 \end{cases} \tag{14}$$

$$\psi = \min\{\rho_{S_1}, \rho_{S_2}\}$$

$$= \begin{cases} \omega_{S_1} \times \min\{\frac{1}{\frac{(1-\alpha^{N+1})}{(1-\alpha^N)} + \frac{\lambda_{S_1}}{\mu_{S_1}}}, \frac{1}{\frac{(1-\alpha^{N+1})}{(1-\alpha^N)} + \alpha \times \frac{\lambda_{S_2}}{\mu_{S_2}}}\} & \text{if } \alpha \neq 1 \\ \omega \times \min\{\frac{1}{\frac{\lambda_{S_1}}{\mu_{S_1}} + \frac{N+1}{N}}, \frac{1}{\frac{\lambda_{S_2}}{\mu_{S_2}} + \frac{N+1}{N}}\} & \text{if } \alpha = 1 \end{cases} \tag{15}$$

Finally, the system throughput can be written as follows:

$$\psi = \begin{cases} \omega_{s_1} \times \frac{1}{\frac{1-(\alpha)^{N+1}}{1-(\alpha)^N} + max(\frac{\lambda_{s_1}}{\mu_{s_1}}, \alpha \times \frac{\lambda_{s_2}}{\mu_{s_2}})} & \text{if } \alpha \neq 1 \\ \omega \times \frac{1}{\frac{N+1}{N} + max(\frac{\lambda_{s_1}}{\mu_{s_1}}, \frac{\lambda_{s_2}}{\mu_{s_2}})} & \text{if } \alpha = 1 \end{cases} \tag{16}$$

NUMERICAL EXPERIMENTS

In order to evaluate the performance and the effectiveness of the proposed model, we have tested it among two test configurations presented by Diamantidis and Papadopoulos (2009). In the first configuration, the first workstation consists of two identical parallel machines and the second one consists of one machine. All the machines are supposed to be identical and the size of the buffer varies from 0 to 10. In the second configuration, both workstations consist of three identical parallel machines and the size of the buffer varies from 10 to 100. The parameters of these two examples are reported in table 1.

Configuration 1		Configuration 2	
WS1	WS2	WS1	WS2
$S_1 = 2$	$S_2 = 1$	$S_1 = 3$	$S_2 = 3$
$\omega_{11} = \omega_{12} = 1$	$\omega_{21} = 1$	$\omega_{1i} = 1$	$\omega_{2i} = 1$
$\lambda_{11} = \lambda_{12} = 0.01$	$\lambda_{21} = 0.01$	$\lambda_{1i} = 0.01$	$\lambda_{2i} = 0.01$
$\mu_{11} = \mu_{12} = 0.1$	$\mu_{11} = 0.1$	$\mu_{1i} = 0.1$	$\mu_{1i} = 0.1$

Table 1: Parameters of the test configurations

The evaluation is made up by comparing the system throughput calculated by the proposed model (ψ_2), using the two replacing techniques described in section 3, with that presented in Diamantidis and Papadopoulos (2009) (ψ_1). This comparison is illustrated by the calculation of the relative difference (gap) between the two models using the equation 17. The computational times of both models are also reported.

$$gap(\%) = 100 \times \frac{| \psi_2 - \psi_1 |}{max(\psi_2, \psi_1)} \tag{17}$$

Tables 2 and 3 present the obtained system throughput and the computational times of the proposed model respectively in the first and the second test configurations. The system throughput calculated by the proposed model is increasing according to the buffer size (which is predictable). We note that the proposed model over-estimates the system throughput in both configurations comparing to the model proposed by Diamantidis and Papadopoulos (2009). This light difference can be explained by the difference on the formulation approaches adopted.
The proposed model obtains the same system throughput with Ancelin and Semery's and Patchong and Willaeys 's replacing techniques. So, it is important to note that this formulation is independent on the choice of the replacing technique. This model obtains the same system throughput with the two different replacing techniques because they provide identical equivalent processing rates ω_{S_j} and identical efficiency ratios $\gamma_{S_j} = \frac{\lambda_{S_j}}{\mu_{S_j}}$ for each workstation. An important remarque also is that the computational time of the proposed approach is less important and constant according to the buffer size comparing with the method presented by Diamantidis and Papadopoulos (2009).

CONCLUSION

In this paper, we have presented a new mathematical formulation to calculate the system throughput of a buffered two-workstation with parallel unreliable machines production line in an operation-dependent-failure context. This mathematical formulation is based on the transformation of the initial model on a simple two-machine one-buffer line using the technique on replacing a parallel machines stage by an equivalent machine. The behavior of the intermediate buffer is analyzed using a birth-death process. The performance of the proposed model is compared with that of Diamantidis and Papadopoulos (2009). One of the important aspects of this model is its reduced computation times. Therefore, the resolution of the obtained equations is independent of the intermediate buffer transitions because this formulation approach considers only the empty and full buffer states.
Future extension of this work may be to consider the presented model as a building block to estimate the system throughput for wider and more complex production lines with parallel unreliable machines.

REFERENCES

Ancelin B. and Semery A., 1987. *Calcul de la productivité dune ligne intégrée de fabrication: CALIF, une méthode analytique industrielle. RAIRO APII*, 21, no. 3, 209–238.

Bergeron D.; Jamali M.; and Yamamoto H., 2010. *Modelling and analysis of manufacturing systems: a review of existing models. International Journal of Product Development*, 1/2/3, 46–61.

Burman M., 1995. *New Results in Flow Line Analysis.* Phd-Thesis Electrical Engineering and Computer Science: Massachusetts Institute of Tecnology.

Diamantidis C. and Papadopoulos T., 2009. *Exact analysis of a two-workstation one-buffer flow line with parallel unreliable machines. European Journal of Operational Research*, 197, 572–580.

Diamantidis C.; Papadopoulos T.; and Heavy C., 2007. *Approximate analysis of serial flow lines with multiple parallel-machine stations. IIE Transactions*, 39, 361–375.

Ignall E. and Silver A., 1977. *The output of a two-stage system with unreliable machines and limited storage. AIIE Transactions*, 9, 183–188.

Li J., 2004. *Modeling and Analysis of Manufacturing Systems With Parallel Lines. IEEE Transactions on automatic control*, 49, no. 10, 1824–1829.

Buffer size	ψ_1	CPU time(seconds)	ψ_2^{PWRM}	ψ_2^{ASRM}	CPU time(seconds)	*Gap*(%)
0	0.7848	0.141	0.8264	0.8264	0.015	5.03
2	0.8374	0.219	0.7895	0.7895	0.015	5.72
4	0.8502	0.234	0.8824	0.8824	0.015	3.65
6	0.8540	0.219	0.9026	0.9026	0.015	5.38
8	0.8554	0.234	0.9075	0.9075	0.015	5.74
10	0.8561	0.250	0.9087	0.9087	0.015	5.79

Table 2: Performances of the proposed model in the first configuration

Buffer size	ψ_1	CPU time(seconds)	ψ_2^{PWRM}	ψ_2^{ASRM}	CPU time(seconds)	*Gap*(%)
10	2.4607	14.485	2.5000	2.5000	0.031	1.57
20	2.5355	58.125	2.6087	2.6087	0.031	2.81
30	2.5747	152.453	2.6471	2.6471	0.031	2.74
40	2.5998	320.140	2.6667	2.6667	0.031	2.51
50	2.6175	613.437	2.6786	2.6786	0.031	2.28
60	2.6306	1089.910	2.6866	2.6866	0.031	2.08
70	2.6408	1710.000	2.6923	2.6923	0.031	1.91
80	2.6490	3003.130	2.6966	2.6966	0.031	1.77
90	2.6556	5389.390	2.7000	2.7000	0.031	1.64
100	2.6612	9166.800	2.7027	2.7027	0.031	1.54

Table 3: Performances of the proposed model in the second configuration

Li J.; Blumenfeld D.; Huang N.; and Alden J., 2006. *comparisons of two-machine line models in throughput analysis. International Journal of Production Research*, 44, no. 7, 1375–1398.

Liang Z., 2009. *System-theoretic properties of production lines.* Phd-Thesis Electrical Engineering: Systems in The University of Michigan.

Liu J. and Wu A., 2011. *Analytical anlysis of production lines with parallel machines.* In *30th Chinese Control Conference*. 5480–5483.

Liu J.; Yang S.; Wu A.; and Hu S., 2012. *Multi-state throughput analysis of a two-stage manufacturing system with parallel unreliable machines and a finite buffer. European Journal of Operational Research*, 219, 296–304.

Ouazene Y.; Chehade H.; and Yalaoui A., 2011. *Approches mono et multi-objective pour la conception dun système de production à deux machines et un stock-tampon.* In *9ème Congrès International de Génie Industriel*. Saint-Sauveur, Québec, Canada.

Patchong A. and Willaeys D., 2001. *Modeling and analysis of an unreliable flow line composed of parallel-machine stages. IIE Transactions*, 33, 559–568.

Songgui S. and Changjun J., 1997. *Modeling and analysis of capacitated transfer lines with unreliable machines and deterministic processing times. Journal of Systems Science and Systems Engineering*, 6, no. 1, 53–60.

Songgui S. and Yao Z., 1992. *The modeling an analysis of the multiple stage repairable CIMS with finite buffers.* In *Priprints of IFAC/IFORS/IMACS Symposium Large Scale System*. Theory and Application.

Tan B. and Gershwin S., 2009. *Analysis of a general Markovian two-stage continuous-flow production system with a finite buffer. International Journal of Production Economics*, 120, 327–339.

BIOGRAPHY

YASSINE OUAZENE is currently a PhD-student in the Industrial Systems Optimization Laboratory (ICD - LOSI, UMR CNRS 6279) in the University of Technology of Troyes in France. His PhD-thesis concerns the optimization of production lines. He received his engineering degree in Industrial Engineering in 2009 from the Polytechnical School of Algiers (Algeria), followed by his master's degree in Systems Optimization and safety in 2010 from the University of Technology of Troyes (UTT, France).

ALICE YALAOUI received her engineering degree in Industrial Systems Engineering in 2001, followed by her Ph.D. degree in Systems Optimization and safety from the University of Technology of Troyes (UTT, France) in 2004. She is currently an assistant professor in the Industrial Systems Optimization Laboratory (ICD - LOSI, UMR CNRS 6279) in the University of Technology of Troyes in France. Her research fields concerns mainly the analysis and optimization of logistic and production systems, reliability and maintenance optimization.

HICHAM CHEHADE received his engineering degree in Industrial Systems Engineering and his master's degree in Systems Optimization and safety in 2005, followed by his Ph.D. degree in Systems Optimization and safety from the University of Technology of Troyes (UTT, France) in 2009. He is currently an assistant professor in the Industrial Systems Optimization Laboratory (ICD - LOSI, UMR CNRS 6279) in the University of Technology of Troyes in France. His research fields concerns mainly the optimization of production, assembly and robotic lines design. He has also other research topic such as scheduling problems, operations research, modeling and simulation.

FAROUK YALAOUI obtained his degree in Industrial Engineering from the Polytechnical School of Algiers in 1995, his master's degree in Industrial System Engineering from Polytechnics Institute of Lorraine in 1997, his Ph.D. degree in Production Management from the Troyes University of Technology (UTT) in 2000 and followed by a Habilitation à diriger les recherches from Compiegne University of Technology in 2006. He is currently a full Professor at UTT, where he is head of Master program and head of Production management and Filed 2 of OSI-ICD team. His research topic focuses on scheduling problems, system design, operations research, modeling, analysis and optimization of logistic and production systems, reliability and maintenance optimization. He is author of more than 180 publications or communications with one book, 7 book chapters and 38 articles in journals. He also published more than 110 papers in conference proceedings. He is a member of several international conference committees and International journal Boards.

LATE PAPERS

Generic Model for University Course Timetabling Problem solved by Genetic Algorithm

Maqsood Ahmad
Christophe Caux
Michel Gourgand

LIMOS UMR CNRS 6158; Université Blaise Pascal
Campus des Cézeaux-BP 10125, 63173 Aubière, France

E-mail: {maqsood|michel.gourgand}@isima.fr, christophe.caux@ifma.fr

Key words: Course timetabling, Genetic Algorithm, Integer programming, Combinatorial optimization.

ABSTRACT

The university course timetabling problem is a NP-hard problem which must be solved by departments in the beginning of the semester. It is a difficult task for which universities devote a large amount of human and material resources every year. We have formulated the new problem which is a generalization of International Timetabling competition 2007. We have added 10 new constraints in competition problem which comes from different university environments (details are given in the "description of the problem" section). We have presented a mathematical model for the problem. The purpose of this paper is to make a model which would be more general and could be used by different universities according to their requirements with little changes in algorithm. Algorithm is very relaxed for the use of hard and soft constraint choices because hard and soft constraints are distinguished by violation penalty given to them (hard constraints have higher penalty than soft). We have randomly generated data for small, medium and large size problems for numerical applications, because the existing data sets could not be used as problem has many more extra constraints now. The objective function values, percentage of used seats, percentage of used slots in numerical application indicates that results are promising under the cover of so many constraints.

INTRODUCTION

Timetabling can be generally defined as the activity of assigning, subject to constraints, a number of events to a limited number of time periods and locations such that desirable objectives are satisfied as nearly as possible (Wren 1996) . Practical cases where such activity arises are, among others, educational timetabling, employee timetabling, sport timetabling, transport timetabling and communication timetabling. We shall deal with educational timetabling here. Educational timetabling can be sub-divided into three main classes, which are school timetabling, course timetabling and exam timetabling (Werra 1985). In university course timetabling, a set of lectures must be scheduled into rooms and timeslots subject to constraints that are usually divided in two categories, which are hard and soft constraints.

Hard constraints must be strictly satisfied with no violation allowed, while in the case of soft constraints it is desirable, but not essential, to minimize violations. Typical characteristics of course timetabling, with respect to school and exam timetabling, are the availability of a limited number of timeslots and the requirements , allocating lectures only into suitable rooms, having no more than one lecture per room and scheduling lectures with common students in different timeslots. It is then desirable that each student has the best balance between too many or too few courses in a day.

Constraints and their importance differ significantly among various countries and institutions. A timetable is considered to be effective when it is feasible and may be realized by the institution, while it is considered to be satisfactory when it carries certain quality characteristics that keep its users satisfied at least to a certain degree. Course timetabling can be formulated as an optimization problem. Given the large number of events (lectures) to be scheduled and the diversity of constraints, the problem to be solved can be very difficult. The timetabling problem, like many others in the area of combinatorial optimization, has been approached by several well-known techniques of the operational research and the computer science fields. Several surveys on course timetabling (Carter and Laporte 1998, Schaerf 1999), as well as others on more focused aspects of the problem, have managed to record this work in a systematic way, categorizing thus the different variations of the problem and solution approaches.

REVIEW OF SIMILAR PROBLEMS AND STATE-OF-THE-ART

The timetabling problems were solved by many authors, some used real data of any university, college or high school and some for assumed data. They described an integer programming approach which has been implemented at the School of Economics and Management at Hannover University, Germany, to create the complete timetable of all courses for a term and formulation was solved with CPLEX solver (Schimmelpfeng and Helber 2007).

They have divided the timetabling problem into four sub problems which are formulated as an integer linear programming problem. The goals of the four divided sub problems are specified as: maximize the number of assigned courses with an urgency, maximize the total assigned workload, minimize the shortage of students to reach the minimum number of students of a section, optimize the timetable. The models introduced was solved by IP solver CPLEX.10.0. (Broek and hurkens 2007). They presented integer programming formulation for university timetabling problem. The timetable for the Electrical and Computer Engineering Department in the University of Patras was used as a case study (Bolanda and Daskalaki and Birbas 2004). They addressed a faculty-class assignment problem, time-slots for classes are initially assumed to be given and then integer programming model is constructed to minimize the individual and collective dissatisfaction of faculty members. Their second article main focus is to design efficient class offering patterns while taking into consideration newly imposed gender policies. The both models were solved with CPLEX (Yakoob and sherali 2006, Yakoob and sherali 2007).

They solved a timetabling problem proposed by Metaheuristic Network and sponsored by PATAT and they solved the problem with genetic algorithm. This problem was a simple problem which has a few constraints (Jat and Yang 2009). They discussed the problem of their university. They expressed the problem and proposed a genetic algorithm but did not solve any problem to test their algorithm (Jain and Chande 2010).

They present Adaptive Tabu Search algorithm to solve CB-CTT. The authors use several features such as a double Kempe chains neighborhood structure, a penalty-guided perturbation operator and an adaptive search mechanism. The algorithm was tested on International timetabling competition track curriculum based course timetabling CB-CTT instances (Zhipeng and Jin-Kao Hao 2010). They proposed memetic algorithm which hybridises a genetic algorithm with a Tabu Search Algorithm. This algorithm uses neighbourhood structures during the search process to get significant improvements in solution quality. They use International timetabling competition track curriculum based course timetabling (ITC 2007: CB-CTT) instances for numerical application (Salwani Abdullah and Hamza Turabieh 2012).

DESCRIPTION OF OUR PROBLEM

This problem is a generalization of the International Timetabling Competition 2007, track: curriculum based course timetabling problem ITC 2007:CB-CTT (International Time tabling competrition, 2007). We have added in this competition track some constraints from (Aladag and Hocaoglu 2007 ,Aladag and Hocaoglu 2009), ITC, 2002 (International timetabling competition 2002) and some more different constraints from literature to make this more general problem.

We have added some new features in the competition track which are as follows. Aladag et al and International Timetabling Competition 2007 used rooms as a permanent source (that these are always available) but in our problem rooms are unavailable on some periods. Which is always possible in real situations.

In international timetabling competition they used only capacity constraint, this article include capacity constraint with room type as well, because this is a more general case (two rooms of same capacity can belong to different types for example one is lecture room and other is computer lab).

This is the list of new constraints which have been added by us in International timetabling competition problem.

1. The number of courses scheduled in period p requiring room type x will be less than or equal to the number of rooms of type x available at period p.
2. If course is not adequate to room type, it could not be assigned to that room.
3. Course will be scheduled on its pre assigned periods.
4. The condition of maximum working days for any course must be satisfied.
5. A course (taught by teacher t), will not be scheduled if teacher is not available.
6. All the lessons of a course scheduled on any day must be adjacent to each other.
7. Each course must be scheduled less or equal than defined maximum teaching hours a day limit.
8. In class timetables, students of a class should not have a day with a single lesson.
9. For each course, it is required that it should not be scheduled after a fix maximum number of days.

10. No class can attend more than maximum daily number of teaching hours allowed for it.

In addition, interrelationships between these sets could be found by five matrices. The first matrix shows that any course is a part of which classes. The second matrix gives that any course could not be scheduled in these specific periods. The third one indicates that any course required which type of room. The fourth matrix gives that which teacher teaches which courses. The fifth matrix indicates that which teacher is teaching which classes.

ALL CONSTRAINTS FOR DEALT GENERALIZED PROBLEM

Timetabling problem has two types of constraints hard and soft. Hard constraints fulfillment is obligatory and soft constraints violation is permissible but their fulfillment show the quality of the solution. There is a list of hard and soft constraints used for our problem.

m_i Hard constraints

m_1 Every course must be scheduled for a specific number of hours a week.

m_2 Any class cannot attend more than one course at any time period.

m_3 Any teacher cannot teach more than one course at any time period.

m_4 Any room cannot host more than one course at any time period.

m_5 If any course *c* is scheduled in any day*,* the number of scheduled hours of the course must be less or equal than defined maximum teaching hours a day limit.

m_6 A course will not be scheduled if room is not available.

m_7 A course (taught by teacher *t*), will not be scheduled if teacher is not available or Course will not be scheduled on its forbidden periods.

m_8 The number of courses scheduled in period *p* requiring room type *x* will be less than or equal to the number of rooms of type *x* available at period *p.*

m_9 If course is not adequate to room type, it could not be assigned to that room.

m_{10} Course will be scheduled on its pre assigned periods.

m_{11} The condition of maximum working days for any course must be satisfied.

Soft constraints (n) number n in bracket shows penalty for one violation

m_{12} Each course must be spread into the given minimum number of working days. Each day below the minimum days counts one violation. (5)

m_{13} The number of students that attend the course must be less or equal than the capacity of room that hosts it. Each student above the capacity counts one violation. (1)

m_{14} All the courses of a class on any day must be adjacent to each other and any course not adjacent to any other counts one violation. (2)

m_{15} All the lessons of a course scheduled on any day must be adjacent to each other and any lesson not adjacent to any other counts one violation. (1)

m_{16} All lessons of a course must be scheduled in the same room and each distinct room used for the lessons of course, but the first, counts one violation. (1)

m_{17} In class timetables, students of a class should not have a day with a single lesson. (2)

m_{18} For each course, it is required that it should not be scheduled after a fix maximum number of days. Each occurrence above this number counts one violation. (1)

m_{19} No class can attend more than maximum daily number of teaching hours allowed for it. Each occurrence above this number counts one violation. (1)

GENERAL FEATURES REQUIRED FOR MATHEMATICAL MODEL

This section we would define sets, parameters, decision variables and present in detail the problem requirements.

- A set of course $C=\{1,\ldots,\bar{c}\}$. Each course has a fixed number of hours per week, which can be taught during the week.
- A set of rooms $R=\{1,\ldots,\bar{r}\}$. Which can be of different types, including computer rooms and laboratories for specific subjects. Each room has a fixed capacity.
- A set of periods $P=\{1,\ldots,\bar{p}\}$. *P* is a set of time periods (all of the same length, 1 hour).
- A set of days $D=\{1,\ldots,\bar{d}\}$. *D* is the teaching days of the week.
- A set of classes $K=\{1,\ldots,\bar{k}\}$. *K* is a set of classes (classes are groups of students attending exactly the same courses).
- A set of teachers $T=\{1,\ldots,\bar{t}\}$. Each course will have a teacher previously assigned to it.
- A set of types of rooms $X=\{1,\ldots,\bar{x}\}$.

n_c = Number of teaching hours to be scheduled per week for every course $c \in C$

$n_{c\max}$ = Maximum daily number of teaching hours (no less than $n_{c\max}$ teaching hours have to be assigned to the day *d*, if the course *c* is scheduled in day *d*)

t_d = First time slot of the morning in day *d*

$C_k \subset C$, set of the courses that the class *k* should attend for every class *k*. Unlike the School Timetabling problem, some classes can be joined to attend the same courses, i.e., $C_{k1} \cap C_{k2} \neq \Phi$.

$C_t \subset C$, subset of courses taught by teacher *t* for every teacher *t*

$l_{\max}$ = The maximum daily number of teaching hours allowed for any class $k \in K$

$P_r \subseteq P$, set of periods for which room *r* is available, for every $r \in R$

$P_t \subseteq P$, set of periods for which teacher *t* is available, for every $t \in T$

$C_x \subset C$, set of courses requiring rooms of type *x*, for every $x \in X$

$R_x \subset R$, set of rooms of type *x*, for every $x \in X$

m_{xp} = Number of rooms of type *x* available at period *p*, for every $p \in P$

$P_c \subset P$, set of pre assigned periods for course *c*, for every $c \in C$

$P_d \subset P$, set of periods for day *d*, for every $d \in D$

$A \subset C$, set of pre assigned courses

$\overline{P}_c \subset P$, set of forbidden periods for course *c*, for every $c \in C$

$F \subset C$, set of courses for which periods are forbidden

$\min_{wc}$ = Minimum working days for course *c*

$\max_{wc}$ = Maximum working days for course *c*

s_c = Number of students in course *c*

a_r = capacity of room *r*

To define the formulation we need three sets of binary variables.

- x_{crp} = 1 if course $c \in C$ is scheduled in room $r \in R$ at time period $p \in P$, x_{crp} = 0 otherwise;
- u_{cd} = 1 if course $c \in C$ is assigned to the day $d \in D$, 0 otherwise;
- z_{rc} = 1 if room $r \in R$ is used by course $c \in C$, 0 otherwise;

OBJECTIVE FUNCTION

We propose this formulation for objective function. The minimum working days for course *c*, room capacity, each course use minimum rooms, compactness for course *c*, compactness for class *k*, limit on class maximum teaching hours, students of a class should not have a day with a single lesson are used as soft constraints. Now multi objective function for these soft constraints can be written in this way.

Minimum working days violation for every course $c \in C$

$$q_c = \min{}_{wc} - \sum_{d \in D} u_{cd} \geq 0 \qquad c \in C$$

(1)

Total number of room capacity violations are counted

$$\sum_{r \in R} \sum_{p \in P} \sum_{c \in C, s_c > a_r} x_{crp}(s_c - a_r)$$

(2)

A course should use same room for every of its lesson, how much more room were used for every *c*

$$g_c = \left(\sum_{r \in R} z_{rc}\right) - 1 \geq 0 \qquad c \in C$$

(3)

For every course $c \in C$, how many violations are there for being compact

$$m_{cd} = \sum_{r \in R} (x_{crp_1} - x_{crp_2} + x_{crp_3}) - 1 \geq 0$$

$c \in C,\ d \in D$ (4)

$$t_d \leq p_1 < p_2 < p_3 < t_{d+1}$$

For every class $k \in K$, how many violations are there for being compact

$$y'_{kd} = \sum_{c \in C_k} \sum_{r \in R} (x_{crp_1} - x_{crp_2} + x_{crp_3}) - 1 \geq 0$$

$k \in K,\ d \in D$ (5)

$$t_d \leq p_1 < p_2 < p_3 < t_{d+1}$$

No class can attend more than $l_{\max}$ teaching hours a day, violations of any class on any day could be counted

$$b_{kd} = \sum_{c \in C_k} \sum_{r \in R} \sum_{t_d \leq p < t_{d+1}} x_{crp} - l_{\max} \geq 0$$

$k \in K,\ d \in D$ (6)

In class timetables, any class should not have a day with a single lesson

$$\sum_{p \in P_d} \sum_{c \in C_k} \sum_{r \in R} x_{crp} = w_{kd}$$

$k \in K,\ d \in D$ (7)

$l_{kd} = 1$ if $w_{kd} = 1$ and $l_{kd} = 0$ if $w_{kd} \neq 1$

Maximum working days violation for every course $c \in C$

$$v_c = \sum_{d \in D} u_{cd} - \max{}_{wc} \geq 0$$

$c \in C$ (8)

So our multi objective function finally gets this shape

Min z = f $\sum_{c\in C} q_c$ + h $\sum_{r\in R}\sum_{p\in P}\sum_{c\in C, s_c>a_r} x_{crp}(s_c - a_r)$ + i $\sum_{c\in C} g_c$ + w $\sum_{c\in C}\sum_{d\in D} m_{cd}$ + m $\sum_{d\in D}\sum_{k\in K} y'_{kd}$ + x $\sum_{d\in D}\sum_{k\in K} b_{kd}$ + n $\sum_{d\in D}\sum_{k\in K} l_{kd}$ + b $\sum_{c\in C} v_c$

(9)

The failure to satisfy constraint type is measured by the non-negative variable q_c, g_c, m_{cd}, y'_{kd}, b_{kd}, l_{kd}, v_c and is penalized via fixed parameters f >0, i>0 , w>0, m>0, x>0, n>0, b>0 each respectively and h is penalized fixed parameter for capacity constraint. Which constraint is more agreeable than other, these penalty parameters will decide.

FORMULATION EXPRESSING HARD CONSTRAINTS

In this section, we shall present our mathematical formulation. All the constraints presented in formulation are used as hard constraints. The soft constraints are written in objective function.

For every course $c \in C$, n_c hours a week must be scheduled.

$$\sum_{r\in R}\sum_{p\in P} x_{crp} = n_c \qquad c \in C \qquad (10)$$

For every class $k \in K$, class k cannot attend more than one course at time period $p \in P$.

$$\sum_{c\in C_k}\sum_{r\in R} x_{crp} \le 1 \qquad k \in K, p \in P \qquad (11)$$

For every teacher $t\in T$, teacher t cannot teach more than one course at time $p \in P$.

$$\sum_{c\in C_t}\sum_{r\in R} x_{crp} \le 1 \qquad p \in P,\ t\in T \qquad (12)$$

For every room $r\in$ R , room r cannot host more than one course at time $p \in P$.

$$\sum_{c\in C} x_{crp} \le 1 \qquad p \in P, r \in R \qquad (13)$$

For every course $c \in C$, if course c is scheduled in day d, i.e., u_{cd} = 1 the

course c must not be scheduled more than $n_{c\max}$ teaching hours.

$$\sum_{t_d<p<t_{d+1}}\sum_{r\in R} x_{crp} \le n_{c\max}\, u_{cd}$$

$$c \in C\ , d \in D \qquad (14)$$

For every course $c\in$ C, course will not be scheduled in period p at room r, if room r will not be available at period p.

$$x_{crp} = 0 \qquad c \in C, r \in R, P\in P \setminus P_r \qquad (15)$$

For every course $c\in$ C (taught by teacher t), course will not be scheduled in period p, if teacher t will not be available at period p (this constraint could be dealt adding teacher unavailable constraint in forbidden period constraint list).

$$x_{crp} = 0 \qquad t \in T, c \in C_t\ , r \in R\ , P\in P \setminus P_t$$

(16)

For every $p\in P$, the number of courses scheduled in period p requiring room type x will be less than or equal to the number of rooms of type x available at period p.

$$\sum_{r\in R_x}\sum_{c\in C_x} x_{crp} \le m_{xp}$$

$$p \in P, x \in X \qquad (17)$$

For every $p\in P$, if course is not adequate to room type, it could not be assigned to that room.

$$\sum_{r\in R_x}\sum_{c\in C-C_x} x_{crp} = 0$$

$$p \in P, x \in X \qquad (18)$$

For every course $c\in A$, courses c will be scheduled in its pre assigned periods.

$$\sum_{p\in P_c}\sum_{r\in R} x_{crp} = |P_c| \qquad c \in A \qquad (19)$$

For every course $c\in F$, that course c will not be scheduled on its forbidden period.

$$\sum_{p\in P_c}\sum_{r\in R} x_{crp} = 0$$

$$c \in F \qquad (20)$$

$x_{crp} \in \{0,1\}$ $\quad p \in P, r \in R, c \in C$, $\quad u_{cd} \in \{0,1\}$ $\quad c \in C\ , d \in D$

THE GENETIC ALGORITHM

Genetic algorithms are metaheuristics which mimics the process of natural evolution. John Holland's book "Adaptation in natural and artificial systems" as well as De Jong's "Adaptation of the behavior of a class of genetic adaptive systems," both published in 1975, are seen as the foundation of Genetic Algorithms (GAs) (Davis 1991). GAs have been used for timetabling since 1990 (Colorni et al. 1990). Since then, there are a number of papers investigating and applying GA methods for the Course timetabling problem (Carter and Laporte 1998).

Algorithm: Pseudo code for Genetic Algorithm
input : A problem instance I
set the generation counter g := 0
{initialize a random population}
while (solution_colony. population_size < **n) do**
create an empty timetable
assign the courses randomly
timetable after applying Two step Verification
Calculate the cost of timetable
Enter this timetable to the population colony
end while
while the termination condition is not reached **do**
kill costly timetables of the colony
while (solution_colony. population_size < **n) do**
Choose two parents via Roulette Wheel Selection
Randomly choose crossover points
Swap genetic material between two chromosomes.
Child solution generated by applying the crossover operator with a cross over rate
Child solution after mutation with a mutation rate
Calculate the cost of child solution
Enter this child timetable to the population colony
e**nd while**
g := g + 1
end while
output : The best achieved solution for the problem instance **I**

CHROMOSOME REPRESENTATION

The timetable is a collection of each room timetable, room timetable is a two dimensional array. If no lesson is booked in any time period, it is called null booking which has value zero. Every timetable stores information that which lesson is placed in which room at what time on which day of the week, each booking is one gene. A timetable has many fields to store information about its genetics, costs, number of violations of constraints. A population is a collection of timetables, which also have many fields to store information about timetables like less costly timetable, most costly timetable, average cost, average number of violations of constraints and the total number of timetables in population. Timetables are ordered from least costly to most costly. A two stage verification strategy is used which ensures that each lesson of a course is scheduled exactly once. It is done in two steps, in first step checking each lesson which appear more than once altered in such a way that it appear exactly once and in second step any lessons which did not appear are booked to spare spaces randomly. The benefit of this representation is that room must not be double booked and every lesson must be scheduled at once.

A university timetable stores information about what classes are booked in each room, at any hour of the day, on any day of the week. Each of these bookings (or NULL bookings) is one gene. A timetable also has fields which describe (decode) some aspect of this genetic information. A timetable has a field which stores its cost. It also has fields which store the number of breaches of each type of hard constraint.

EVALUATION

Each timetable is evaluated when considering all constraints.

If M={1,…, $\bar{m}$ }is the set of total constraints for problem and α is the total number of violations of constraint m∈ M. If ϑ_i is the penalty of ith violation of constraint m∈ M then cost of a chromosome could be determine with this formula.
Fitness value of a timetable = $\sum_{m \in M} P_{(m)}$, where

$$P_{(m)} = \sum_{i=1}^{\alpha} \vartheta_i$$

GENETIC OPERATORS

Roulette wheel Selection

This genetic operator is used to select potentially useful solutions for recombination. We select time-tables with Roulette wheel selection from the population for breeding.

Crossover

We bred a child by performing single point cross over on the parents. The benefit of this crossover is that each parent has balance chance of sending each gene.

Mutation

Mutation is used to avoid from getting trapped on local optima. Probability that a gene will undergo mutation is 2 * Mutation rate /1000.

We have used an elitist natural selection operator for timetables eradication because timetables are in ordered link list, so its easy to use. We eradicate 50 % of timetables in each generations.

DATA GENERATION

We generate data for our algorithm randomly. Total number of courses, classes, teachers, rooms, room types, days, number of periods per day are fixed.
1. For each class, number of courses is randomly generated from 1 to 4, a course can be part of more than one class.

2. Each course is assigned randomly to a teacher in such a way that each teacher must have a minimum one course to teach.
3. Number of students who attend a certain course is randomly chosen between 20 and 100.
4. Every Course is randomly assigned a room type from a given set of room types (laboratory, with computers, projector or any other equipment).
5. For each course, we choose a random number 2 to 5 which shows the number of lectures of course.
6. For each course, it is required that it should not finish its lectures before a minimum number of days. We have taken randomly minimum number of day from 1 to 4.
7. For each teacher, we have certain number of unavailabilities, for small problem 2, for medium and large 3 slots. These slots are chosen randomly.
8. Each room capacity is randomly generated between 30 to 150.
9. Each room is unavailable randomly for five periods.
10. For each course, it is required that it should not be scheduled after a fix maximum number of days. We have taken randomly maximum number of day from 3 to 5.
11. One lecture of each course is randomly pre assigned a period.
12. The maximum daily number of teaching hours allowed for any class are 5.
13. For each course, maximum limit on number of teaching hours per day is randomly taken from 1 to 3.

EXPERIMENTAL RESULTS

To check the performance and efficiency of the algorithm, we apply the algorithm on randomly generated instances. We have generated 7 small, 6 medium and 6 large size problem instances for this purpose. The program is coded in C and run on a Intel 2.5 GHz RAM 3.5 Go pc. We run algorithm 10 times for each problem instance to see its performance in detail.

C=10, T=7, K=4, R=3, RT=1, P=6, D=5

Problem instance	Minimum cost	Maximum cost	Average cost	S.D.	Time (sec)	No. of generations	No. of events	% of used seats	% of used slots
1	0	1	.2	.42	4	685	46	92	76
2	0	1	.1	.31	2	243	45	86	75
3	0	2	.5	.70	5	923	41	88	68
4	0	1	.3	.48	2	474	46	89	76
5	0	0	0	0	2	330	41	71	68
6	0	0	0	0	1	303	38	81	63
7	0	0	0	0	1	193	43	75	71

Numerical results for small problems Table 1

Table 1 shows results for small size problem instances, table 2 shows results for medium size problem instances and table 3 shows for large size problem instances. In each table C is number of course, T is number of teachers, K is number of classes, R is number of rooms, RT is number of room types, P is number of periods and D for working days. After running each problem instance 10 times, we have given minimum cost, maximum cost, average cost, S.D. (standard deviation), time in seconds, total number of generations of minimum cost solution run, total number of lessons (events) and percentage of used slots and seats.

C=30, T=24, K=14, R=6, RT=2, P=6, D=5

Problem instance	Minimum cost	Maximum cost	Average cost	S.D.	Time (sec)	No. of generations	No. of events	% of used seats	% of used slots
1	0	0	0	0	25	581	122	83	67
2	0	5	2.09	1.6	55	1247	127	76	70
3	0	3	1.36	1.0	50	1178	127	86	70
4	1	5	2.6	1.5	84	2000	144	76	80
5	0	0	0	0	10	220	108	85	60
6	0	4	2.1	1.2	60	1399	127	77	70

Numerical results for medium problems Table 2

Our stopping criteria for algorithm is minimum cost zero or 2000 iterations, which comes first. Parameters for genetic algorithm are as follows: mutation rate = 3, crossover rate =2, population size for small and medium problem = 100, population size for large problem 250. Reason behind taking big population size for large problem is that it enhance diversity and reduce the number of generations but for small size problems results are already good.

C=50, T=40, K=25, R=10, RT=3, P=6, D=5

Problem instance	Minimum cost	Maximum cost	Average cost	S.D.	Time (sec)	No. of generations	No. of events	% of used seats	% of used slots
1	2	8	4.7	2.0	226	2000	210	83	70
2	2	5	3.6	.84	217	2000	209	74	69
3	1	4	2.2	1.3	216	2000	206	81	66
4	3	9	6.7	1.8	220	2000	220	80	73
5	1	5	3.7	1.	217	2000	210	85	70
6	3	11	6.8	2.7	220	2000	228	79	76

Numerical results for large problems Table 3

CONCLUSION

This paper describes a university course timetabling problem with many constraints and formulates this in a mathematical model. This work shows that with adequate solution representation through genetic algorithm, one can obtain good quality solutions. This problem is highly constrained, we believe that this problem is quite general and may be applied on wider range of university course timetabling problems, simply by omitting some constraints or specializing the further ones. Any soft (hard) constraint can be changed as hard (soft) constraint easily according to university requirements which varies normally from institution to institution. In our algorithm, there is no difference between hard and soft constraints, we give very high penalty value for violation of hard constraints than soft constraints, in this way hard constraints would be eliminated very soon. So it is really practical to use any constraint (soft or hard) according to choice and need. We use randomly generated data, so it is possible to have a set of constraints that make it impossible to get feasible solution. However in our experiments we have not faced this type of situation. We expect improvements could be obtained by experimenting carefully different variants of crossover, mutation and population size because these have significant impact on the performance.

REFERENCES

Abdullah,S. and Turabieh,H. 2012. "On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems". *Information Sciences*, 191 pages 146-168, 2012.

Aladag.C.H and, Hocaoglu,G. 2007. "A tabu search algorithm to solve course timetabling problem". *Hacettepe Journal of Mathematics and Statistics*, 36(1) pages 53–64, 2007.

Aladag.C.H, Hocaoglu,G and Basaran, M.A. 2009. "The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem". *Expert Systems with Applications*, 36, pages 12349–12356,2009.

Broek,J., Hurkens, C. and Woeginger, G. 2007. "Timetabling Problems at the TU Eindhoven". E.K. Burke and H. Rudov´a (Eds.): *PATAT 2006, LNCS* 3867, pages.210-227, 2007.

Carter, M.W. and G. Laporte, G. 1998. "Recent developments in practical course scheduling". E. Burke, M. Carter (Eds.), Practice and Theory of Timetabling II, *Springer-Verlag*: 3–19, 1998.

Carter, M.W. and G. Laporte, G. 1998. "Recent developments in practical course scheduling". *Proc. of the 2nd Int. Conf. on Practice and Theory of Automated Timetabling, LNCS 1408, pages 3–19, 1998.*

Colorni,A., Dorigo,M. and Maniezzo,V. 1990. "Genetic algorithms - A new approach to the timetable problem". In Akgul et al. (eds.), *NATO ASI Series, Combinatorial Optimization*, Lecture Notes in Computer Science, F(82), pages. 235-239, 1990.

Daskalaki, D., Birbas,T. and Housos, E. 2004. An integer programming formulation for a case study in university timetabling". Houck, C. R., J. A. Joines, and M. G. Kay, 1996. *European Journal of Operational Research,*153, pages 117-135, 2004.

Davis,L. 1991. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991

International Time tabling competrition, 2007. http://www.cs.qub.ac.uk/itc2007/curriculmcourse/course_curriculm_index.htm

International Time tabling competrition, 2002. http://www.idsia.ch/Files/ttcomp2002

Jat, S.N. and Yang, S. 2009. "A Guided Search Genetic Algorithm for the University Course Timetabling Problem". *Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2009),* Dublin, Ireland, 10-12 August 2009.

Jain, A., Jain.S. and Chande, P.K. 2010. "Formulation of Genetic Algorithm to Generate Good Quality Course Timetable". *International Journal of Innovation, Management and Technology*, Vol. 1, No. 3, ISSN: 2010-0248, August 2010.

Schaerf, A. 1999. "A survey of automated timetabling". *Artificial Intelligence Review*, 13 (2),pages 87-127, 1999.

Schimmelpfeng, K. and Helber, S. 2007. "Application of a real-world university-course timetabling model solved by integer programming*". OR Spectrum 29, pages 783–803, 2007.*

Wren,A.1996. "Scheduling, timetabling and rostering- A special relationship". Burke and Ross (eds), Springer-Verlag Berlin Heidelberg, pages 46- 75, 1996.

Werra,D. 1985. "An introduction to timetabling". *European journal of Operational research, 19*, pages 151-162, 1985.

Yakoob, Sherali, S. 2006. "Mathematical Joines, programming models and algorithms for a class-faculty assignment problem*". European Journal of Operational Research, 173(2),* pages 1028-1044, 2007.

Yakoob,A. and Sherali, S. 2007. "A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations". *European Journal of Operational Research, 180(3),* pages 1028-1044, 2007.

Zhipeng,L. and Hao,J.K. 2010. "Adaptive Tabu Search for course timetabling". *European journal of Operational research, 200*, pages 235-244, 2010.

Quality of Service in TDMA MANETs using Prioritised Time Slot Assignment

Benny M Nyambo[1]
Gerrit K. Janssens[2]
Wim Lamotte[3]

[1]Computer Science Department, University of Zimbabwe, Harare, Zimbabwe
Email: nyambo@science.uz.ac.zw

[2] Transportation Research Institute (IMOB), Hasselt University, Diepenbeek, Belgium
E-mail: gerrit.janssens@uhasselt.be

[3] EDM, Hasselt University, Diepenbeek, Belgium
Email: wim.lamotte@uhasselt.be

KEYWORDS

Mobile ad hoc networks (MANET), Quality of Service, Time Division Multiple Access.

ABSTRACT

Mobile ad hoc networks (MANETs) find use in a wide variety of areas, which include defence-based applications, disaster relief operations, commercial applications and sensor networks. The increase in multimedia applications creates the need for a wireless network to provide Quality of Service (QoS) guarantees to provide effective communication of real-time traffic. MANETs have limited wireless resources and a time-varying topology, which makes it difficult to provide QoS in Time Division Multiple Access (TDMA)-based networks. In TDMA each user is allocated a time slot within each time frame. Therefore a scheduling algorithms is required to schedule traffic by optimally allocating time slots in TDMA. The TDMA scheduling algorithm is an optimisation problem on a directed graph (the MANET). This research applies the Dining Philosophers' (DP) problem to the time slot assignment in TDMA-based MANETs. The DP-model of the MANET is represented as a Petri net. A variant of that assignment problem is developed in which priorities within nodes are provided such that nodes with multimedia traffic will have higher priority. The doorway concept is used to handle starvation, required to solve the Prioritised Dining Philosophers' (PDP) problem. The pseudo-code of the algorithm describes in detail how it finds the solution.

INTRODUCTION

Data communication networks have been for a very long time wireline networks but in the last decade wireless networks have been introduced and have shown to be an interesting medium for many new applications. Mostly wireless communication takes place between a user (notebook, PDA or cell-phone) and a base station, but recently the wireless medium has offered some more opportunities through a concept called a Mobile ad-hoc network (MANET). A mobile ad hoc network is a network of self-organized mobile nodes without initial existing infrastructure. Wireless devices are able to create their own wireless network (which refers to the self-organising property) in a distributional fashion. Users may move and thereby change communications links in the network. Nodes forward data packets for each other using multi-hop wireless links to achieve communication (while in the classical wireless network a single-hop communication is used) (Chlamtac et al., 2003).

Nodes make use of the same random access wireless channel, cooperating in a friendly manner to engaging themselves in multi-hop forwarding. In this way each node acts at the same time as a node and router that route data to/from other nodes in the network (Abolhasan et al., 2004). Mobile Ad Hoc Networks represent the future generation wireless networks, with a high degree of versatility and robustness, capable of being deployed quickly and economically at places lacking any infrastructure. The characteristics of these networks such as bandwidth scarcity and fluctuations, node mobility, hostile working conditions and battery power constraints, have hindered their development. Projected applications of such networks include defence-based applications (war scenarios), disaster relief operations (earthquake, rural areas), commercial applications (home networking, extending Internet connectivity, inter-vehicular networks) and sensor networks. Figure 1 shows an example of a MANET.

Extensive research has been done on routing protocols for MANETs (e.g. Liao et al., 2001). The evolution of the multimedia technology and the commercial interest of companies to reach widely civilian applications call for a complete solution for QoS in TDMA MANETs in order to provide effective communication of real-time traffic. This evolution calls for a complete solution for QoS in TDMA MANETs in order to provide effective communication of real-time traffic. With increase in multimedia applications, there is a need for wireless network to provide some QoS guarantees for highly sensitive application. An increase of the high requirement traffic load increases the network congestion due to the failure of the traffic differentiation schemes to adapt new traffic application requirements to the network available bandwidth.

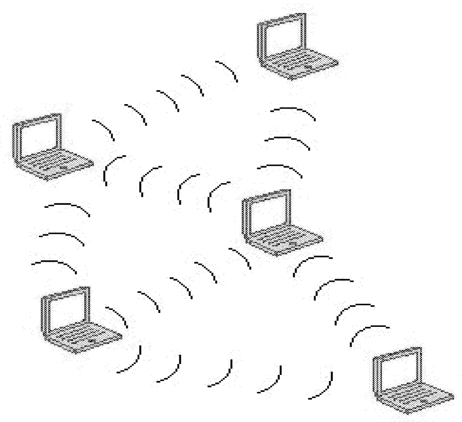


Figure 1: An example of a mobile ad hoc network

Ad-Hoc Networks have limited wireless resources; time varying topology and lack of infrastructure. These factors make it difficult to estimate and allocate resources to nodes thus making it difficult to provide QoS in TDMA MANETs. The challenges posed by TDMA MANETS can mainly be addressed by effective QoS reservation and admission control strategies. Mobile ad hoc networks are the future of wireless networks because they're practical, versatile and easy to use. These networks provide a new approach for wireless communication and by operating in a license free frequency band prove to be relatively inexpensive. Ad hoc Networks have limited wireless resources; time varying topology and lack of infrastructure. These factors make it difficult to provide QoS in TDMA MANETs.

PROBLEM FORMULATION

Time division Multiple Access (TDMA) is a channel access method that allows several users to share the same frequency by dividing time into time frames and each user is allocated a time slot within each time frame. Figure 2, illustrates the concept of splitting time into time frames and time slots in TDMA. [references]

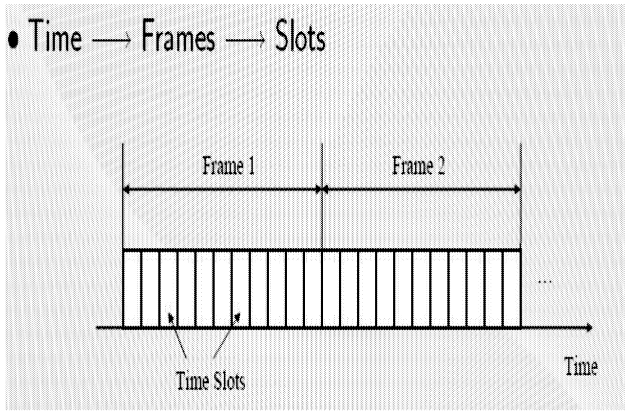


Figure 2: an illustration of the division of time into frames and subsequently into time slots.

In this model, a node can broadcast a message to its adjacent nodes. The time is divided into a non-overlapping equal time period time frame which is divided into a number of non-overlapping equal time periods, called time slots. The slots are numbered from 1 to *MaxSlot*. We assume that *MaxSlot* is sufficiently large to handle all the assignment strategies for an input graph. Informally, the objective of the scheduling is that each node picks a time slot during which it can transmit without conflict. We say that two nodes are in conflict if and only if they are in one or two hops away from each other. Typically this definition of conflict is used in a broadcast mode of TDMA scheduling where any two nodes within a two hop range can have radio interference at some node in their transmission ranges due to the hidden terminal problem and their radio broadcast transmission causes that node to receive a degraded signal. [11];

We need to find a scheduling algorithm that will allow us to schedule traffic by optimally allocating time slots in TDMA for a MANET. The number of unused slots is supposed to be minimized. The algorithm should take into account the different priorities in the traffic, giving higher priority traffic more slots. There is a need that if the highest priority traffic requests bandwidth and the channel is busy and we want the algorithm to release the slots allocated to the least priority flow.

FORMULATION AS A GRAPH TOWARDS AN OPTIMISATION PROBLEM

A MANET can be modelled by a directed graph $G=(V, E)$, which is a bipartite multigraph, where V is the node set and E is the edge set of G. The nodes denote stations in a network. E is such that for any two distinct nodes u and v, edge $(u, v) \in E$ if v can receive transmission from u. We assume that $(u, v) \in E$ if and on if $(v, u) \in E$, that is links are bidirectional.

We define the TDMA scheduling problem to be a problem of finding a time slot for each node, given an input graph, such that if any two nodes are in conflict, they do not have the same time slot. We say that two nodes *u* and *v* are in conflict if and only if *u* and *v* are in one or two hopes away from each other.

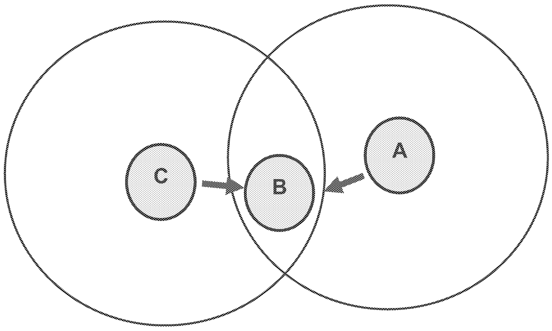


Figure 3: Illustration of nodes in conflict

Typically this definition of conflict is used in a broadcast mode of TDMA scheduling where any two nodes within a two hope range can have radio interference at some node in their transmission ranges due to hidden terminal problems and their radio broadcast transmission causes that node to receive graded signals.

This TDMA scheduling problem is often known as the *static channel assignment* problem or *reuse channel assignment* problem. After each node finds its slot, it (re)uses that slot at each time frame for collision-free data transmission. Thus, an algorithm that minimizes the number of time slots being assigned allows the system to minimize the frame size (originally set to *MaxSlot*), thus increasing channel utilization. After the channel assignment, the maximum time slot being assigned in the network must be broadcasted to the entire network. Our definition of the TDMA scheduling problem deals only with channel assignment part. We assume that synchronizing with all the nodes on the maximum slot number is not counted for the cost of the scheduling algorithm [11].

In the TDMA model, anode's use of a slot depends not only on the status of its 1 -hop neighbours' use of the slot but also on the 2-hop neighbours' current use of the slot. This is due to the hidden and exposed terminal problem which must be taken into account. This means that a slot is free if and only if no other node in the 2-hop neighbourhood is using it.

In a fully distributed broadcast scheduling algorithm, each node calculates its own schedule based on its information and information from its 1-hop and 2-hop neighbours. Schedules are computed in a parallel fashion, making such algorithms more scalable and practical than centralized scheduling.

In order to prevent interference in TDMA environment, a slot is considered free to be allocated to send data from a node x to a node y if the following conditions are satisfied:

Slot t is not scheduled for transmitting or receiving in either node x nor node y

Slot t is not scheduled for receiving in any node z that is a 1-hop or 2-hop neighbour of x

Slot t is not scheduled for receiving in any node z that is a 1-hop or 2-hop neighbour of y

THE DINING PHILOSOPHERS' PROBLEM

In [11] they used a modified dining philosophers' algorithm called DRAND to solve the assignment problem in MANETs. The Dining Philosophers (DP) problem is a classical resource allocation problem that formulates a common synchronization need of multiple processes in accessing a set of exclusive resources. The DP problem can be defined in the following way. There are n philosophers in the system and a Chop-stick set F. Each philosopher rotates its state from thinking, hungry, eating and releasing. In order to eat, each philosopher needs a fixed set of Chop-sticks (a subset of F), and it needs to acquire all of them to start eating. We say that two philosophers are contending if their Chop-stick sets contain a common Chop-stick. When finished eating, he releases his Chop-sticks for use by his contenders or by himself later when he becomes hungry again. No two contending philosophers can eat at the same time. The goal is to minimize the waiting time of hungry philosophers (or known as response time).

The DP problem captures the type of synchronization and resource allocation requirements commonly arising in distributed systems such as database transaction systems and distributed file systems where multiple processes need to update several data items or files "consistently" at the same time [11]. The problem is shown diagrammatically in figure 3. The problem has five philosophers and a Chop-stick each between adjacent philosophers. Each philosopher must use two Chop-sticks at a time every time.

In this research we apply the DP to the timeslot assignment problem in TDMA based MANETs. We reformulate the problem in the following way. There are ***n*** nodes in the system and a Frameset ***F*** which is composed of a set of time slots **s**. Each node rotates its state from ***Idle, Trying, Slot usage and Releasing***. In order to transmit, each node requires a fixed time slot ***s*** which is a subset of the Frameset ***F***. It needs to acquire the time slot in order to start transmitting. We say two nodes are contending if they require use for one time slot simultaneously. When finished transmitting or receiving, the node releases the time slot for other nodes to use or for use by itself latter when it requires channel usage again. No two contending nodes can use the time slot at the same time. The goal is to minimise waiting time for high priority channels

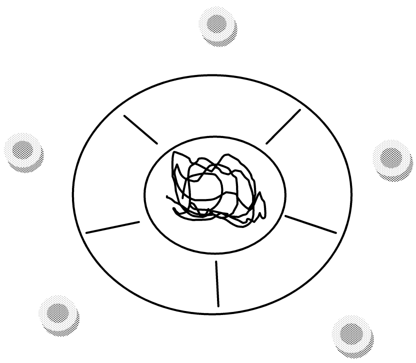


Figure 4: the dining philosophers' problem consisting of five philosophers and five Chop-sticks and a bowl of spaghetti at the centre of the table.

A frameset **F** is composed of the time slots s which is equivalent to the components of the Chop-stick set. It this comparison, Node (**n**) =Philosopher (**p**), Frameset (**F**) =Chop-stick set (**F**) Slot Usage= Eating (which is either transmitting or receiving), Thinking=Idle time, Hungry=Trying, Releasing =Releasing. We can model the DP model in MANETs using Petri Nets as shown in figure 5 below.

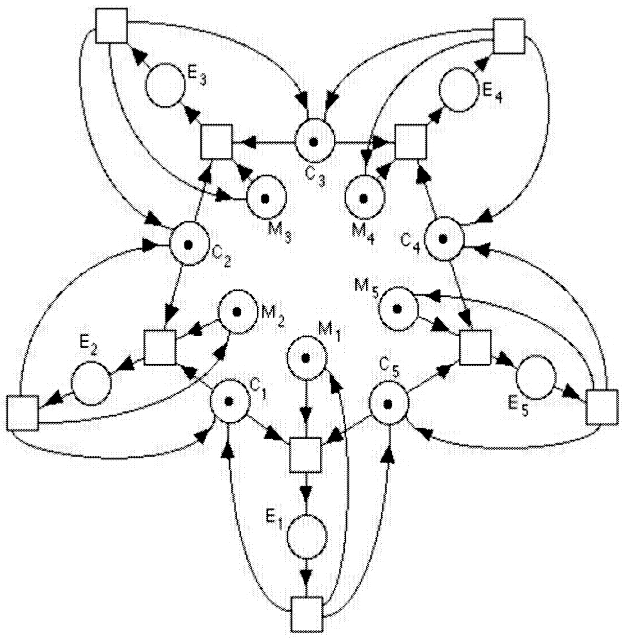


Figure 5: Petri net for the dining philosophers' problem. M is meditating, E is eating and C is the chop-stick set (taken from Michaels and Rosen)

PRIORITISED DINING PHILOSOPHERS' PROBLEM

In our solution we need to factor in priorities within nodes such those nodes with multimedia traffic will have higher priority than Best-effort traffic. We call the algorithm prioritized Dining philosophers algorithm. Initially the node is in an idle state, which is similar to the thinking state in the dining philosopher's algorithm. If it decides it now wants to receive or transmit, it sends a request and changes from the idle state to the trying state. If we draw parallels to the DP algorithm, this is similar to the Hungry state of the philosopher. If the node succeeds in gaining access to the channel it gets connected and graduates into the slot usage state.

A race condition exists between the contending nodes based on the priorities assigned on each node. The outcome depends upon which of two or more competing processes is granted a resource first. In the worst case scenario, if there are no more free slots and a higher priority nodes requests slot usage whilst a lower priority node is using it, the algorithm allows the higher priority traffic containing node to take precedence over the lower priority node. The lower priority node will temporarily go a suspense state (HALT STATE). If the higher priority nodes are done the lower priority will continue with the transmission. To avoid the case of lower priority being locked out in Halt state, the algorithm promotes the suspended process to the highest priority so that it is not suspended again once it comes back on-line. This avoids starvation of lower priority traffic.

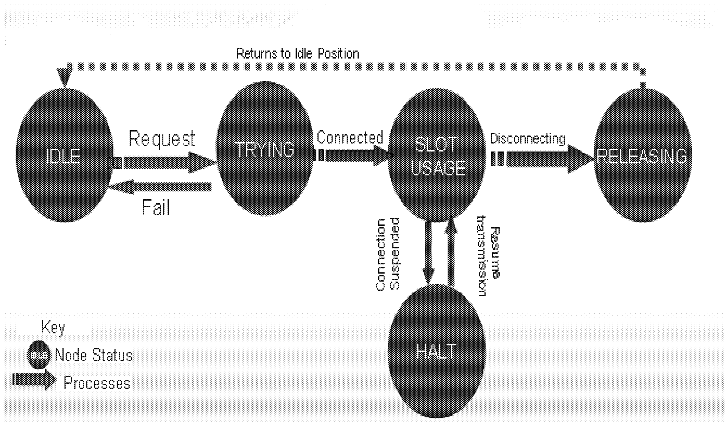


Figure 6: Variation of states in the PDP assignment algorithm

This is illustrated well by the diagram in figure 6 and the Petri Net in figure 7 illustrates the transitions from one state to the other. In the diagram node 1 has higher priority than node 2. However node 2 have access to slot Si before node 1 and there are no more free time slots. Node 1 is forced to go into Halt state and node 1 get use of slot Si and use it for a certain time. When finished transmitting, node 1 releases the timeslot to node 2 to continue transmitting.

Each node will maintain a status table for all the time slots. When a node gains the right to use a slot it broadcast to all its neighbours that the slot is in usage and all nodes change the status of the nodes from free to busy. When the node goes into releasing state, it broadcasts that the slot is now free and all other nodes will update the status of the slot in their tables. This will make it easy for nodes to identify free slots when they go into trying state.

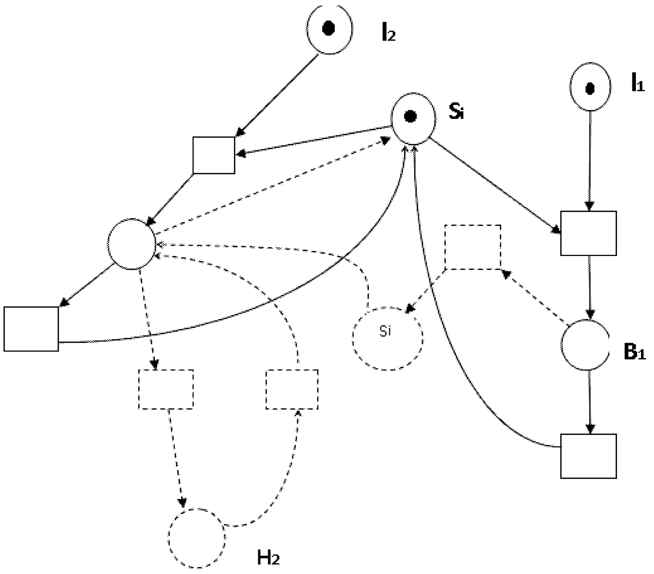


Figure 7: A Petri Net model for the PDP model for two nodes in contention for Slot Si. Node 1 has higher priority than node 2 and has forced node 2 to go into halt state H2. When Node 1 finishes it releases the timeslot back to Node1 for it to finish its transmission.

PDP PRIORITIES ASSIGNMENT METRICS

In this case we define a process as a combination of node and traffic. To solve the PDP we need to provide mechanisms to handle starvation because a process may repeatedly enter the trying state after eating, possibly pre-empting the slots of its contenders. We use the doorway concept.

A doorway code [14] is a piece of code such that if a process *p* finishes executing the doorway code, all neighbouring processes are blocked until *p* finishes eating. The doorway concept allows contending processes to set priority based on arrival to the doorway. Any contenders who cross the doorway will have a higher priority (Multiple processes may do so at the same time) over the process outside the doorway, and processes outside the doorway need to wait for those contending processes inside the doorway to finish eating. The main idea behind the doorway concept is that once any two contenders find each other outside the doorway, they do not need to check with each other again; thus only when the contenders currently inside the doorway finish, they can enter the doorway to contend for the slot. The processes leaving the doorway need to check with all of its contenders before trying for the slot again. This guarantees starvation freedom.

We use the doorway algorithm for every process *j*. A set $I_{(j)}$ keeps track of contenders that are inside the doorway, and a set $O_{(j)}$ keeps track of those outside the doorway. The algorithm is merged with the PDP algorithm of process *j* in to obtain the full scale PDP algorithm. To facilitate the merge, we add an additional state, called *pending,* to the process states. After the thinking state and before moving to the trying state, a process enters the pending state in which it executes the doorway algorithm. If a process passes the pending state, it is inside the doorway and changes to the tying state. We omit the formal proof of correctness to save the space.

Assignment Algorithm Pseudo Code

Process j Assignment Algorithm

INPUTS: Frame set **F** with slots **s**
PROCESSES: Trying to get the channel access, Halting, Disconnecting
OUTPUTS: Priority Based assignment of Slots

```
If new traffic
 State (j) = TRYING
Check table for free slots
Choosing slot ( )
{
  For slot (i)
  Broadcast request to all neighbours
  If  receive reject message
        try another slot
  else
        assign slot( i)
        change slot status to busy  in table
  If  no slot is free
        embark on priority assignment ( )
 }
 Use channel ( )
{
Broadcast  assignment to all neighbours
State = SLOT-USAGE
When done
Release slot
State = RELEASING
Change slot status to 0
Broadcast slot release message
State = IDLE
}
When (receiving a request(s) from a process j)→ if (granted ≠≠-1)
 Send a reject message to node j;
 When receive slot status message —› change slot status in table
```

Priority Assignment Algorithm

INPUTS: Interactive traffic, Streaming Traffic, Best Effort Traffic, Priorities (High, Medium, Low), States (IDLE, TRYING, SLOT-USAGE, HALT, RELEASING)
PROCESSES: Assignment of priorities to traffic and substitution mechanisms
OUTPUTS: Priority based assignment of traffic

#Define Priority High, Low, Medium

#Define Traffic Interactive, Streaming, Best Effort

Interactive=high, Streaming=medium, Best Effort=Low

For node (k) with priority =high

If (state (i)=SLOT-USAGE and (node (i) using channel has

priority=medium)

Or (the node (i) using the channel has priority=low)

Node (k) send halt message to node (i)

Move node (i) to the HALT state. Node (k) will GetSlot () and

transmits/receives

Invoke doorway algorithm ( )

If (state (j) = RELEASE);

Node(k) sends wake up message to node (i)

move node (i) from HALT state back to the SLOT-USAGE

node (i) priority promoted to high.

CONCLUSIONS

As routing in Mobile Ad-Hoc Networks (MANETs) is considered a difficult problem, research into routing protocols is well considered. A modified version of the Dining Philosophers' Problem has been considered in the communications literature to solve the assignment problem in MANETs. The algorithm is reformulated towards a TDMA-based MANET. The algorithm is further refined to factor in priorities amongst nodes so that multimedia traffic has a higher priority than best-effort traffic.

REFERENCES

Abolhasa M., Wysocki T. and Duthiewicz, E., 2004, A review of routing protocols for mobile ad hoc networks, *Ad Hoc Networks,* vol. 2, pp. 1-22.

Chlamtac I., Conti M. and Liu J. J.-N., 2003, Mobile ad hoc networking: imperatives and challenges, *Ad Hoc Networks,* vol. 1, 13-64.

Liao W.-H., Tseng Y.-C., and Sheu J.-P., 2001, GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks, *Telecommunication Systems*, vol. 18, no. 1, pp. 37–60.

Michaels J.G., Rosen K.H., Applications of Discrete Mathematics. McGraw-Hill Higher Education, 2007

Jawhar I., Wu J., QoS Support in TDMA based Mobile ad hoc networks, Journal of Computer Science and Technology, Nov 2005

Choy M., Singh A.K., Efficient fault-tolerant algorithms for distributed resource allocation. ACM Transactions on Programming Languages and Systems, 17:535-559, 1995.

BIOGRAPHIES

BENNY M. NYAMBO obtained the Bachelor of Science Honours degree in Physics in 1999 and MSc in Applied Physics in 2002, both from the University of Zimbabwe, Harare, Zimbabwe. He is currently studying towards a PhD in Computer Science at the Hasselt University, Belgium. He has been a lecturer in Computer Science and the University of Zimbabwe since 2003. He has research interests in performance modelling of communication networks and in embedded systems.

GERRIT K. JANSSENS received degrees of M.Sc. in Engineering with Economy from the University of Antwerp (RUCA), Belgium, M.Sc. in Computer Science from the University of Ghent (RUG), Belgium, and Ph.D. from the Free University of Brussels (VUB), Belgium. After some years of work at General Motors Continental, Antwerp, he joined the University of Antwerp until the year 2000. Currently he is Professor of Operations Management and Logistics at Hasselt University (UHasselt) within the Faculty of Business Administration. His main research interests include the development and application of operations research models in production and distribution logistics.

THERMAL MODELING OF AN ELECTRIC MOTOR USED ON ROAD VEHICLE POWERTRAIN

Valerian CROITORESCU[1]
Jan ANTHONIS[2]
Nicolae VASILIU[3]

University POLITEHNICA of Bucharest
[1]Automotive Engineering Department, [3]Fluid Power Laboratory
313 Splaiul Independentei st., 6th Sector,
060042, Room JC004, Bucharest, ROMANIA
[2]LMS International
68 Interleuvenlaan,
3001 Leuven, BELGIUM

E-mail: valerian.croitorescu@gmail.com, jan.anthonis@lmsintl.com, vasiliu@fluid-power.pub.ro

KEYWORDS
Hybrid vehicle, electric motor, thermal modeling, simulation, losses

ABSTRACT

2012 brings new European Union regulations concerning harmful emissions for road vehicles. The hybrid electric and electric mobility is a technology revolution for the automotive industry. In response to the stringent regulations and requirements enforced, vehicle manufacturers are developing new strategies, using hybrid electric or electric powertrain solutions. In order to obtain the goal of using alternative powertrain solutions, it is important to integrate from the early beginning different devices and to optimize their behavior. However, with a minimum of one electric motor architecture, a hybrid electric powertrain triggers more concerns surrounding energy flows. During operation, there are energy losses caused by heat generation. Generated heat, emphasized as losses, is having a negative effect on motor efficiency. Apart from the functional behavior of the electric motor, which is the production of torque/rotational speed, due to internal losses it also has a thermal behavior, which needs to be taken into account. Thereby this paper aims to present the thermal behavior of an electric motor in order to study the energy flows corresponding to the electric motor and to determine its efficiency. Based on thermal resistances, a thermal network was developed to compute internal losses for different parts of the motor. Using simulation, both the motor and its thermal model, were integrated in a mock-up virtual hybrid electric vehicle to study its behavior during a predefined cycle.

INTRODUCTION

New European standards will require auto manufacturers to reduce CO_2 tailpipe emissions down to 130 g/km by 2015. For that it is need to predict and to minimize the harmful emissions from the model developing stage, by using a complete and detailed energy balance for a given powertrain configuration.

The automotive manufacturers are developing an array of other promising technologies. Hybrid vehicles are not the only answer to reduce harmful emissions and fuel consumption, but at this moment they are a strong competitor for other solutions aiming this purpose. Hybrid vehicles are defined as having at least two different energy storage devices for vehicle propulsion, as the fuel tank – liquid chemical storage and the battery pack – electrical storage. A hybrid electric vehicle (HEV) stands out having at least one electric motor/generator, a battery pack and other components that every ordinary vehicle has.

The hybrid electric vehicle development process, within respect to energy management, contains more than one level for designing the model. The primary level may be used for developing and optimizing the control strategies, based on maps and data tables.

Another level may include only partial physical aspects, trying to find an optimal balance between maps and physical data. Using all the previous levels, it is mandatory to add a more developed level as a need for the detailed physical subsystems.

Many differences can be noticed between a conventional and a hybrid vehicle, but the most important ones are the powertrain architecture, the control and the different modes of operation.

The electric motors/generators are the most used solutions for the additional power sources/storage. There are many types of electric motors applied in hybrid powertrains. For all of them the concerns are regarding their power, functional behavior and lifetime. The electric motors/generators` efficiency may decrease while operating due to its thermal behavior. The electric motors/generators are generating heat that may reduce their lifetime and may lead to motors/generators failure.

The goal of this paper is to study the thermal effects of an electric motor/generator and to emphasize the effects of the high temperature on electric motors/generators efficiency.

To investigate the thermal effects, the basic geometric design choices act as a major constraint. Based on geometrical considerations a thermal model was developed.

The thermal model computes the heat production and temperature distribution of the electric motor/generator.
The thermal model was integrated in the design of the hybrid electric vehicle.
The electric motor/generator used as a part of the powertrain architecture was a switched reluctance motor.
The result of this defined modeling approach is that design parameters for the motor can be linked to multiple vehicle performance parameters.
The paper will focus on power losses estimation with respect to electric motor/generator functional behavior based on geometrical design considerations for its component parts.

SIMULATION OF A VIRTUAL VEHICLE MODEL

The HEV equipped with an electric motor/generator will be simulated using the AMESim software tool. The construction of the model for simulation, basically consists of linking sub-models from various software`s libraries. The sub-models are shown as icons. After linking the icons (which represent each component of the vehicle) in sketch mode and choosing the proper sub-model in sub-model mode, the parameters will be set. Each icon covers a fragment of C code, written using the specific equations for the system.

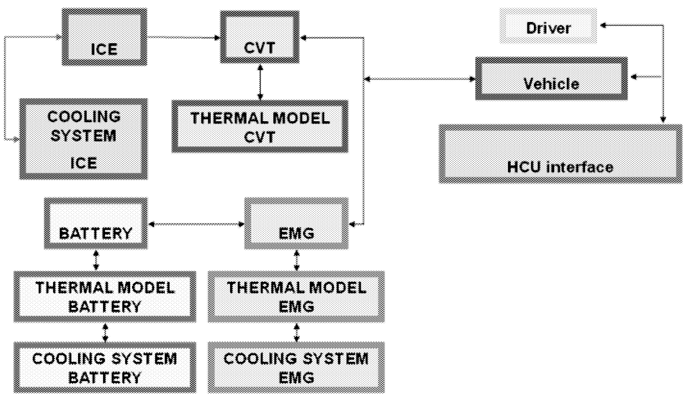


Figure 1: Model of the HEV equipped with a SRM

Using a pre-defined mission profile for setting the vehicle control speed with regard to the selected driving cycle, it is possible to input the ambient data (wind speed, density and ambient temperature) as real parameters. Acceleration and brake control take into account the anticipation of the speed control.
The driver used in this model controls/calculates braking, acceleration and gear shifting using the vehicle speed. The driver is not able to calculate the gearbox ratio or clutch control.
The Hybrid Control Unit (HCU) performs regulations for idle speed and maximum speed. Also the HCU controls the batteries (the state of charge), analyzing them in order to minimize the energy consumption. The HCU contains two different functional modules/levels, based on power demand and the hybrid mode selection. The HCU estimates the load demand based on current load due to road conditions, mass, speed, inertia. The hybrid modes are selected from predefined operating modes, based on power demand, driver's request and vehicle state.

The model for the electric motor can be used as an electric motor and a generator, being independent from the technology of the machine and converter. It is using the torque reference, the battery voltage and the rotary velocity for being able to compute the efficiency tables. It is controlled by the HCU to deliver the motor torque and the losses.
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 $V=V_0-I*R$, where V [V] is the output voltage, V_0 [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 transmission includes a continuous variable transmission to get the engine torque and to deliver it to the wheels.
The vehicle models a simple vehicle load when considering no longitudinal slip between the tire and the ground.
The hybrid powertrain architecture used in the simulation is a parallel configuration equipped with continuous variable transmission.
The electric motor/generator is connected to its thermal model, exchanging information about temperature and power losses.
The parameters used in the simulation are the benefit of existing vehicle and components. The most important are shown in Table 1.

Table 1: The most important parameters used in simulation

Sub-model	Parameter	Value / Unit
Electric motor	Type	SRM
	Maximum power	30kW
	Maximum torque	200 Nm
Battery	Voltage	~300 V
	Rated capacity	6.35 kWh
	Cells in series per battery bank	96
	Battery banks in parallel/series	1/1
Transmission	Type	CVT
	Efficiency	~ 0.85 - based on efficiency tables

GEOMETRICAL MESHING AND THERMAL MODEL OF THE ELECTRIC MOTOR

The geometrical parameters are very important in modeling the power losses and the thermal effects, also as input parameters for the AMESim blocks used to develop the thermal model.
The electric motor/generator was divided in simple parts (Matveev 2006). For each of the components a thermal mass was associated.

The thermal masses were defined based on the made off materials and the shapes volumes. For the masses common determination it is need to use simple geometrical shapes. Therefore, for each of the components simple geometrical shapes were associated in order to define the heat transfer.
Taking into account geometrical considerations (partially presented in figure 2), the electric machine was defined as further is presented. The stator yoke was considered a hollow cylinder and its teeth a partial hollow cylinder. The rotor teeth were considered a partial hollow cylinder and its core as a hollow cylinder. The windings were considered as partial hollow cylinders. The insulation was divided in two separately regions, the first region being considered as a partial hollow cylinder and the second region as a flat plate. The air-gap was considered as a hollow cylinder, while the axial shaft was considered a cylinder. Both the bearings and the frame were considered hollow cylinders. As a result of big temperature gradient for the same component, the windings were also divided in two parts, W1 and W2, W1 near the stator yoke and W2 neighbor with rotor teeth, as shown in figure 3.

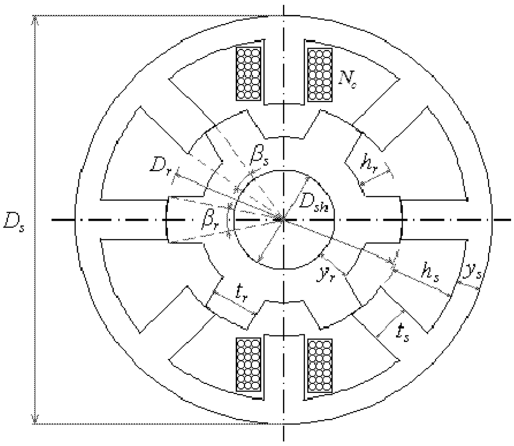


Figure 2: SRM cross sectional view

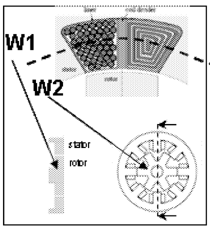


Figure 3: Windings meshing

The heat is transferred to the ambient and inside the motor. Inside the motor the heat exchange is possible between all electric machine components (Figure 4). The internal air has a very strong influence on cooling, being required to be taken into account.

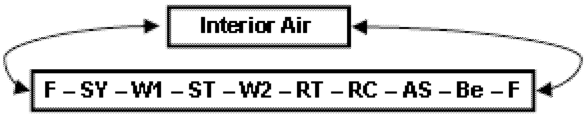


Figure 4: Heat exchange between SRM parts

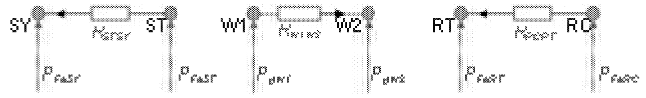


Figure 5: Heat sources and losses distribution

Starting from the known losses that can be met inside of a SRM and associating the thermal behavior of the most important components of a SRM, a thermal network was built (figure 6).
The thermal masses were computed based on geometrical considerations. Based on conduction and convection the detailed thermal network was used to model the thermal effects between the components (figure 7).

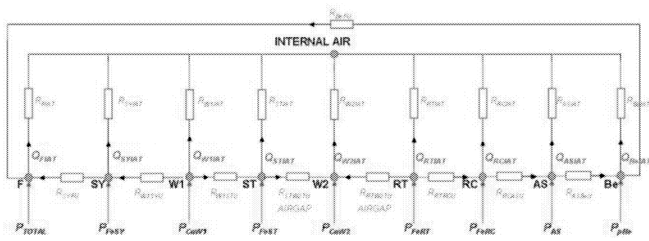


Figure 6: SRM thermal network

The frame, the stator, the rotor, the windings and the insulation were considered thermal masses, separated as presented above. Due to heat transfer, thermal resistances appear between all the thermal masses. Each resistance corresponds to conduction and convection heat transfer.
The thermal model is using all the heat flows – ambient and cooling system – in order to compute the temperature, based on power losses.

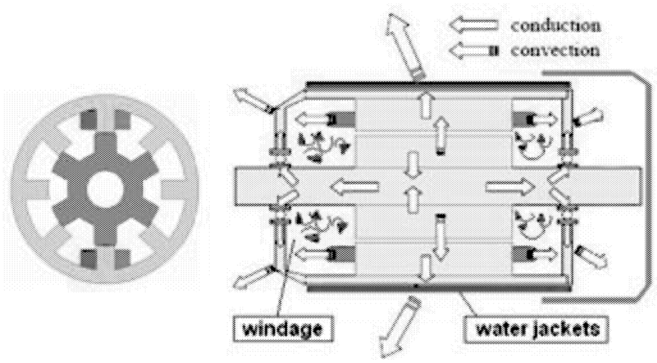


Figure 7: SRM heat flows

The thermal model developed in AMESim contains three different parts: the losses distribution, the temperature distribution in solid parts and the temperature distribution in internal air.

Losses distribution

The heat is generated by the losses. Each part of the electric motor is able to release heat to the nearest part.
Power losses in the electric motor/generator are presented as resistive or cooper losses (windings), iron losses in magnetic circuit, a part in stator (yoke and teeth) and a part in rotor (teeth and core), and mechanical or friction losses in bearings and internal air (figure 8).

Resistive losses, known also as Joule losses or cooper losses, are very important for computing the total losses of the electric motor/generator. The cooper losses can be analytically determined as the sum of products between resistance and currents for all the phases.
The resistance can be described taking into account the conductor temperature, for two different temperature values, taking into account the appropriate temperature coefficient.

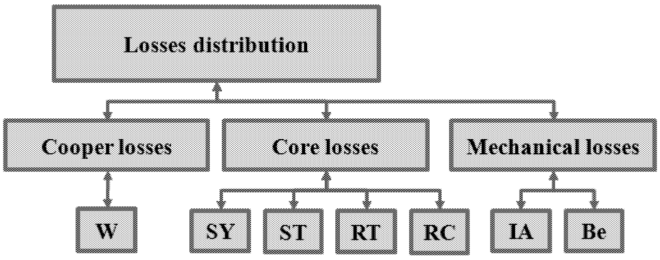


Figure 8: Losses distribution

Core losses, also known as iron losses, play an important role in design of the machine and in determination of its thermal rating. The complicated magnetic circuit of electric motor/generator makes it difficult to calculate the core losses analytically. Even though its difficulty determination, very few methods are known, one of them based on Steinmetz`s equation (and a modified version of it).
But, for a better accuracy to determine core losses, the previous principle, regarding meshing the electric motor/generator in various parts, is used. The specific masses and losses for stator pole, stator yoke, rotor pole and rotor yoke is needed in order to determine the iron losses.
Mechanical losses contain air frictions losses and bearing frictions losses.
Air frictions losses result following the air circulation around the rotor while its operation. The air friction losses are divided in two parts: the losses corresponding to the rotor seen as a rotating cylinder and the losses corresponding to the end surfaces of the rotor. Both are taking into account the rotor geometrical dimensions, its rotational speed and the made off materials.
Bearing friction losses are the results of the relative motion in bearings. It is important to determine the shaft loads in order to find the appropriate bearings load. The bearings dimensions are used additional to the shaft rotational speed.

Temperature distribution in solid parts

Being an energy transfer mode due to a temperature gradient, the conduction is taking place between all solid parts of the electric motor/generator. Convection is a major mode of heat transfer and major mode in mass transfer in fluids.

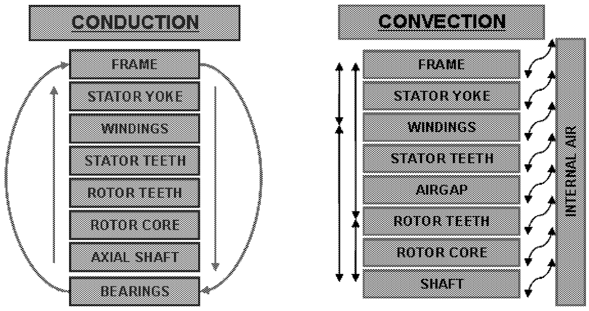


Figure 9: Heat exchange between solid parts and internal air

Each solid part has its own thermal properties due to the material they are made of. Besides material density, the specific heat and the thermal conductivity have a temperature dependency that cannot be neglected. Taking into account the various shapes assigned to electric motor/generator parts and the how heat exchange is done, it is necessary to model the conduction and convection for all of them (figure 10).

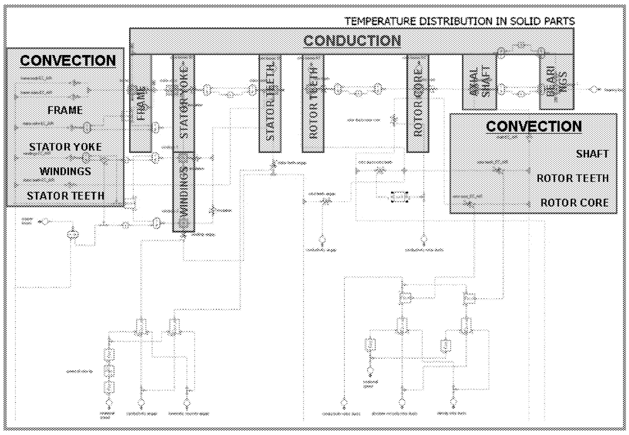


Figure 10: Temperature distribution in solid parts

As it was specified above, the frame, stator yoke, rotor core can be approximated by hollow cylindrical shapes. The same counts for the stator and the rotor teeth but instead of the diameter of 2π, only the total length of the arc is required. This assumption is possible only if the temperature gradient in the circumferential direction may be considered negligible compared to the temperature gradient in the radial or axial direction. For the windings, because they were separated in two singular parts due to the temperature gradient between stator neighboring part and rotor neighboring part, two hollow cylinders shapes were used.
Conductive heat transfer generates different ways to determine the resistance, due to the shape of the investigated part (e.g. the hollow cylinder figure 11).

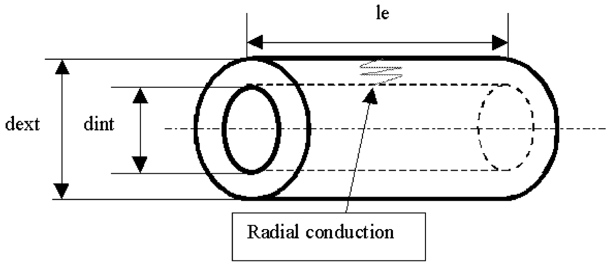


Figure 11: Radial heat transfer for a hollow cylinder

Temperature distribution in internal air

The internal air temperature distribution (figure 12) is possible between the end-cupping air and the rotor ducts. The end-cupping air consists in the two lateral sides between the rotor and the frame. The rotor ducts consist in the space between the stator and rotor teeth. The temperature in the end-cupping air varies, having the windings temperature strong influence. The gas properties are changing with temperature modification. In addition, the turbulent air flow is making it more unstable. Both gas properties and turbulent air flow are considered to be very important, being able to introduce a high difficulty level in thermal modeling,

Therefore, the density of the internal air, the constant pressure specific heat, the perfect gas constant, the absolute viscosity and the thermal conductivity are very important to develop the thermal model corresponding to the internal air. After deep investigations, the natural and forced convection were modeled for this step, knowing the influence of Reynolds and Nusselt numbers.

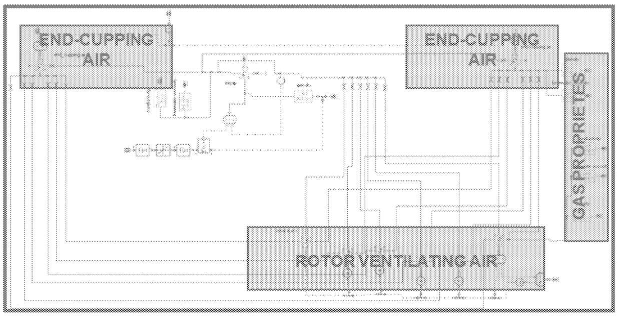


Figure 12: Temperature distributions in internal air

Similar assumptions as for the conductive heat transfer regarding the determination method for the resistances were taken into account for the convection heat transfer.

Additionally, there were taken the following assumptions: the heat flow between stator teeth, windings and the air gap was considered turbulent, the air flow velocity was the same with rotor rotary velocity, the end cupping air were modeled as vertical plate shapes, for the rotor the air gap cylinders shapes were modeled, for the stator-windings air gap vertical plate shapes were modeled, the axial heat transfer in air-gap was ignored, only the radial heat transfer being modeled.

SIMULATION RESULTS

The procedure used in this paper to estimate power losses with respect to electric machine thermal behavior have been validated. In addition, the total efficiency can be estimated based on this interpretation.

The simulation was done during a defined cycle. Starting from an ambient temperature of 20 degC and taking the assumption that all electric motor/generator component parts have the same temperature, the following results were determined. The used electric motor is also a generator and its behavior during simulation cycle is to charge the battery package during operation (figure 13). The maximum charging torque values correspond to medium rotary velocity. The torque demand for propulsion and charging respects the HCU strategies. The electric machine dynamic behavior influences the temperature evolution. The end of the cycle brings the highest temperature. While braking with the electric machine, the temperature is having a small increase, but as soon as the electric machine will continue to provide motor torque the temperature is increasing.

The power losses highest level was reached near 22 degC while the electric machine was operating near 2000 rev/min. Around 25+ degC it can be noticed that, while the electric motor is operating as generator, the losses are increasing (figure 14). Immediately after the machine is providing motor torque, the temperature decrease, and the power losses maintains its upward trend with temperature. The power losses are almost the tenth part of the maximum value for the last quarter of the cycle, corresponding to the extra-urban driving. The end of the simulated cycle records the maximum temperature value nearly 33 degC (figure 15). For higher rotational speed the temperature increases due to the high windings currents.

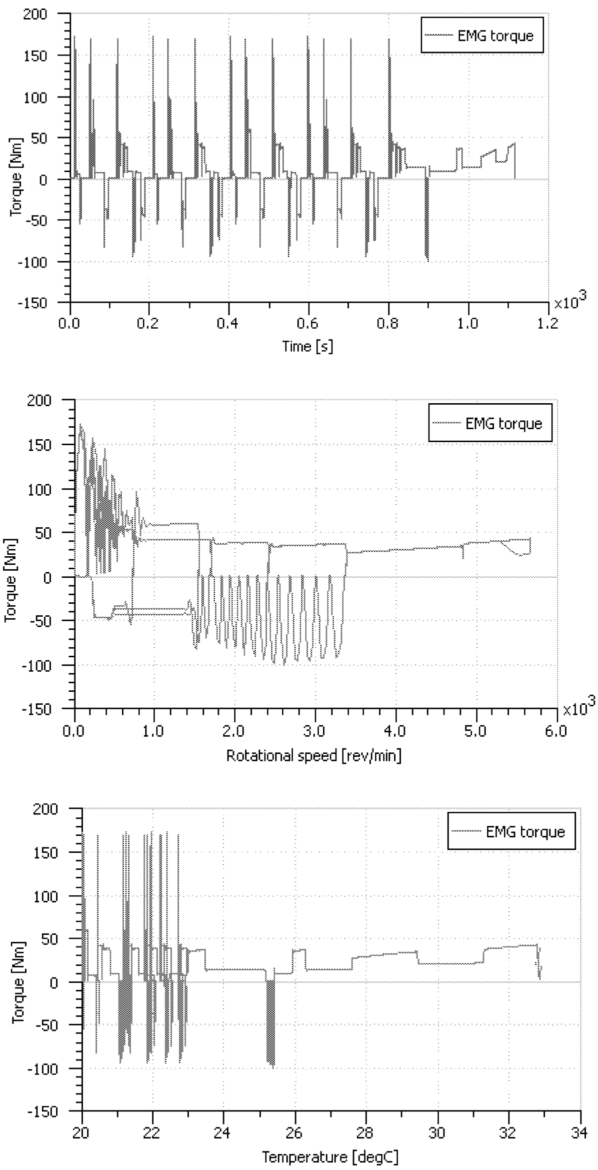


Figure 13: Electric machine torque

The losses distribution evolution (figure 16) shows that the windings losses are higher than core losses and mechanical losses. The losses have the same upwards trend.
The cooper losses are the highest due to the windings temperature increase during operation (figure 17). The warmest windings part, considering the previous mesh, is W1, near the stator yoke. Their thermal state/condition influences the neighboring bodies/parts to heat up, influencing the global motor temperature. As long as the internal air is having a turbulent flow, the heat transfer is possible. The core losses include the stator yoke, stator teeth, rotor teeth and rotor core losses (with respect to their own temperature dependency). The stator teeth are those who heat most due to windings proximity, while the stator yoke is heating less, being advantaged by its position near to the frame (figure 18). The rotor teeth and the rotor core are advantaged by the rotating air inside the motor, which is able to dissipate heat and to allow the temperature decrease.

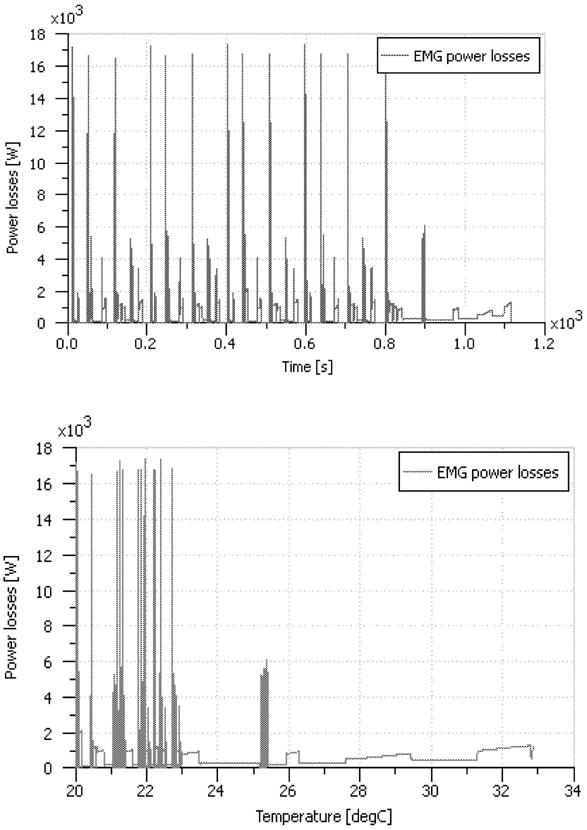
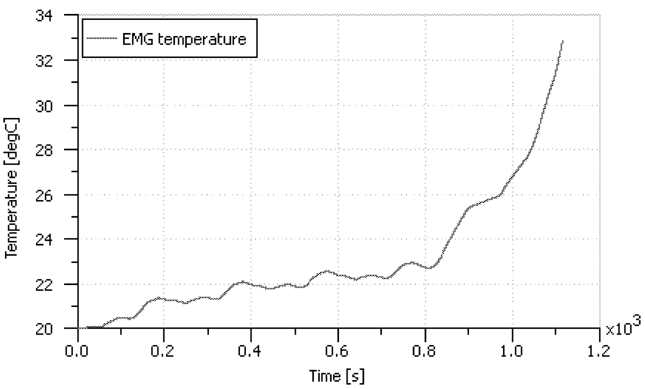


Figure 14: Electric machine power losses



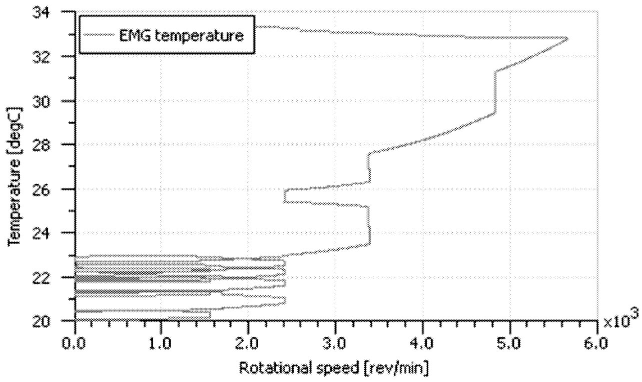


Figure 15: Global electric machine temperature

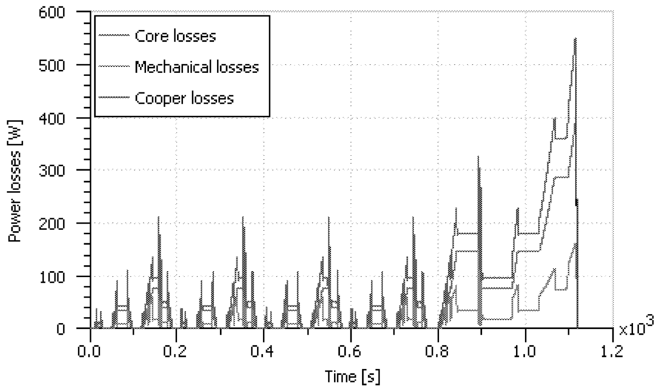


Figure 16: Losses distribution evolution

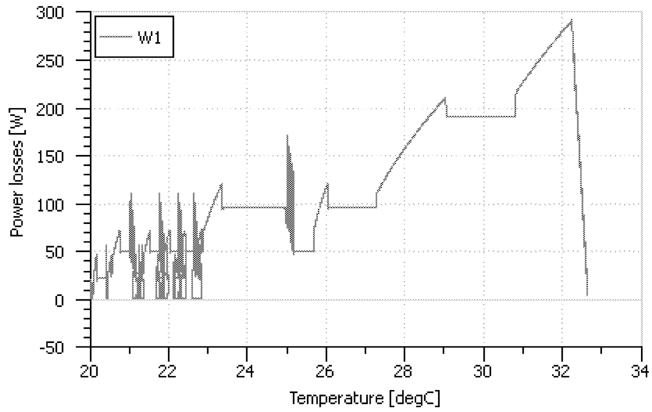
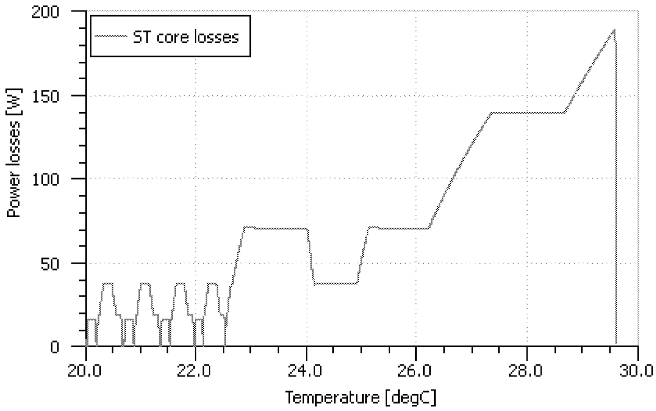


Figure 17: Cooper losses



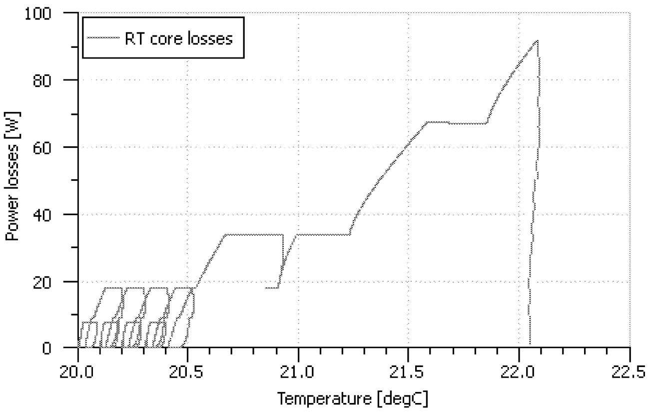


Figure 18: Core losses

The mechanical losses from air friction weight less than the bearing friction losses with regard to total losses. The bearing losses depend on shafts load and friction coefficient. Therefore, the bearing heat generation is more than twenty times bigger than air friction losses.

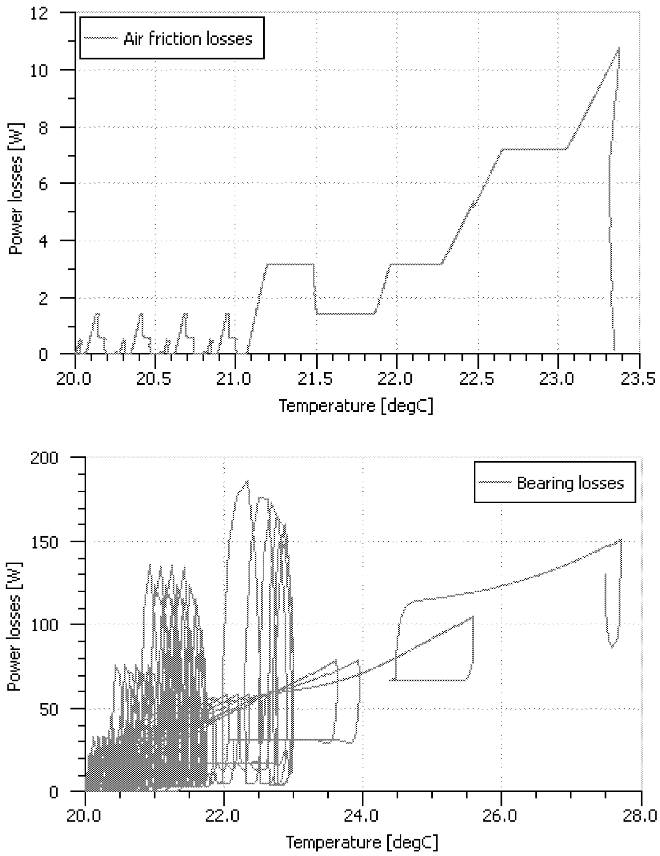


Figure 19: Mechanical losses

Temperature distributions in solid parts (figure 20) comprise the heat transfer from the warmest component (the windings) to the neighboring components. The less heated components are the axial shaft, the rotor core and the frame.

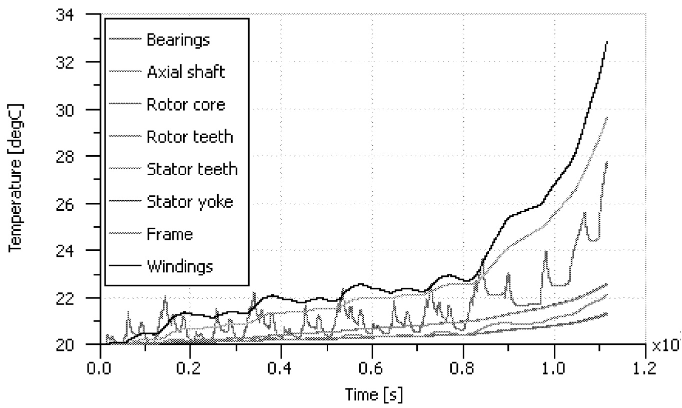


Figure 20: Temperature evolutions for solid component parts

Temperature distribution in internal air has an important influence in cooling the components by allowing the heat transfer between them.
The end-cupping air temperature is higher than the air rotor ducts temperature; the end-cupping air plates sharing the temperature with all the windings ends (figure 21).

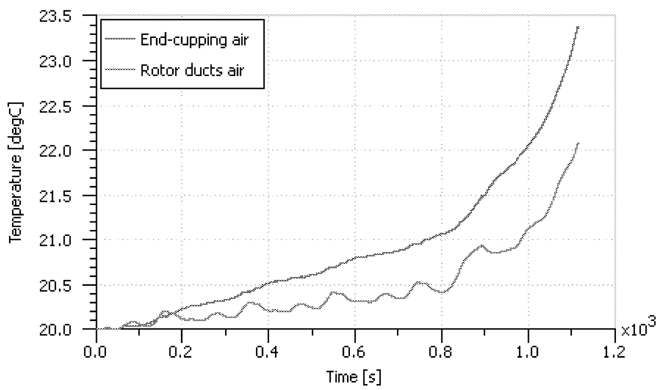


Figure 21: Temperature evolutions into internal air

The corresponding functional model allows determining the efficiency of the electric motor/generator. Changing the ambient and functional temperature it is possible to investigate the influence of power losses regarding the thermal behavior.
The simulation results may be validated using HIL simulation test bench.

CONCLUSIONS

A detailed thermal model was created by means of integrated approach that uses geometry design for the electric motor/generator. The thermal model analysis of electric motor/generator was based on the thermal equivalent circuit, built in respect with heat transfer fundamentals, by conduction and by convection.
An analysis of temperature rise of electric motor/generator due to the power losses using AMESim was introduced.
The geometrical model of the electric motor/generator has been developed being able to evaluate the geometrical parameters changes influence. The thermal properties of the electric motor/generator have been evaluated in order to

determine how the material proprieties affect the energy distribution. The metal solid parts of the motor were assumed as thermal lumped masses.

In order to estimate the power losses, the electric motor/generator was investigated in respect with its geometry, material made of and functional behavior. For each part of the electric motor/generator simple shapes were assigned. Using the lumped thermal model it was highlighted the important role that the internal air is having in entirely electric motor/generator cooling.

The thermal models for battery package and for the transmission are also very important to predict the global efficiency for entirely hybrid electric powertrain.

Similar works on this method include the study of the cooling systems design closely related with the geometric dimensions for all the component parts and design of the electric motor/generator, battery package, transmission and engine.

In addition, based on this interpretation, the total efficiency can be estimated, allowing to determine and to avoid high energy losses and possible electric motor/generator failure due to over-heating, transmission and battery package.

Finally, integrating thermal models for major hybrid powertrain components, the performance will be investigated trying to achieve the motor control law.

AKNOWLEDGEMENT

The present paper is mainly a part of first author`s PhD Thesis, being accomplished during his training period with full support of LMS International, Engineering Services.

REFERENCES

Anthonis, J., Santos, F., Croitorescu, V., Van der Auweraer, H., "Multiscale thermal and NVH models for EV/HEV Integration of an SR-based Drivetrain", VPC, Pune, India 2011

Croitorescu, V., "Modern drives using unconventional energy storage devices – Hybrid Electric Vehicles", PhD Thesis, Bucharest, Romania, 2012

Croitorescu, V., Maciac, A., Oprean, M., Marin, A., Andreescu, C., "Hybrid Powertrain Simulation", Proceedings ISC2008, Lyon, France

Croitorescu, V., Maciac, A., Negrus, E., Oprean, M., Andreescu, C., "Simulation of an all wheel drive hybrid electric vehicle", Proceedings ESFA 2009, Bucharest, Romania

Faiz, J., Iranpour, R., "Thermal model for a switched reluctance motor of TEFC design during steady state and transient operation", Electric Machines and Power System, 1998, 26(1): 77-91

Krishnan, R., "Switched reluctance motor drives: modeling, simulation, analysis, design and applications", Industrial Electronics Series, CRC Press, 2001

Krishnan, R., Arumugam, R., Lindsay, J.F., "Design procedure for switched reluctance motors", IEEE Transactions on Industry Applications, 1988, 24(3): 456-461

Matveev, A., "Development of methods, algorithms and software for optimal design of switched reluctance drives", PhD Thesis, Eindhoven, The Netherlands, 2006

Miller, J.M., "Propulsion systems for hybrid vehicles", Institution of Engineering and Technology, Power and energy series 45, London, United Kingdom

Miller, T.J.A., "Switched reluctance motor and their control", Clarendon Press, Oxford, United Kingdom, 1993

Pyrhonen, J., Jokinen, T., Hrabovcova, V., "Design of rotating electrical machines", John Wiley & Sons, Ltd., 2008

Rouhani, H., Faiz, J., Lucas, C., "Lumped thermal model for switched reluctance motor applied to mechanical design optimization", Mathematical and Computer Modeling 45, 2007, 625-638

Shoujun, S., Weiguo, L., Peitsch, D., Schaefer, U., "Detailed design of a high speed switched reluctance starter/generator for more/all electric aircraft", Chinese Journal of Aeronautics, 23(2):216-226, 2010

Van der Auweraer, H., Santos, F., Wickaert, K., Mas, P., Anthonis, J., "Vehicle integration of advanced hybrid and electric powertrain concepts", EEVC, Brussels, Belgium, 2011

LMS Imagine Lab AMESim – Tutorials

http://www.bergab.ru

http://www.infolytica.com

http://www.oica.net

http://www.privod.net.ua

http://www.simulation-research.com

http://www.wenku.baidu.com

BIOGRAPHIES

VALERIAN CROITORESCU earned his Automotive Engineer Degree in 2007, as valedictorian. He prepared the diploma project in France, inside Ecole Nationale D`Ingenieurs De Tarbes. He also attended two master of sciences programs, at University POLITEHNICA of Bucharest: `Efficiency and Security in Automotive Engineering` and `Environmental Management`. His academic records include numerous awards and certifications. In September 2007 Valerian joined the academic stuff of Automotive Engineering Department. In 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. In 2012, he earned a Ph.D. Degree at University POLITEHNICA of Bucharest, concerning Hybrid Electric Vehicles Development. A part of his Ph.D. thesis has been accomplished during a research stages at LMS International.

JAN ANTHONIS received the M.Sc. degree in mechanical engineering from the KU Leuven in 1994, and the PhD degree in 2000. From 2001 till 2007 he was post doctoral researcher at the Fund of Scientific Research in Belgium, developing researches on control and mechatronics with application in agro-machinery and agro-industry. From 2003 he is part time lecturer at the KU Leuven. In 2007 he became employee of LMS International, a spin-off company of the KU Leuven founded in 1980, specialized in noise, vibrations and mechatronics in automotive, aerospace and other mechanical industries. Here he manages a research team on control and mechatronics.

NICOLAE VASILIU is a hydropower engineer who earned the Ph.D. degree of University POLITEHNICA of Bucharest after a research stage in Ghent State University and Von KARMAN Fluid Mechanics Institute from Bruxelles. He is Professor of Fluid Control Systems, and head of Fluid Power Laboratory at University

POLITEHNICA of Bucharest. He is scientific advisor for many industrial companies. Between 2006 and 2010 he managed the ROMANIAN INNOVATION FINANCING AGENCY. He is working mainly in modeling, simulation and dynamic identification of the hydraulic and electro-hydraulic control systems with LMS International from Belgium and National Instruments from U.S.A.

AUTHOR LISTING

AUTHOR LISTING

AUTHOR LISTING