ADVANCES IN INTELLIGENT SYSTEMS AND COMPUTING 197

Nuno Pina
Janusz Kacprzyk
Joaquim Filipe (Eds.)

Simulation and Modeling Methodologies, Technologies and Applications

Springer

Advances in Intelligent Systems and Computing 197

For further volumes:
http://www.springer.com/series/11156

Nuno Pina, Janusz Kacprzyk,
and Joaquim Filipe (Eds.)

Simulation and Modeling Methodologies, Technologies and Applications

International Conference, SIMULTECH 2011
Noordwijkerhout, The Netherlands,
July 29–31, 2011 Revised Selected Papers

Springer

Editors
Nuno Pina
Systems and Informatics Department
Superior School of Technology of Setúbal
Setúbal
Portugal

Joaquim Filipe
Polytechnic Institute of Setubal / INSTICC
Department of Systems and Informatics
(Office F260)
Escola Superior de Tecnologia de Setúbal
Setúbal
Portugal

Janusz Kacprzyk
Polish Academy of Sciences
Systems Research Institute
Warsaw
Poland

ISSN 2194-5357
ISBN 978-3-642-34335-3
DOI 10.1007/978-3-642-34336-0
Springer Heidelberg New York Dordrecht London

e-ISSN 2194-5365
e-ISBN 978-3-642-34336-0

Library of Congress Control Number: 2012949998

Preface

The present book includes extended and revised versions of a set of selected papers from the 1st International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2011) which was sponsored by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC) and held in Noordwijkerhout, The Netherlands. SIMULTECH 2011 was technically co-sponsored by the Society for Modeling & Simulation International (SCS), GDR I3, Lionphant Simulation and Simulation Team and held in cooperation with ACM Special Interest Group on Simulation and Modeling (ACM SIGSIM) and the AIS Special Interest Group of Modeling and Simulation (AIS SIGMAS).

This conference brings together researchers, engineers and practitioners interested in methodologies and applications related to the education field. It has two main topic areas, covering different aspects of Simulation and Modeling, including "Methodologies and Technologies" and "Applications and Tools". We believe the proceedings here published, demonstrate new and innovative solutions, and highlight technical problems in each field that are challenging and worthwhile.

SIMULTECH 2011 received 141 paper submissions from 45 countries on all continents. A double blind paper review was performed by the Program Committee members, all of them internationally recognized in one of the main conference topic areas. After reviewing, 25 papers were selected to be published and presented as full papers and 36 additional papers, describing work-in-progress, as short papers. Furthermore, 14 papers were presented as posters. The full-paper acceptance ratio was 18%, and the total oral paper acceptance ratio was 44%.

The papers included in this book were selected from those with the best reviews taking also into account the quality of their presentation at the conference, assessed by the session chairs. Therefore, we hope that you find the papers included in this book interesting, and we trust they may represent a helpful reference for all those who need to address any of the research areas above mentioned.

We wish to thank all those who supported and helped to organize the conference. On behalf of the conference Organizing Committee, we would like to thank the authors, whose work mostly contributed to a very successful conference and the members of the Program Committee, whose expertise and diligence were instrumental to ensure the

quality of final contributions. We also wish to thank all the members of the Organizing Committee whose work and commitment was invaluable. Last but not least, we would like to thank INSTICC for sponsoring and organizing the conference.

March 2012

Nuno Pina
Janusz Kacprzyk
Joaquim Filipe

Organization

Conference Chair

Joaquim Filipe	Polytechnic Institute of Setúbal / INSTICC, Portugal

Program Co-chairs

Janusz Kacprzyk	Systems Research Institute - Polish Academy of Sciences, Poland
Nuno Pina	EST-Setúbal / IPS, Portugal

Organizing Committee

Patrícia Alves	INSTICC, Portugal
Sérgio Brissos	INSTICC, Portugal
Helder Coelhas	INSTICC, Portugal
Vera Coelho	INSTICC, Portugal
Andreia Costa	INSTICC, Portugal
Patrícia Duarte	INSTICC, Portugal
Bruno Encarnação	INSTICC, Portugal
Liliana Medina	INSTICC, Portugal
Carla Mota	INSTICC, Portugal
Raquel Pedrosa	INSTICC, Portugal
Vitor Pedrosa	INSTICC, Portugal
Daniel Pereira	INSTICC, Portugal
Cláudia Pinto	INSTICC, Portugal
João Teixeira	INSTICC, Portugal
José Varela	INSTICC, Portugal
Pedro Varela	INSTICC, Portugal

Program Committee

Erika Ábrahám, Germany
Marco Aldinucci, Italy
Mikulas Alexik, Slovak Republic
Manuel Alfonseca, Spain
Jan Awrejcewicz, Poland
Gianfranco Balbo, Italy
Simonetta Balsamo, Italy
Isaac Barjis, USA
Joseph Barjis, The Netherlands
Fernando Barros, Portugal
David W. Bauer, USA
Yolanda Becerra, Spain
Steffen Becker, Germany
Marenglen Biba, Albania
Keith Bisset, USA
Leon Bobrowski, Poland
Artur Boronat, UK
Wolfgang Borutzky, Germany
Ipek Bozkurt, USA
Jeremy Bradley, UK
Christiano Braga, Brazil
Felix Breitenecker, Austria
Christian Callegari, Italy
David Carrera, Spain
Jesus Carretero, Spain
Emiliano Casalicchio, Italy
Erdal Cayirci, Norway
Mecit Cetin, USA
Srinivas Chakravarthy, USA
Naoufel Cheikhrouhou, Switzerland
Chun-Hung Chen, USA
E. Jack Chen, USA
Priami Corrado, Italy
Christine Currie, UK
Gabriella Dellino, Italy
Francisco J. Durán, Spain
Hala ElAarag, USA
Adel Elmaghraby, USA
Roland Ewald, Germany
Jesus Favela, Mexico
Charlotte Gerritsen, The Netherlands
John (Yannis) Goulermas, UK
Alexandra Grancharova, Bulgaria
Zhi Han, USA
Scott Y. Harmon, USA
Pavel Herout, Czech Republic
David Hill, France
Brian Hollocks, UK
Xiaolin Hu, USA
Eric S. Imsand, USA
Mura Ivan, Italy
Mats Jägstam, Sweden
Tania Jiménez, France
Björn Johansson, Sweden
Rihard Karba, Slovenia
W. David Kelton, USA
Franziska Klügl, Sweden
William Knottenbelt, UK
Juš Kocijan, Slovenia
Petia Koprinkova, Bulgaria
Samuel Kounev, Germany
Raymond Kristiansen, Norway
Witold Kwasnicki, Poland
Stephanie Jane Lackey, USA
Gianluca De Leo, USA
Margaret Loper, USA
Johannes Lüthi, Austria
Radek Matušu, Czech Republic
Carlo Meloni, Italy
Adel Mhamdi, Germany
Qi Mi, USA
Jairo R. Montoya-Torres, Colombia
Il-Chul Moon, Korea, Republic of
Maria do Rosário Moreira, Portugal
Navonil Mustafee, UK
Àngela Nebot, Spain
Eckehard Neugebauer, Germany
Libero Nigro, Italy
Volker Nissen, Germany
Michael J. North, USA
James J. Nutaro, USA
Peter Csaba Ölveczky, Norway
Stephan Onggo, UK
C. Michael Overstreet, USA
Manolis Papadrakakis, Greece
James Parker, Canada

George Pavlidis, Greece
Petr Peringer, Czech Republic
L. Felipe Perrone, USA
Malgorzata Peszynska, USA
H. Pierreval, France
Katalin Popovici, USA
Ghaith Rabadi, USA
Manuel Resinas, Spain
M.R. Riazi, Kuwait
José Risco-Martín, Spain
Theresa Roeder, USA
Paolo Romano, Portugal
Oliver Rose, Germany
Rosaldo Rossetti, Portugal
Willem Hermanus le Roux, South Africa
Werner Sandmann, Germany
Jean-François Santucci, France
Hessam Sarjoughian, USA
Herb Schwetman, USA
Jaroslav Sklenar, Malta
Young-Jun Son, USA
James C. Spall, USA
Florin Stanciulescu, Romania
Giovanni Stea, Italy
Steffen Straßburger, Germany
Nary Subramanian, USA
Antuela A. Tako, UK
Elena Tànfani, Italy
Pietro Terna, Italy
Gilles Teyssiere, Denmark
Klaus G. Troitzsch, Germany
Bruno Tuffin, France
Alfonso Urquia, Spain
Giuseppe Vizzari, Italy
Gert-Jan de Vreede, USA
Hannes Werthner, Austria
Olaf Wolkenhauer, Germany
Katinka Wolter, Germany
Wai Peng Wong, Malaysia
Shaoen Wu, USA
Muzhou Xiong, China
Nong Ye, USA
Levent Yilmaz, USA
Gregory Zacharewicz, France
František Zboril, Czech Republic
Yu Zhang, USA
Laurent Zimmer, France
Armin Zimmermann, Germany
Leon Zlajpah, Slovenia

Auxiliary Reviewers

Florian Corzilius, Germany
Mahmoud Khasawneh, USA

Invited Speakers

Oleg Gusikhin	Ford Research & Adv. Engineering, USA
Simon Taylor	Brunel University, UK
Agostino Bruzzone	University of Genoa, Italy

Contents

Invited Paper

Full Papers

Invited Paper

Integration of Traffic Simulation and Propulsion Modeling to Estimate Energy Consumption for Battery Electric Vehicles

Perry MacNeille, Oleg Gusikhin, Mark Jennings, Ciro Soto, and Sujith Rapolu

Research and Advanced Engineering, Ford Motor Company, 2101 Village Road, Dearborn, MI 48121, U.S.A.
{pmacneil,ogusikhi,mjennin5,csoto}@ford.com,
sujithreddy.iitr@gmail.com

Abstract. The introduction of battery electric vehicles (BEV) creates many new challenges. Among them is driving a vehicle with limited driving range, long charging time and sparse deployment of charging stations. This combination may cause range anxiety for prospective owners as well as serious practical problems with using the products. Tools are needed to help BEV owners plan routes that avoid both range anxiety and practical problems involved with being stranded by a discharged battery. Most of these tools are enabled by algorithms that provide accurate energy consumption estimates under real-world driving conditions. The tools, and therefore the algorithms must be available at vehicle launch even though there is insufficient time and vehicles to collect good statistics. This paper describes an approach to derive such models based on the integration of traffic simulation and vehicle propulsion modeling.

1 Introduction

Increasing motorization of the developing world has led to political and economic problems such as increased cost of automotive fuels and balance of trade difficulties between nations. Presently automotive fuels are almost exclusively petroleum based, and in recent years petroleum production has not kept up with increased demand. Battery electric vehicles (BEV) promise to enable diversification of the transportation energy feedstock thereby reducing the dependence on petroleum for transport. In addition to reducing gasoline dependence it can also help to reduce greenhouse gas and other emissions, reduce global warming and provide more sustainable individual transportation.

The governments of the US, European Union, China, Japan, Korea and others have aggressively promoted vehicle electrification objectives and the major automobile companies of the world are being challenged for the first time by consumers and governments to produce battery electric vehicles. Several companies have accepted the challenge, even though there is relatively little technical acumen on deployment of these vehicles.

Deployments of BEVs present a host of new challenges including those resulting from a current lack of supporting infrastructure. Charging stations are relatively rare,

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 3–19.
DOI: 10.1007/978-3-642-34336-0_1

charging takes considerable time and currently range is more limited than with conventional vehicles. It results in range anxiety and hampers customer acceptance of the product.

To alleviate range anxiety, new vehicle electronics features are needed to help vehicle operators make driving choices that avoid discharged battery situations, extend vehicle range, and combine charging with other good uses of time. Development of these features requires practical meta-models that can accurately predict energy consumption on the public roads.

Building meta-models from field-test vehicle data requires statistical regression of public-road vehicle data (PRVD) over very large geographic areas. At present; there are not enough production test vehicles available to collect a sufficient amount of data, noise factors are not well controlled, and data collection is too time consuming to support product launch. As a result modeling and simulation are essential tools in analysis of BEV performance.

In this work we propose implementation of traffic simulation combined with propulsion modeling for determining electric vehicle energy consumption. We use traffic micro-simulation to create surrogate PRVD data that has many of the properties of actual PRVD data, specifically capturing the stochastic nature of vehicles moving through roads with traffic. The surrogate data is analyzed using propulsion simulation to estimate the amount of energy the vehicles will consume in a specific driving maneuver to derive statistical information.

2 Simulating Energy Consumption

The simulation approach used here begins with geographical information about roads and basic parameters about the vehicle of interest. The road data is coded into a microscopic traffic simulation program in which model vehicles are input into the simulation and surrogate drive data is produced. This is input into a propulsion modeling system that outputs the energy consumption of individual vehicles in the model. This data is collected and analyzed through statistical regression. The resulting energy consumption data is used to calibrate an energy consumption meta-model that can be used to estimate the energy consumption of a vehicle using surrogate data from the traffic simulation.

Scenarios of different road networks under different external conditions are simulated. For example, a road network could consist of a highway interchange or a stretch of a specific kind of road. External conditions would be such things as topography, weather, and traffic load. The traffic simulation contains BEV vehicle and other vehicles that create the traffic conditions for the sample vehicles. Regression analysis is used to create an energy consumption meta-model that predicts average energy consumption as well as the stochastic distribution. By comparing scenarios it is possible to determine main effects to build a meta-model of energy consumption from geographical data and data available in traveller information systems.

Advantages of the method include developing energy consumption models during the design process before physical testing is possible. These models can be used to improve the design and support the deployment of BEV vehicles into the consumer

market. In addition, statistical information is produced that can be used to bound the results and produce better low-energy routing algorithms.

In the simulation planning phase representative scenarios are defined that can be used to develop extensible meta-models (see Fig. 1). The scenarios must be defined using the kinds of data that are typically found in vehicle route guidance systems so they will be useful in the car.

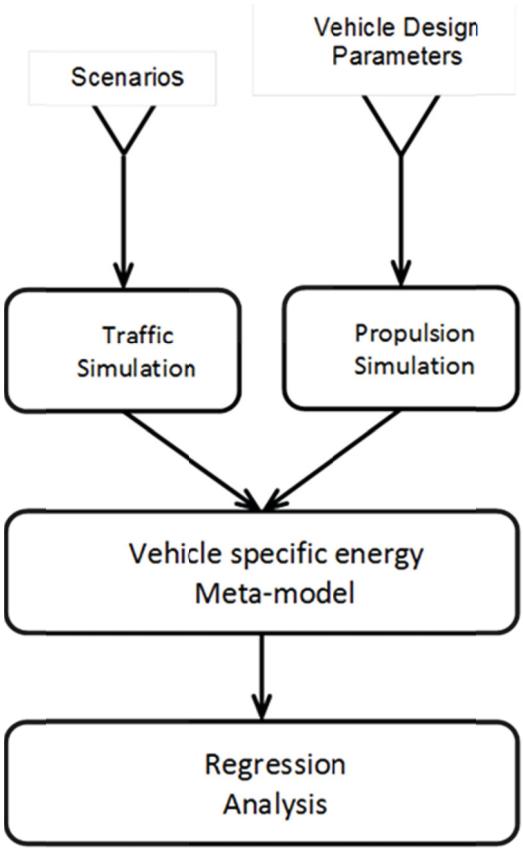


Fig. 1. Process flow for estimation of energy consumption based on integration of traffic and propulsion simulation

The scenarios are then coded into a traffic simulation code and tested under a variety of conditions that include road type, weather, traffic, gradient and vehicle parameters. The output of the traffic simulation is a set of drive cycles which are a time series of distance travelled, velocity, acceleration, lane changes.

Drive cycles are then converted into estimated energy consumption for each sample vehicle driving through a scenario. This can be done using a meta-model for energy consumption for the vehicle. In our project this meta-model was a set of energy maps, each map for a different cargo load on the vehicle. The maps were developed using regression analysis of surrogate data from propulsion modeling and

related acceleration and speed to energy consumption per distance travelled. Non-propulsion losses (accessory loads) were handled separately based on travel time.

The total energy consumption for each vehicle in a scenario under a given set of conditions is then analyzed using statistical regression to provide a predictive model for the scenario under the given conditions.

2.1 Propulsion Simulation

Propulsion system simulation is a desktop computer method for directly simulating vehicle drive cycles using a complete model of a vehicle propulsion system that represents key interactions between driver, environment, vehicle hardware and vehicle controls. There are a number of examples of tools, methods and applications of propulsion system simulation programs in the literature that use dynamic systems modeling to estimate energy consumption given a drive cycle. The primary purpose of these applications is to identify key design elements that influence performance, test control algorithm alternatives and determine the effect specific propulsion system features have on drivability. One significant example is the PSAT (Powertrain System Analysis Toolkit) (Argonne National Laboratory) tool developed in 1999 as part of a collaborative effort with U.S. OEM's (Ford, GM, and Chrysler).

Many automotive OEMs and Tier 1 suppliers have proprietary propulsion system modeling and simulation tools with a team of developers and simulators capable of computing energy consumption from drive cycles. Typically the drive cycles are from laboratory tests specified by the government regulations, obtained from driving studies or created by simulation. Generally these tools are supported by databases of proprietary hardware component information and controls strategies and calibration data specifically representative of the manufacturer's products. They are configured to support investigation of systems that design changes can be modeled through simulation. One such modeling application is Ford Motor Company's Corporate Vehicle Simulation Program (CVSP) that was used in the modeling effort described in this paper [3].

CVSP is a critical tool used mainly for projection of fuel economy capability of vehicles with internal combustion engines. These projections are used to make critical hardware and technology decisions that determine vehicle program content and ultimately impact vehicle program cost. Results from the CVSP simulations are also used to cascade targets to key subsystems and components (e.g. battery, power electronics and electric machines for HEV's).

Within Ford, a significant amount of time and effort has been invested in verifying the accuracy of CVSP simulations. This is critical for development of high confidence fuel economy roadmaps and subsystem/component targets for vehicle programs. With good system model accuracy, targets can be specified with much higher precision, thus avoiding over-design of components to deliver aggressive fuel economy targets. In the later stages of a program, an accurate system model can support vehicle testing for fuel economy attribute development. The model can be used to assess selected propulsion system control strategy and calibration changes which can help refine vehicle test plans and improve efficiency of vehicle test efforts.

The vehicle system model integral to CVSP is implemented in the Matlab/Simulink® environment using the Vehicle Model Architecture (VMA) standard [2]. Models of each VMA subsystem are stored in libraries and inserted into

the architecture for simulation using automated model configuration tools. The system model incorporates submodels for physical and control elements of the vehicle system. As an example, for power split hybrid electric vehicle system simulation, the set of submodels includes a power split transaxle subsystem model, a high voltage (HV) electrical subsystem model including high voltage battery and a model of the power split vehicle system controller. The system model represents the longitudinal dynamics of the chassis and one-dimensional rotational dynamics of the powertrain system. All mechanical and electrical component losses that have impact on energy usage and fuel consumption are included and distributed across the relevant subsystems. These losses are typically represented by component maps derived from bench/lab tests or high fidelity physical models. Further discussion of the CVSP simulation environment and how it is applied to electric vehicle system assessment can be found in [4] and [3].

2.2 Traffic Simulation

Traffic micro-simulations are proposed as a way to produce realistic drive cycles. These simulations have been developed for testing the performance of roadway designs and signal light timing schedules, and generally for improving the performance of transportation infrastructure. Typically they are used in the domain of the traffic engineer and not traditionally used for vehicle simulation.

Traffic micro-simulations are time-event driven simulations that implement a driver model for individual vehicles that are placed on a model roadway. They implement psychophysical driver models that employ vehicle physics and a physiological model of driver following behavior. A detailed discussion of driver models for micro-simulation modeling is found in [6].

Roadways are modeled as directed graphs in micro-simulation and individual vehicles placed on the model roadways have proven to be a reasonable way to model traffic flows that consider jams, congestion and different driver behaviors. Other factors can also be considered such as weather and topography.

A primary advantage of micro-simulation over other methods of producing drive cycles is that there is a straightforward analytical model that links physical features of the road, traffic and human perception to the creation of synthetic drive cycles. Models are calibrated using in-vitro data such as that collected in driving simulators, established psychological theory and observation traffic behavior from aircraft. Frequently micro-simulation software is calibrated for good results for bulk traffic flows consistent with those observed by traffic monitors or detection equipment placed in the roadway.

There are a number of traffic micro-simulation packages readily available from open-source, commercial and academic sources. In our study we used VISSIM [5]; a mature, full featured traffic simulation package. VISSIM is a time driven microscopic simulation package from PTV that can analyze private and public transport operations under constraints such as lane configuration, traffic composition, traffic signals, public transportation stops, etc., thus making it a useful tool for the evaluation of various alternatives based on transportation engineering and planning measures of effectiveness. VISSIM can be applied as a useful tool in a variety of transportation problem settings. Simulated Vehicles are allowed to run through a road model, each

vehicle having a driver model and a vehicle dynamics model. The driver model consists primarily of four parts; a psycho-physical following model [7]; [8], a lane changing model, a launch model and a speed holding model. The output of the driver model is the driver's desired speed, acceleration and lane angle. These are later modified to conform to the vehicle's performance limits.

In addition, VISSIM has two features that may be important in future work; the ability to introduce a user driver model with lane-changing and following behavior, and an interface for the dynamic routing function that allows exploration of routing algorithms using a programming interface.

2.3 The Scenarios

Three representative road types were used to build traffic scenarios. The road types were chosen to be exemplary of the types of roads that might populate a full scale analysis project; the residential street, urban highway and limited access highway. Road models were based on actual roads near Dearborn, MI, USA where the work was done and coded into the traffic simulation program. This allowed easy access to collect data and calibrate the models (see Fig. 2). Fig. 2 presents a 3-mile stretch along US I-96 (between exit 179 and 183) that was coded into the simulator as a representative section of freeway. The base model is a 3 lane road with no ramps entering or leaving the freeway. The traffic composition included 4% heavy goods vehicles and 2% battery electric vehicles. The remaining 94% were internal combustion vehicle of varying lengths consistent with personal transportation. To differentiate between a BEV and an internal combustion vehicle drivetrain, different desired acceleration profiles (speed vs. maximum acceleration desired by the driver) have been used. The vehicle input was set at 5000 vehicles per hour over 3 lanes. Stochastic distributions of driver desired speeds are defined for each vehicle type within each traffic composition. The desired speed of both the conventional cars and the BEV was a roughly normal distribution with a mean of 62 MPH (100 km/h) distributed between 83 MPH (130 km/h) and 50 MPH (80 km/h). At this speed aerodynamic drag exceeds all other vehicle specific load components except possibly accessory loads. If not hindered by other vehicles, a driver will travel at his desired speed with variations determined by the driver following model.

The urban highway model is based on a 6 mile stretch along US-24 (Fig. 3) with multiple traffic signals at intersections 1 mile apart. The vehicles enter at one end of the road and exit only at the other end. There were a total of nine synchronized traffic signals along the road, with multiple signals at some intersections. Although the actual road had 4 lanes along part of the stretch, the base model had 3 lanes throughout to simplify interpretation of the results. The traffic was composed of 98% conventional cars and 2% battery electric vehicles. The desired speed varied between 42 and 48 MPH (68 km/h – 77 km/h).

Traffic light timing was on roughly 60 second cycles such that during a typical evening rush hour packs of about 90 cars build up at a red light. The light would change to green and the vehicles would launch from a standstill. Except for the lead vehicles, each vehicle's launch rate was limited by the vehicle ahead. The pack of vehicles would reach the next light and stop for a few moments, and then continue to the next light.

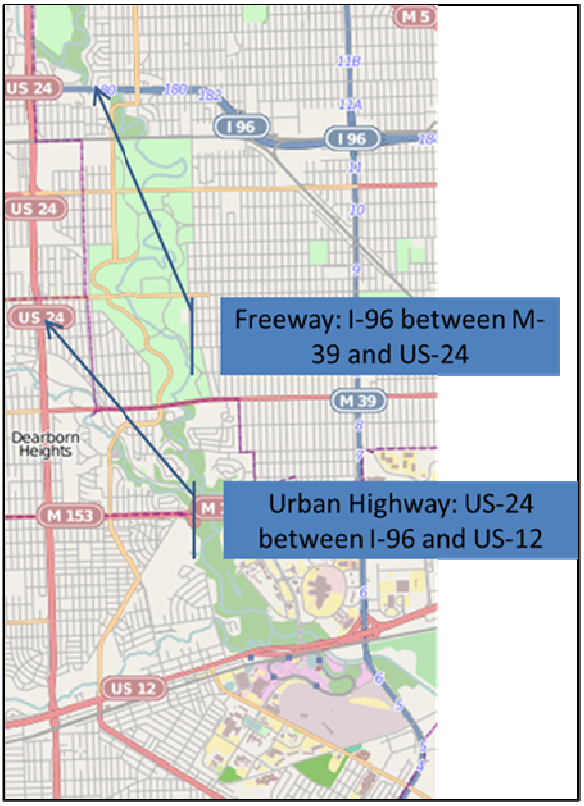


Fig. 2. The urban highway is a six mile stretch of Telegraph road and the freeway is a three mile stretch of I=96. For reference, the GPS coordinates of the intersection of M-39 and I-96 is 42.378780 and -83.216972.

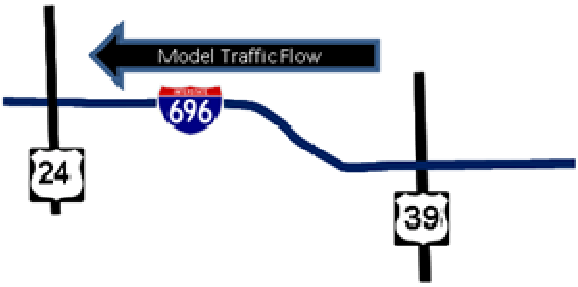


Fig. 3. This is a schematic view of the three miles stretch along I-696 between state routes 39 and 24 that was used as the "freeway model"

The residential road scenario was one mile long with multiple stop signs. The base model had 5 stop signs in each direction with a flow of 80 vehicles per hour in each

direction. Similar to most residential roads, there was a single lane in each direction. The desired speed of the vehicles had a distribution that varied between 22 and 28 MPH (36 and 46 km/h), based on the assumption that the median desired speed would be the speed limit, 25 MPH (41 km/h).

Deceleration into and launch from each stop sign was largely under the control of the driver model, not constrained by the vehicle ahead. There was a dwell time at each sign in which vehicle speed dropped to zero followed by a launch. The length of the dwell was based on cross traffic and the driver characteristics of each individual car.

The scenarios were created using a road model with varying external conditions. They were selected to explore the scenario space to determine which conditions were significant factors for energy consumption. The factors used were as follows:

- Road characteristics
  - Road Gradient
  - Number of lanes
  - Traffic characteristics
  - Vehicle flow rate
  - Vehicle mix (Number of trucks, buses, cars and battery electric vehicles)
- Driver characteristics
  - Desired speed
  - Use of cruise control
- Accessory load per unit time

Two types of energy consumption were considered in this analysis; propulsive energy consumption and accessory energy consumption. Propulsive loads were computed using maps of energy per distance travelled. Accessory energy consumption was in units of energy per unit time and kept constant through any given scenario.

The independent variables for the energy maps were vehicle speed and acceleration. Four maps were made for the BEV vehicle for different payloads weights; 1-4 occupants. The acceleration used in the energy calculation was the sum of road gradient acceleration and vehicle acceleration.

We determined both experimentally and using the Student-T analysis that 125 BEV test vehicles were necessary to get sufficient statistical power. So each scenario was run until 125 BEV had passed through the scenario. For each BEV the energy consumption was computed for each time step, multiplied by the distance of each time step and accumulated for the entire drive cycle. This was added to the energy consumption attributable to the accessory loads and saved. The average and standard deviation of all the drive cycles were then computed for each scenario, and the scenarios were plotted and compared to determine the main effects.

3 Results of the Energy Consumption Modeling

The results are presented in the following manner. First the results of energy consumption under different scenarios for each of the road types are presented followed by a comparison of these results across road types. The tables give the mean energy

consumed by a battery electric vehicle for that particular scenario. The units are Watt-hours unless mentioned otherwise. The original analysis was done with data for a specific vehicle, but because of the proprietary nature of this data it has been normalized and the results are more qualitative.

3.1 Freeway

Fig. 4 shows the average energy required by 125 battery electric vehicles to travel the 3 mile stretch of freeway at various gradients and traffic flows. The bold lines in Fig. 4 represent the mean energy consumed and the dotted lines represent the 95% confidence interval. It can be seen that gradient has a prominent effect on the energy consumption of a battery electric vehicle. There is a rapid increase in the energy values as we move from a gradient of -4% to 4%. This is because, the vehicle needs more energy to climb uphill (positive gradient) and it can gain energy through regenerative braking while going downhill (negative gradient). Congestion has a much smaller effect than gradient on energy consumption. There is a slight reduction in energy as the flow conditions approach a congested scenario. This effect is directly related to the decrease in the speeds for congested flows.

Table 1 shows the impact of desired speed and the number of lanes on the energy consumption under different flow regimes. Fig. 5 presents the data in such a way as to show that vehicle speed is much more significant than the number of lanes. A vehicle travelling at around 60 mph will consume about 30% more energy than a vehicle travelling at around 50 mph. Also, the number of lanes on the freeway doesn't seem to have an effect on the overall energy consumption.

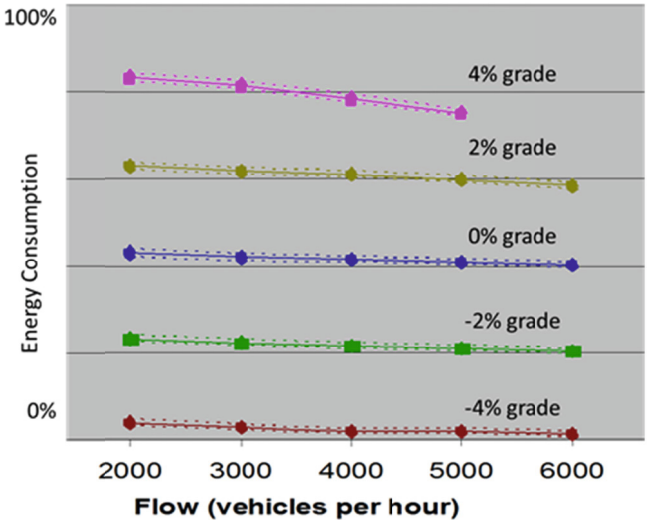


Fig. 4. Energy consumption for vehicle travelling on the freeway at different grades

Table 1. The effect of desired speed and number of lanes at different flow rates

	60 mph		50 mph	
Flow (VPH)	**3 lanes**	**4 lanes**	**3 lanes**	**4 lanes**
2000	1.16	1.18	0.88	0.88
3000	1.14	1.15	0.86	0.87
4000	1.12	1.14	0.85	0.86
5000	1.11	1.15	0.85	0.86
6000	1.09	1.14	0.84	0.86

Table 2 shows a comparison of the energy usage of vehicles travelling in cruise control with that of vehicles not travelling in cruise control for two different speeds. It is assumed that in the cruise control scenario only the battery electric vehicles are in cruise control mode. All other vehicles are travelling without cruise control.

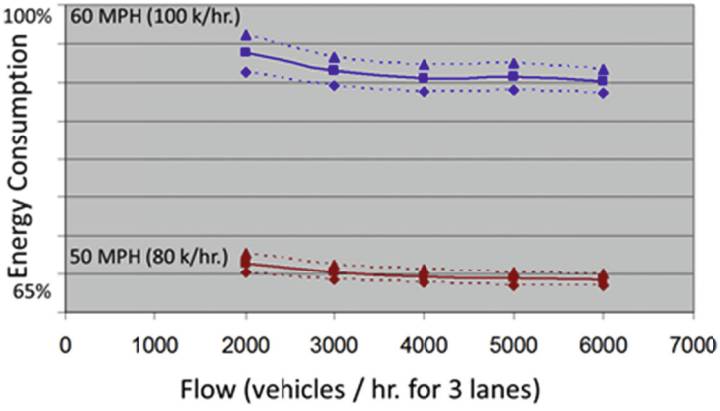


Fig. 5. The figure shows energy consumption for two vehicle parameter sets, one with drivers whose average desired speed is 60 MPH and the other averaging 50 MPH

Table 2. Effect of gradient on energy consumption at different vehicle flow rates for a 3 lane road

	Grade				
Flow (vehicles per hour)	**-4%**	**-2%**	**0%**	**2%**	**4%**
2000	0.09	0.58	1.08	1.59	2.11
3000	0.07	0.56	1.06	1.56	2.06
4000	0.05	0.55	1.04	1.55	1.98
5000	0.05	0.53	1.03	1.51	1.90
6000	0.03	0.51	1.01	1.48	

Table 3. The effects of traffic load, desired speed and cruise control on energy consumption

	60mph (100 k/hr.)		50mph (80 k/hr.)
Flow (VPH)	**No cruise**	**Cruise**	**No cruise**
2000	1.20	1.11	0.91
3000	1.18	1.11	0.89
4000	1.16	1.11	0.89
5000	1.15	1.09	0.88
6000	1.13	1.09	0.87

From Table 3 and Fig. 6, it can be seen that the vehicles travelling in cruise control use significantly lower energy than the vehicles travelling without cruise. The fluctuations in acceleration/deceleration and hence the speed results in higher energy consumption for vehicles which are not using cruise control. The drop in energy consumption with increase in flow values is directly related to the drop in speeds as the flow conditions become congested. It should also be noted that the energy consumption and its variation remains fixed across different flow values when the cruise control is set to 50mph. But, in the case of cruise control at 60mph, there is a larger statistical variance in energy usage (dotted lines diverge) as the flow values increase. This is because at high flows, the vehicles are unable to maintain a cruise speed of 60mph due to the increase in flow density. But, the vehicles seem to maintain a cruise speed of 50mph even when the traffic flow increases.

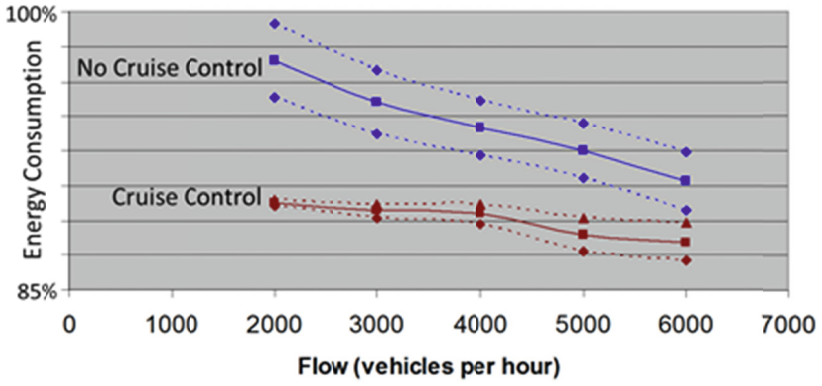


Fig. 6. The effect of traffic load on energy consumption both with and without cruise control driver models is demonstrated for 125 vehicles

3.2 Urban Highway

The results show the mean energy required by a battery electric vehicle to travel the 6 mile stretch. The main difference between an urban road and a freeway would be the stop-go behaviour of the vehicles at traffic signals. As a result, because of the regenerative capability of a battery electric vehicle, the energy consumed per mile will be

lower on an urban road. Also, the lower speeds on urban roads will result in lower energy consumption. But the trade-off between an urban road and a freeway will be in the travel times. A comparison is presented in the later sections of this chapter.

Table 4 shows the mean energy consumed on the urban highway by a BEV under different traffic flow and gradient conditions. Similar to the freeway, the gradient has a very significant effect on the energy consumption on the urban highway as well. It is significantly greater than the effect of traffic loads.

Table 4. The effect of grade and traffic load (VPH) on energy consumption for the urban highway

	Grade		
Flow (VPH)	**-2%**	**0%**	**2%**
1000	16%	57%	100%
2000	15%	56%	99%
3000	14%	56%	98%
4000	14%	55%	0%

Also, simulations have been performed to understand the effect of the number of lanes on the overall energy consumption. It has been observed that the number of lanes has a very small effect on consumption in these traffic flow scenarios.

3.3 Residential Street

The results for residential roads show the mean energy required by a battery electric vehicle to travel the 1 mile stretch of residential street with multiple stop signs. The stop signs and the lower speeds on residential roads will significantly reduce the energy consumption of a BEV due to aerodynamic drag, but will increase the significance of other factors such as road grade or stop starts. Due to regenerative braking the effects of stop start are expected to be smaller on BEV than would be normally expected from conventional vehicles (see Table 5).

Fig. 7 shows the effect of gradient on energy consumption. It is expected that the energy usage will be negative for a gradient of -4%. This shows that the battery of the vehicle is gaining energy because of the regenerative braking.

Up to this point the effect of various traffic/road/driver characteristics on the energy usage for each of the individual road types has been discussed. In the next section, a detailed comparison of the energy usage across the three different road-types is presented to better understand how some of the scenarios impact the lowest energy routes and distance-to-empty.

Table 5. The effect of stop-signs per mile on the energy consumption

Number of stop-signs per mile	Energy (W-hours)
5	149.5
10	148.4
15	148.0

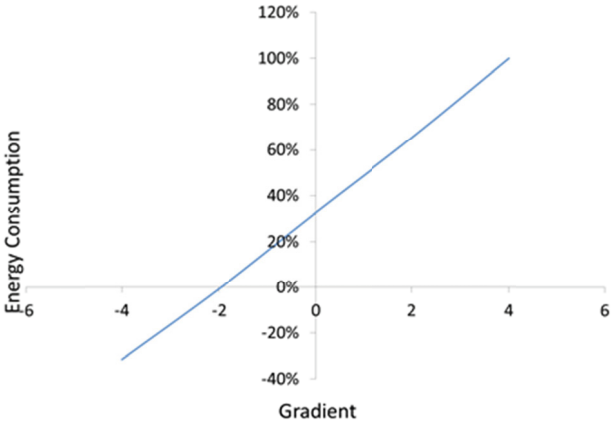


Fig. 7. The effect of gradient on average energy consumption for 125 vehicles for the residential street

3.4 Comparison across Road Types

Figure 8 shows the impact of accessory loads on the energy consumption across the three road types. The bars represent the energy usage in W-hours per mile and the three bold dots represent the average travel time in seconds to travel a distance of one mile on each of those roads. It can be seen that for a given accessory load the energy usage is the lowest for a residential road and highest for a freeway, mainly because of low speeds and stop-go nature of traffic on a residential road. In fact, the energy usage per mile with 400W accessory load is more than halved from a freeway to a residential road. But, the travel time on a residential road is almost four times that on a freeway. This shows a trade-off between travel times and energy usage. Increase in the accessory loads has very little impact on the energy usage on a freeway where high speed is the primary driver of energy consumed. On the other hand, the accessory loads drastically affect the energy usage on a residential road to such an extent that, the energy used per mile with 2000W accessory load is almost the same as that on an urban road.

Fig. 8 shows that gradient has a very dominating effect on the energy consumption of a battery electric vehicle. For a particular road type, the increase in energy consumption with every 2% increase in gradient is about 150W-hrs per mile. High gradients like 4% or -4% are not very common for a freeway.

The three road types each have different speed limits, and for each simulation the drivers' 'desired speed' is input as a statistical distribution around the speed limit for the road. Some drivers will drive above the limit and others below, but most will be close to the speed limit. Fig. 8 shows the energy consumption for vehicles with the following desired speeds: Freeway – 65 MPH, Urban road –45 MPH, Residential road – 25 MPH. These values are almost identical to the speed limits on the corresponding road types. The graph shows that the speed of the vehicles has a strong influence on the energy usage. Also, the stop-go behavior of vehicles on urban and residential roads has a significant effect on energy consumption because of regenerative braking.

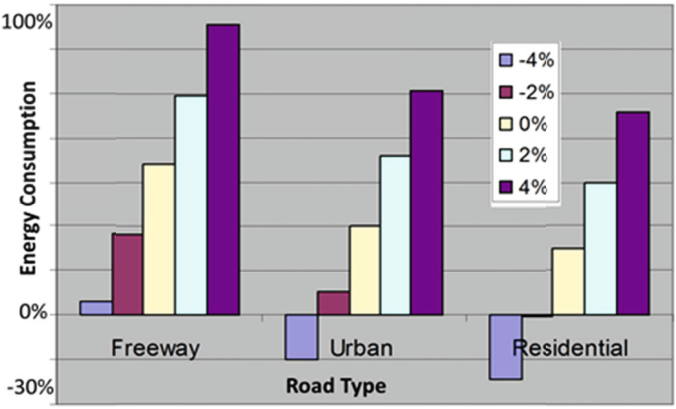


Fig. 8. Comparison of the effect of road gradient on the three types of roads: freeway, urban highway and residential road

4 Energy Consumption Calculation

Using the results presented in Fig. 8 and Fig. 9 It is possible to develop a meta-model to estimate energy consumption over a section of road if the distance d is known and the travel time t can be estimated.

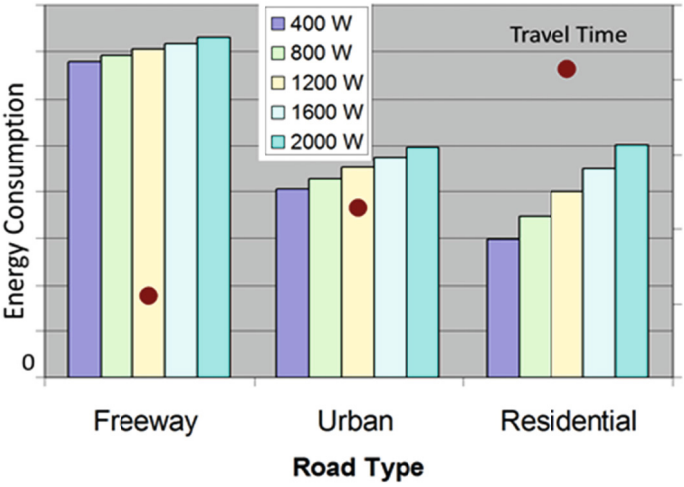


Fig. 9. The effect of accessory loads on the energy consumption on flat ground on the three road types

The energy consumed on a segment is given by an equation of the form:

$$E = Pt + Wd \quad (1)$$

Where:

E = The energy consumed
P = The power consumed by accessory loads
W = The work done on the vehicle based on distance traveled
t = time
d = distance traveled

Power is the sum of the time dependent terms consisting primarily of accessory loads. A model is needed to predict those loads based on factors such as climate control requirements, lighting requirements, windshield wipers, etc. Lacking that model we use different levels of accessory loads to create scenarios.

The prior results demonstrate that speed and gradient are major factors in work, and to a lesser extent the road type. We take work to be the sum of gradient factors B and speed factors A. An equation for B is developed for each road type in Fig. 8 where s is the slope of the road in degrees.

The speed (V) component (C) of work is given by:

$$C = A_{aero}V^2 + B_{aero}V + C_{aero} \quad (2)$$

Substituting (B+C) for W and A for P, energy consumption can be computed as:

$$E = At + (B+C)d \quad (3)$$

The factors $A_{freeway}$, $C_{freeway}$, $A_{highway}$, $C_{highway}$, A_{street}, C_{street}, D are fit functions primarily related to the road, and the factors A_{aero}, B_{aero}, C_{aero} are fit factors related to the aerodynamics of the vehicle.

This equation was plotted for a set of 150 routes generated at a mapping website assuming no gradient. The results are plotted in Fig. 10 where it is seen that under high accessory loads there is a minimum energy speed at about 20 MPH. In the case where there is low accessory load the energy consumption appears to asymptotically approach zero energy consumption. In fact, in the low accessory load cases there is also an optimal energy consumption speed, but at a much lower speed than vehicles are ordinarily driven.

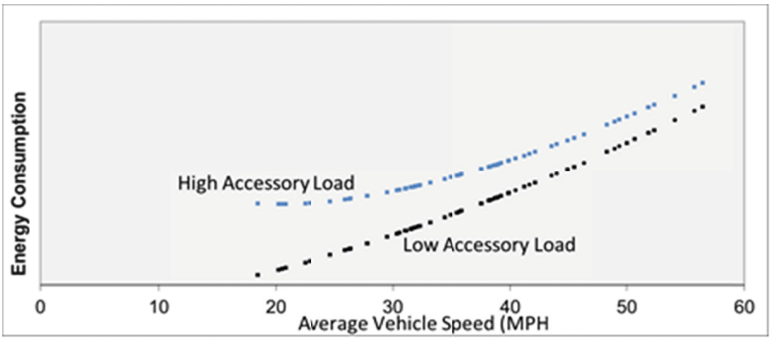


Fig. 10. The meta-model results applied to routes through 20 different cities in the US

5 Conclusions

A methodology for creating meta-models for energy consumption from surrogate data from simulation has been demonstrated. The method was used to determine the main effects and a simple meta-model that can be run in embedded processors in the vehicle or on off-board computing platforms has been developed. The energy consumption meta-model enables vehicle functions such as; distance to empty, remaining charge needed and low-energy routing. These in turn enable vehicle features that give the driver greater confidence in the product and therefore improves acceptance and deployment of electric vehicles.

The traffic simulation code VISSIM and the propulsion modelling code sCVSP were combined via a surrogate meta-model to provide energy consumption data for developing a comprehensive energy consumption meta-model.

By developing different scenarios it was possible to determine main effects such as road gradient and vehicle speed. It is widely believed that the driver has a large impact on energy consumption. From our results we see this is likely mostly a factor of the driver's desired speed, how it is moderated by traffic. Probably route choice is a significant factor that would differentiate drivers.

Road gradient is another important factor, but is mostly a factor of the potential energy build-up or loss going up or down a hill. BEV are different from hybrid electric and conventional vehicles in that much of the energy lost going uphill is regained coming down. Unlike hybrid vehicles, the BEV battery is large enough to recover this energy on many hills.

A third main effect is accessory loads. These come from many sources in a BEV, but the largest factor is generally warming or cooling the vehicle. This can easily account for 50% of the energy consumed by the vehicle and is the only factor that favours faster travel time.

Each of these three areas; vehicle speed, road gradient and climate control require further study. Vehicle speed on an un-crowded road is largely a factor of how fast the driver wishes to drive. This may vary depending on the speed limit and on safety considerations. On a crowded road vehicle speed also can be determined by how effective a driver wishing to travel quickly is at maintaining this desired speed while interacting with slower moving vehicles. The biggest factor in velocity drag is aerodynamic drag which is assumed in our models to increase with the square of the velocity. However, the situation on the public roads is quite a bit more complicated where there is wind, ground turbulence, air density, disturbance by nearby traffic and other factors to consider.

Road gradient is expected to always be a major contributor to energy consumption, but other road factors that are not included in our model are soft road surfaces, partially inflated tires and many other factors. For the energy consumption meta-model to be accurate it is necessary to break a road into segments that either rise or fall. If a segment goes up then down a hill, the energy consumption will not be accurate. How real road surfaces will be broken into segments that provide good results must be considered.

Finally, the predicting of accessory loads is quite complex and is the topic of another paper. Loss by the climate control system is controlled by several factors. External factors include ambient temperature, humidity and sun load. The vehicle configuration is a

major factor as are several controls that are under the manual control of the driver. However, this is changing and in the near future most climate control features operated by the driver will migrate to a control system for driver convenience and to meet new Corporate Average Fuel Economy (CAFÉ) standards.

References

1. Argonne National Laboratory (N.D.). Argonne TTRDC - Modeling, Simulation & Software - PSAT. Transportation Technology R&D Center, `http://www.transportation.anl.gov/modeling_simulation/PSAT/index.html` (retrieved October 21, 2011)
2. Belton, C., Bennett, P., Burchill, P., Copp, D., Darnton, N., Che, J., et al.: A Vehicle Model Architecture for Vehicle System Control Design. In: 2003 SAE World Congress. SAE Technical Paper Series. SAE International, Detroit (2003)
3. Jennings, M., Brigham, D., Meng, Y., Bell, D.: A Comparitive Analysis Methodology for Hybrid Electric Vehicle Concept Assessment. In: 2004 SAE World Congress. SAE International, Detroit (2004)
4. Meng, Y.J.: Test Correlation Framework for Hybrid Electric Vehicle System Model. In: 2011 SAE World Congress. SAE International, Detroit (2011)
5. PTV AG, VISSIM 5.20 User Manual. D-76131. Planung Transport Verkehr AG, Karlsruhe (2009)
6. Tomer, T.: Integrated Driving Behavior Modeling. Department of Civil and Environmental Engineering, MIT, Boston (2003)
7. Weidemann, R.: Simulation des Straßen-verkehrsflusses. Heft 8: Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe (1974)
8. Weidemann, R.: Modeling of RTI-Elements on multi-lane roads. In: Advanced Telematics in Road Transport Edited by the Commission of the European Community. Commission of the European Community, DG XIII, Brussels (1991)

Full Papers

Speeding Up the Evaluation of a Mathematical Model for VANETs Using OpenMP

Carolina García-Costa, Juan Bautista Tomás-Gabarrón, Esteban Egea-López, and Joan García-Haro

Department of Information and Communications Technologies,
Universidad Politécnica de Cartagena (UPCT), Plaza del Hospital 1, Cartagena, Spain
{carolina.garcia,juanba.tomas,esteban.egea,joang.haro}@upct.es

Abstract. Vehicular Ad-hoc Networks (VANETs) are having a significant impact on Intelligent Transportation Systems, specially on the improvement of road safety. Cooperative/Chain Collision Avoidance (CCA) application comes up as a solution for decreasing accidents on the road, therefore it is highly convenient to study how the system of vehicles in a platoon will behave at different stages of technology deployment until full penetration in the market. In the present paper we describe an analytical model to compute the average number of accidents in a chain of vehicles. The use of this model when the CCA technology penetration rate is not 100% leads to a vast increase in the number of operations. Using the OpenMP directives for parallel processing with shared memory we achieve a significant reduction in the computation time consumed by our analytical model.

Keywords: OpenMP, VANET, Supercomputing, Cooperative/Chain Collision Avoidance application.

1 Introduction

Vehicular networks, also known as VANETs, are defined as *ad-hoc* mobile networks with two main communication features. On the one hand, VANETs are in charge of transmitting information among vehicles (V2V communications). In this first case, cars carry out the information interchange without any infrastructure support for regulating the access. On the other hand, an intercommunication among vehicles and infrastructures also exists (V2I communications), making possible a connection through cars and a backbone network, reaching in this way those vehicular entities allocated out of the direct communication range.

One of the aims of vehicular networks development is the improvement of road safety. The main goal of these innovative systems is to provide drivers a better knowledge about road conditions, decreasing the number of accidents and their severity, and simultaneously aiding to a more comfortable and fluent driving. Other vehicular applications are also considered, such as Internet access, driving cooperation and public information services support.

A Cooperative/Chain Collision Avoidance (CCA) application [1] uses VANET communications for warning drivers and decreasing the number of traffic accidents. CCA

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 23–37.
DOI: 10.1007/978-3-642-34336-0_2

takes advantage of vehicles with cooperative communication skills, in a way that these cars are able to react to possible accident risks or emergence situations. The CCA mechanism generates an encapsulated notification which is sent as a message through a one-hop communication scheme to all vehicles within a potential danger coverage (relay schemes are also possible). It should be noted that the establishment of this VANET application will be deployed gradually, equipping vehicles with the proper hardware and software so as they can communicate in an effective way within the vehicular environment.

In our research we consider a platoon (or chain) of N vehicles following a leading one. The leading vehicle stops instantly and the following vehicles start to brake when they are aware of the risk of collision, because of a warning message reception or the perception of a reduction in the speed of the vehicle immediately ahead. To test the worst case situation, vehicles cannot change lane or perform evasive maneuvers.

We have developed a first approach mathematical model to calculate the average percentage of accidents in the platoon, varying the number of considered vehicles, their average speed, the average inter-vehicle spacing and the penetration ratio of the CCA technology. Specifically when the CCA penetration ratio is taken into account, the growth in the number of operations of the analytical model is such that the sequential computation of a numerical solution is no longer feasible. Consequently, we resort to the use of the OpenMP parallelization techniques for solving those computational cases considered as unapproachable by means of sequential procedures.

Additionally, we execute our programs in the Ben-Arabi Supercomputing environment [2], taking the advantage of utilizing the fourth fastest Supercomputer in Spain. In the current work we show how the parallelization techniques coordinated with supercomputing resources make the simulation process a more suitable and efficient one, allowing a thorough evaluation of the CCA application.

The remainder of this paper is organized as follows. In Section 2 we briefly review the related work. In Section 3 the OpenMP environment is briefly reviewed and the Ben-Arabi Supercomputer architecture introduced. A description of the mathematical model, its implementation and parallelization are provided in Sections 4 and 5. Finally, some results are shown and discussed in Section 6 to illustrate the performance of the resulting parallel algorithm. In this section it is also described our unsuccessful experience of using the MPI parallelization technique to further reduce the computation times. Conclusions and future work are remarked in Section 7. Let us mention that this paper is an extension of the work presented by the authors in [3].

2 Related Work

So far, most typical High Performance Computing (HPC) problems focused on those fields related with certain fundamental problems in several areas of science and engineering. Other typical applications are the ones related to commerce, like databases and data mining [4]. That is the reason why we consider our VANET mathematical model approximation as a non-classical issue to be solved under HPC conditions, contributing to extend the use of supercomputing to other fields of interest.

In the implementation of our mathematical model we parallelize a sparse matrix-vector multiplication. This operation is considered as a relevant computational kernel in

scientific applications, which performs not optimally on modern processors because of the lack of compromise between memory and computing power and irregular memory access patterns [5]. In general, we find quite a lot of done work in the field of sparse matrix-vector multiplications using parallelization techniques [6], [7], [8]. These papers study in depth the optimal performance of this operation, but in this paper, we show that even using a simpler parallelization routine, the computation time is noticeably shortened.

Several mathematical models have been developed to study different aspects of VANETs. Most of them are related with the vehicle routing optimization [9], [10], the broadcasting methods [11], [12], [13], the mobility of vehicles [14], [15] and the communication delay time [16], [17], [18]. Other related VANET issues have been studied as well, like network connectivity [19], or survivability [20]. In this paper we focus on collision models for a chain of vehicles, particularly those based on physical parameters to assess the collision process itself [21], [22], [23].

However in an attempt of searching related work we find that few work has been done specifically regarding to the parallelization of these VANET mathematical models, strictly speaking. Moreover, to the best of our knowledge, only the vehicle routing problem has been approached using parallelization techniques [24], [25], [26].

Summing up, in this paper we describe a preliminary model (although computationally expensive) for a CCA application to compute the number of chain collisions and we address the benefits of using parallelization techniques in the VANET field.

3 Supporting Tools

3.1 The OpenMP Technique

OpenMP is a well-known open standard for providing parallelization mechanisms to multiprocessors with shared memory [27]. OpenMP API supports shared memory programming, multi-platform techniques for the programming languages like Fortran, C and C++, and for every architecture including Unix and Windows platforms. OpenMP is a scalable and portable model developed for hardware and software distributors which provides shared memory programmers with a simple and flexible interface for developing parallel applications which can run not only in a personal computer but also in a supercomputer.

OpenMP uses the parallel paradigm known as *fork-join* with the generation of multiple threads, where a heavy computational task is divided into k threads (*forks*) with less weight and afterwards it collects their results and combines them at the end of the execution in a single result (*join*).

The master thread runs sequentially till it finds an OpenMP guideline and since this moment a bifurcation is generated with the corresponding slave threads. These threads can be distributed and executed in different processors, decreasing in this way the execution time.

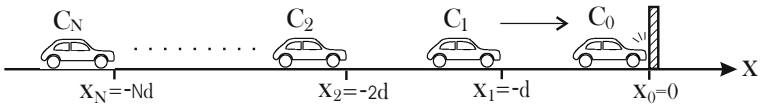


Fig. 1. The scenario under consideration. d is the average inter-vehicle distance.

3.2 The Ben-Arabi Supercomputer

Our model is executed under the Ben-Arabi supercomputer resources, which is placed in the Scientific Park of Murcia (Spain). The Ben-Arabi system consists of two different architectures; on the one hand the central node HP Integrity Superdome $SX2000$ with 128 cores of the Intel Itanium-2 dual-core Montvale (1.6 Ghz, 18 MB of cache L3) processor and 1.5 TB of shared memory, called Ben. On the other hand, Arabi is a cluster consisting of 102 nodes, which offers a total of 816 Intel Xeon Quad-Core E5450 (3 GHz y 6 MB of cache L2) processor cores and a total of 1072 GB of shared memory.

We run our mathematical model within a node of the Arabi cluster environment using 2, 4 and 8 processors in order to compare the resulting execution times. Les us remark that we are using a shared memory parallelization technique, so we are not allowed to combine the use of processors from different nodes.

Next we summarize the technical features of the cluster:

- Capacity: 9.72 Tflops.
- Processor: Intel Xeon Quad-Core E5450.
- Nodes number: 102.
- Processors number: 816.
- Processors/Node: 8.
- Memory/Node: 32 nodes of 16 GB and 70 of 8 GB.
- Memory/Core: 3 MB (6 MB shared among 2 cores).
- Clock frequency: 3 Ghz.

4 Model Description

We are interested in evaluating the performance of a CCA application for a chain of N vehicles when the technology penetration rate is not 100%. We consider the inter-vehicle spacing is normally distributed and each vehicle C_i, $i \in \{1, ..., N\}$, moves at constant velocity V_i. Vehicles drive in convoy (see Figure 1), reacting to the first collision of another car, C_0, according to two possible schemes: starting to brake because of a previously received warning message transmited by a collided vehicle (if the vehicle is equipped with CCA technology) or starting to decelerate after noticing a reduction in the speed of the vehicle immediately ahead (if the vehicle under consideration is not equipped with CCA technology).

With this model the final outcome of a vehicle depends on the outcome of the preceding vehicles. Therefore, the collision model is based on the construction of the following probability tree. We consider an initial state in which no vehicle has collided. Once the danger of collision has been detected, the first vehicle in the chain C_1 (immediately after the leading one) may collide or stop successfully. From both of these states two

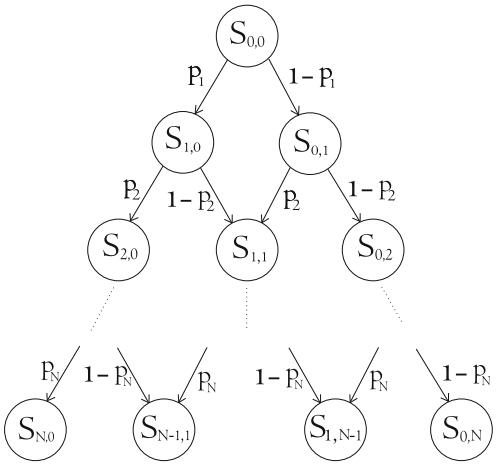


Fig. 2. Probability tree diagram that defines the model. $S_{i,j}$ represents the state with i collided vehicles and j successfully stopped vehicles.

possible cases spring as well, that is either the following vehicle in the chain C_2 may collide or stop successfully. And so on until the last vehicle in the chain. At the last level of the tree we have $N + 1$ possible outcomes (final outcomes) which represent the number of collided vehicles in the chain, that is, from 0 to N collisions (Figure 2).

The transition probability between the nodes of the tree is the probability of collision of the corresponding vehicle in the chain p_i (or its complementary). These probabilities are calculated recursively, regarding different kinematic parameters, as the average velocity of the vehicles in the chain (used to compute the distance to stop), the average inter-vehicle distance and the driver's reaction time, among others.

We start calculating the collision probability of the nearest to the incidence vehicle, C_1. The position of C_i when it starts to decelerate is normally distributed with mean $\mu_i = -i * d$ and standard deviation $\sigma = d/2$, where d is the average inter-vehicle distance. Vehicle C_1 will collide if and only if the distance to C_0 is less than the distance that it needs to stop, D_s, so its collision probability is given by:

$$p_1 = 1 - \int_{-\infty}^{-L-D_s} f(x; \mu_1, \sigma)\, dx \ , \tag{1}$$

where L is the average vehicle length and $f(x; \mu, \sigma)$ is the probability density function of the normal distribution with mean μ and standard deviation σ.

To compute the collision probability of the second vehicle we will use the average position of the first vehicle when it has stopped (either by collision or successfully stop). This average position is determined by:

$$\overline{X_1} = \int_{-\infty}^{-L} x \cdot f(x; \mu_1 + D_s, \sigma)\, dx \ + \ (-L) \cdot \int_{-L}^{+\infty} f(x; \mu_1 + D_s, \sigma)\, dx \ . \tag{2}$$

The second term of the sum means that the vehicle cannot cross the position $-L$ when it collides, since we are assuming that when a vehicle collides it stops instantly at the point of collision.

Once we have obtained $\overline{X_1}$ we can compute p_2, and recursively we can obtain all the collision probabilities:

$$p_i = 1 - \int_{-\infty}^{\overline{X_{i-1}}-L-D_s} f(x;\mu_i,\sigma)\,dx, \quad i = 2,\ldots,N\ , \tag{3}$$

where

$$\overline{X_i} = \int_{-\infty}^{\overline{X_{i-1}}-L} x\cdot f(x;\mu_i+D_s,\sigma)\,dx + (\overline{X_{i-1}}-L)\cdot\int_{\overline{X_{i-1}}-L}^{+\infty} f(x;\mu_i+D_s,\sigma)\,dx,\ \ i=2,\ldots,N\ . \tag{4}$$

We want to remark that this model for the collision probabilities is a preliminary approximation and does not describe realistically the collision process. However, the method to compute the probabilities of the path outcomes is independent of the correctness or accuracy of the transition probabilities used, and the goal of this paper is to evaluate the benefits of parallelization for this technique to compute the average number of accidents. An improved model for the transition probabilities can be found in [28].

Let us note how every path in the tree from the root to the leaves leads to a possible outcome involving every vehicle in the chain. The probability of a particular path is the product of the transition probabilities that belongs to the path. Since there are multiple paths that lead to the same final outcome (leaf node in the tree), the probability of that outcome will be the sum of the probabilities of every path reaching it.

In order to compute the probabilities of the final outcomes, we can construct a Markov chain whose state diagram is shown in Figure 2 and is based on the previously discussed probability tree. It is a homogeneous Markov chain with $\frac{(N+1)(N+2)}{2}$ states,

$$(S_{0,0},\ S_{1,0},\ S_{0,1},\ldots,\ S_{N,0},\ S_{N-1,1},\ldots,\ S_{1,N-1},\ S_{0,N})\ . \tag{5}$$

The transition matrix P of the resulting Markov chain is a square matrix of dimension $\frac{(N+1)(N+2)}{2}$, which is a sparse matrix, since from each state it is only possible to move to two of the other subsequent states.

Then, we need to compute the probabilities of going from the initial state to each of the $N+1$ final states in N steps, which are given by matrix P^N. Therefore, the final outcome probabilities are the last $N+1$ entries of the first row of the matrix P^N.

Let Π_i be the probability of reaching the final outcome with i collided vehicles, that is, state $S_{i,N-i}$. We obtain the average of the total number of accidents in the chain using the weighted sum:

$$N_{acc} = \sum_{i=0}^{N} i\cdot\Pi_i\ . \tag{6}$$

Our purpose is to evaluate the functionality of the CCA system depending on the current penetration rate of this technology. So that, we have to solve the model assuming

different technology penetration ratios. This assumption implies that we have to calculate the number of collisions once for each of the possible combinations in the chain of vehicles equipped with and without CCA technology, that is,

$$\binom{N}{m} = \frac{N!}{(N-m)!\,m!}\,, \tag{7}$$

where N is the total number of vehicles in the chain and m is the number of vehicles equipped with the CCA technology. It is worth to notice that the number of combinations for m vehicles set with CCA technology and $N - m$ without it is the same that for $N - m$ vehicles with CCA and m without it. Therefore, in order to analyze the computation time, we solve the model varying the CCA penetration rate between 0% and 50%, since the rest of cases are computationally (but not numerically) identical. As we can see in Table 1, the number of combinations grows quickly by an increase on the CCA penetration rate as well as by an increase on the number of vehicles.

Table 1. Number of combinations of $N = \{10, 20, 30\}$ vehicles with and without CCA technology

CCA%	10 veh.	20 veh.	30 veh.
0%	1	1	1
10%	10	190	4060
20%	45	4845	593775
30%	120	38760	14307150
40%	210	125970	86493225
50%	252	184756	155117520

In addition to that, we also aim at evaluating the impact on the number of accidents of the inter-vehicle distance d, varying this parameter in a wide range.

5 Implementation

In this section we firstly introduce the algorithm for the model implementation (Algorithm 1) and then, we explain the method we have used to parallelize it.

Examining the algorithm we can make the following observations:

1. The iterations of the *for* loop that covers the number of *Combinations* resulting from the CCA technology penetration rate are independent for each other, so they can be executed in parallel by different threads.
2. The same occurs with the *for* loop that covers the *RangeOfDistances* (for the inter-vehicle spacing) to be evaluated.
3. Since the collision probabilities of the vehicles in the platoon is computed recursively, each iteration of the *for* loop that considers each vehicle in the chain needs the results of the preceding iteration, so this loop should be executed sequentially.
4. To obtain the first row of matrix P^N we have to multiply N times a vector of dimension $\frac{(N+1)(N+2)}{2}$ by a matrix of dimension $\frac{(N+1)(N+2)}{2} \times \frac{(N+1)(N+2)}{2}$. The vector-matrix multiplication can be also parallelized so that each thread executes the multiplication of the vector by part of the matrix columns. However, the N multiplications should be done one after the other, that is, sequentially.

Algorithm 1. Computation of the number of collisions in a chain of vehicles.

for all $comb$ in $Combinations$ **do**
  for all d in $RangeOfDistances$ **do**
    for $i = 1$ **to** N **do**
      $p_i = f(p_{i-1}, comb, d, i, veloc, reactTime)$
    end for
    for $j = 0$ **to** N **do**
      $\Pi_j = P^N(1, \frac{(N+1)(N+2)}{2} - j)$
    end for
    $N_{acc} = \sum_{j=0}^{N} j \cdot \Pi_j$
  end for
end for

Table 2. Resulting programs with different parallelized tasks. X means that the corresponding parallelization takes place.

Program	A	B	C
Program 1			
Program 2	×		
Program 3		×	
Program 4			×
Program 5	×	×	
Program 6	×		×
Program 7		×	×
Program 8	×	×	×

For the sake of clarity, we will parallelize the following tasks:

- A: Vector-Matrix multiplication.
- B: Average inter-vehicle distance variation.
- C: Technology penetration rate variation.

Next, we will combine the different parallelized tasks (see Table 2) and execute the resulting programs in order to assess the actual improvement obtained from each one.

6 Results

In this section we summarize the results obtained by executing the programs shown in Table 2 in a node of the Arabi cluster. We have used 2, 4 and 8 processors in order to assess the improvement on the execution time achieved by each one.

The parameters used to execute the model are the following:

- CCA penetration rate: $0\% - 50\%$, in 10% steps.
- Average inter-vehicle distance: $6 - 70\ m$, in 1 meter steps.
- Number of vehicles: 20 vehicles.
- Average velocity: $33\ m/s$.
- Average driver's reaction time: $1\ s$.

Table 3. Execution times in minutes and speedup (SU) for each program using 2 processors

	0%		10%		20%		30%		40%		50%	
	Time	**SU**	**Time**	**SU**	**Time**	**SU**	**Time**	**SU**	**Time**	**SU**	**Time**	**SU**
P1	0.002	1.00	0.307	1.00	7.848	1.00	62.876	1.00	203.896	1.00	297.975	1.00
P2	0.002	1.00	0.247	1.24	6.694	1.17	40.693	1.54	125.658	1.62	188.396	1.58
P3	0.001	2.00	0.175	1.75	4.315	1.82	33.858	1.86	110.142	1.85	159.558	1.87
P4	0.003	0.67	0.147	2.09	3.655	2.15	29.323	2.14	95.671	2.13	157.483	1.90
P5	0.001	2.00	0.173	1.77	4.326	1.81	34.208	1.84	108.542	1.88	161.026	1.85
P6	0.004	0.50	0.167	1.84	4.227	1.86	33.009	1.90	107.534	1.90	156.688	1.90
P7	0.002	1.00	0.167	1.84	4.176	1.88	32.771	1.92	106.119	1.92	156.433	1.90
P8	0.002	1.00	0.168	1.83	4.226	1.86	32.962	1.91	107.422	1.90	158.509	1.88

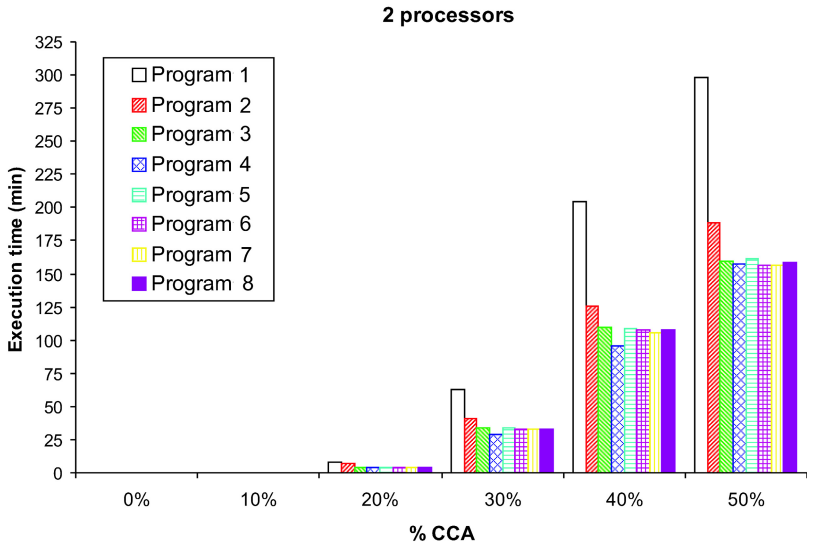


Fig. 3. Execution times in minutes for each program using 2 processors

6.1 Execution with 2 Processors

The computation times resulting from the execution of the eight programs with the selected penetration rates of CCA technology using 2 processors are gathered in Table 3 and illustrated in Figure 3.

Now we focus on the results associated to the 50% CCA penetration rate, since for this value we obtain the highest number of combinations, specifically for a chain of 20 vehicles we obtain a total of 184756 combinations. Therefore, it is for this particular penetration rate when we obtain a higher execution time and it can be considered as the critical case in terms of the solving time.

The sequential program (Program 1) lasts a total of 297.975 minutes, that is approximately 5 hours of computation. If we make a comparison among the parallelized programs we conclude that the best result is given by the Program 7, with a computation time of 156.433 minutes, what implies around 2.6 hours of calculation time. It is worth

Table 4. Execution times in minutes and speedup for each program using 4 processors

	0%		10%		20%		30%		40%		50%	
	Time	**SU**	**Time**	**SU**	**Time**	**SU**	**Time**	**SU**	**Time**	**SU**	**Time**	**SU**
P1	0.002	1.00	0.308	1.00	7.838	1.00	62.653	1.00	203.757	1.00	297.930	1.00
P2	0.001	2.00	0.199	1.55	5.053	1.55	30.676	2.04	94.173	2.16	135.907	2.19
P3	0.001	2.00	0.098	3.14	2.473	3.17	19.488	3.21	59.724	3.41	95.360	3.12
P4	0.004	0.50	0.078	3.95	1.998	3.92	16.072	3.90	51.830	3.93	86.175	3.45
P5	0.002	1.00	0.101	3.05	2.494	3.14	19.933	3.14	63.464	3.21	95.158	3.13
P6	0.005	0.40	0.091	3.38	2.251	3.48	18.013	3.48	59.810	3.40	89.064	3.34
P7	0.004	0.50	0.089	3.46	2.232	3.51	17.754	3.53	57.699	3.53	85.988	3.46
P8	0.003	0.67	0.090	3.42	2.245	3.49	17.926	3.49	59.453	3.43	88.422	3.37

mentioning that Program 7 is built by a combination of the parallelized tasks B and C, parallelizing the *for* loops that cover the range of average inter-vehicle distances and the number of combinations resulting from the technology penetration rate respectively. We obtain thus:

- Sequential time (P1): 297.975 minutes.
- Parallel time (P7): 156.433 minutes.

The achieved speedup (P1/P7) is 1.9, which implies an improvement of around 47.5% referred to the execution time.

6.2 Execution with 4 Processors

The computation times resulting from the execution of the eight programs with the selected penetration rates of CCA technology using 4 processors are presented in Table 4 and depicted in Figure 4.

When the CCA penetration rate equals the 50% we reach the highest computational load. So we also analyze the results with this penetration rate using 4 processors, focusing on the best and worst execution times achieved. The reference is still the sequential Program 1 with a duration of 297.93 minutes (around 5 hours). If we make a comparison among the parallelized programs we conclude that the best result is given again by the Program 7 with a calculation time of 85.988 minutes (around 1.43 hours). We obtain thus:

- Sequential time (P1): 297.93 minutes.
- Parallel time (P7): 85.988 minutes.

The achieved speedup is 3.46, which implies an improvement of around 71.1% referred to the execution time.

6.3 Execution with 8 Processors

The computation times resulting from the execution of the eight programs with the selected penetration rates of CCA technology using 8 processors are gathered in Table 5 and illustrated in Figure 5.

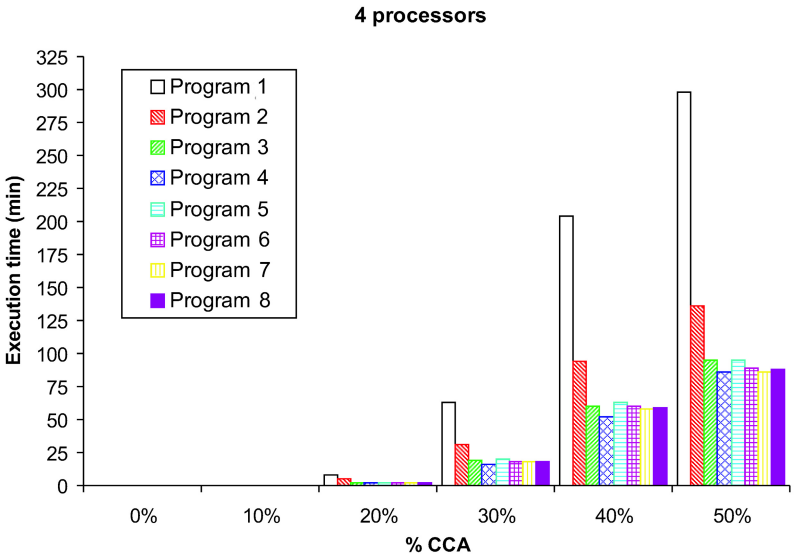


Fig. 4. Execution times in minutes for each program using 4 processors

Table 5. Execution times in minutes and speedup for each program using 8 processors

	0%		10%		20%		30%		40%		50%	
	Time	**SU**	**Time**	**SU**	**Time**	**SU**	**Time**	**SU**	**Time**	**SU**	**Time**	**SU**
P1	0.002	1.00	0.308	1.00	7.844	1.00	62.695	1.00	203.416	1.00	296.691	1.00
P2	0.003	0.67	0.193	1.59	4.610	1.70	26.578	2.36	78.644	2.58	117.415	2.53
P3	0.001	2.00	0.067	4.60	1.767	4.44	13.634	4.60	45.213	4.50	62.572	4.74
P4	0.008	0.25	0.047	6.55	1.155	6.79	9.310	6.73	32.142	6.33	54.165	5.48
P5	0.002	1.00	0.071	4.34	1.739	4.51	15.125	4.14	45.858	4.43	62.572	4.74
P6	0.005	0.40	0.055	5.60	1.258	6.23	10.158	6.17	35.275	5.76	54.006	5.49
P7	0.008	0.25	0.054	5.70	1.232	6.37	10.041	6.24	34.800	5.84	50.402	5.89
P8	0.007	0.28	0.051	6.04	1.248	6.28	10.143	6.18	35.376	5.75	53.031	5.59

Finally we analyze what happens if we use 8 processors to solve the problem. Once more, we obtain for the parallelized Program 7 the least computation time, 50.402 minutes with a 50% CCA penetration rate. So if we compare this result with the execution time of the sequential program we obtain an improvement of the 83%, that is, a speedup factor of 5.89.

6.4 Results Discussion

In conclusion, on the one hand, we have achieved an improvement of 83% in the computation time of the most complex case, what can be considered as a pretty much outstanding improvement. On the other hand, if we compare the best execution times between the two technical extremes under study, that is the use of 2 or 8 processors belonging to the shared nodes architecture in the Arabi cluster, we reach to an improvement of

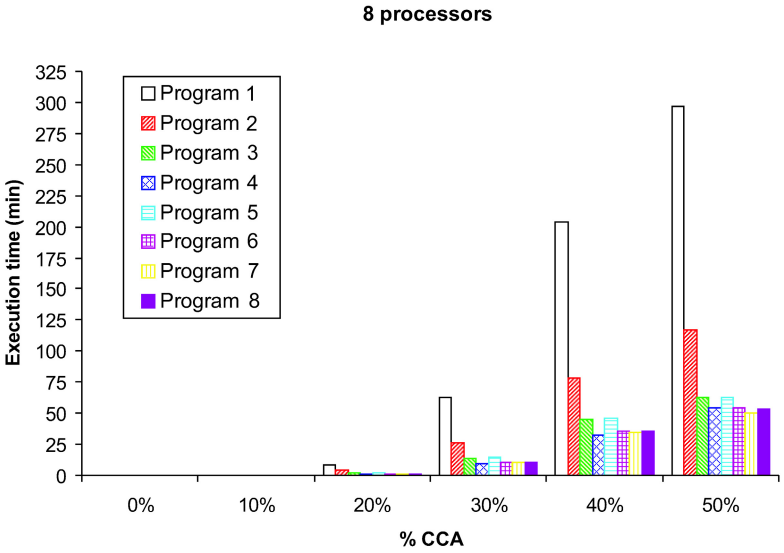


Fig. 5. Execution times in minutes for each program using 8 processors

67.78%, which implies an upwards trend with increasing the number of processors, as expected. Moreover, we can observe that those programs including the parallelization of task C, which implies an acceleration on the loop varying the CCA technology penetration rate, are the fastest ones. Nevertheless, the results obtained from Program 2 show that the improvement achieved parallelizing only the vector-matrix multiplication (task A) is already significant, reaching 60.4% using 8 processors.

Analyzing the speedup for programs 7 and 8 it surprises that P7, with two parallelized tasks, wins P8 including one more task. But this is a common fact in parallel computing due to load balancing and synchronization overhead [29]. This explains also that all programs including parallelized task C have similar execution times, since this is the heaviest computational task and outshines the improvement derived from the A and B tasks parallelization.

Let us compare now the obtained results for the Program 7, the one with the best execution times, centering on the 50% CCA penetration rate, since as we already mentioned, this is the heaviest option in terms of computational load. We find out an inverse relationship between computation time and the number of processors in use, since when we duplicate the number of processors the execution time of Program 7 is reduced almost to a half. Specifically, the speedup achieved passing from 2 to 4 processors is 1.82, and from 4 to 8 processors, 1.7. However, this speedup is limited according to Amdahl's law [30]. We have calculated for each program the theoretical speedup obtained from this law, as depicted in Figure 6.

Amdahl's law states that if α is the proportion of a program that can be made parallel then the maximum speedup, SU, that can be achieved by using n processors is:

$$SU = \frac{1}{(1-\alpha) + \frac{\alpha}{n}} \,. \tag{8}$$

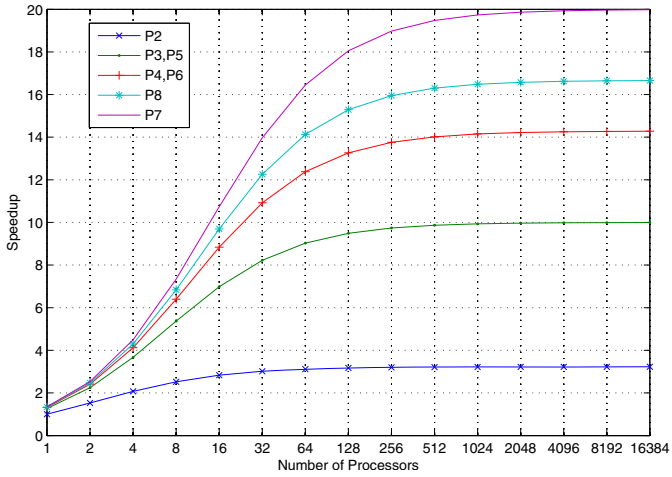


Fig. 6. Theoretical speedup limits calculated from Amdahl's law

We can estimate α by using the measured speedup SU on a specific number of processors sn as follows:

$$\alpha_{estimated} = \frac{\frac{1}{SU} - 1}{\frac{1}{sn} - 1} \,. \tag{9}$$

The results show that for Program 2 the speedup obtained with 8 processors is almost the limit for it, but the speedup for Program 7 can still grow up to 20, which implies reducing the execution time to less than 15 minutes.

Unfortunately, we have not been able to check how the results of Amdahl's law approach to reality. We tried to execute the Program 7 in the Superdome Ben, but executing it using 32 cores the time consumed was much higher than using 2 cores in a node of the cluster. It is owing to the computing speed (819 Gflops in the Superdome and 9.72 Tflops in the cluster).

As an alternative, we tried using MPI (Message Passing Interface Standard) [31] in order to execute our programs using different nodes of the cluster simultaneously. However, we encountered the problem of an excessive memory requirement, due to the need to replicate data across processes, and consequently we failed in the execution of the programs by this way too.

7 Conclusions and Outlook

Thanks to OpenMP parallelization techniques running under a supercomputing shared memory environment we succeded to evaluate the perfomance of a CCA application at different stages of technology deployment. To conclude, we were able to solve a program with a sequential execution time of 297.975 minutes in only 50.402 minutes.

Regarding the problems we have encountered, as future work, we aim to improve our analytical model, trying to reduce as possible the computational and memory costs.

We are also facing similar tasks to improve the efficiency of the VANET simulation environments we are using in order to validate our mathematical analyses.

Acknowledgements. This research has been supported by the MICINN/FEDER project grant TEC2010-21405-C02-02/TCM (CALM) and Fundación Séneca RM grant 00002/CS/08 FORMA. It is also developed in the framework of "Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, de la Fundación Séneca, Agencia de Ciencia y Tecnología de la RM". J. García-Haro acknowledges personal grant PR2009-0337 (MICINN/SEU/DGU Programa Nacional de Movilidad de Recursos Humanos de Investigación). J. B. Tomas-Gabarrón thanks the Spanish MICINN for a FPU (REF AP2008-02244) pre-doctoral fellowship. C. García-Costa acknowledges the Fundación Seneca for a FPI (REF 12347/FPI/09) pre-doctoral fellowship. We also want to express our gratitude to Rocío Murcia-Hernández for her programming assistance.

References

1. Tomas-Gabarron, J.B., Egea-Lopez, E., Garcia-Haro, J., Murcia-Hernandez, R.: Performance evaluation of a CCA application for VANETs using IEEE 802.11p. In: IEEE International Conference on Communications Workshops (ICC), pp. 1–5 (2010)
2. The official webpage of the Scientific Park of Murcia, http://www.cesmu.es
3. Murcia-Hernandez, R., Garcia-Costa, C., Tomas-Gabarronm, J.B., Egea-Lopez, E., Garcia-Haro, J.: Parallelization of a mathematical model to evaluate a CCA application for VANETs. In: 1st International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), The Netherlands (2011)
4. Barney, B.: Introduction to Parallel Computing. Lawrence Livermore National Laboratory, USA (2011), https://computing.llnl.gov/tutorials/parallel_comp/
5. Liu, S., Zhang, Y., Sun, X., Qiu, R.: Performance Evaluation of Multithreaded Sparse Matrix-Vector Multiplication using OpenMP. In: IEEE International Conference on High Performance Computing and Communications, pp. 659–665 (2009)
6. Kotakemori, H., Hasegawa, H., Kajiyama, T., Nukada, A., Suda, R., Nishida, A.: Performance Evaluation of Parallel Sparse Matrix–Vector Products on SGI Altix3700. In: Mueller, M.S., Chapman, B.M., de Supinski, B.R., Malony, A.D., Voss, M. (eds.) IWOMP 2005 and IWOMP 2006. LNCS, vol. 4315, pp. 153–163. Springer, Heidelberg (2008)
7. Goumas, G., Kourtis, K., Anastopoulos, N., Karakasis, V., Koziris, N.: Performance evaluation of the sparse matrix-vector multiplication on modern architectures. The Journal of Supercomputing 50(1), 36–77 (2009)
8. Williams, S., Olikera, L., Vuducc, R., Shalfa, J., Yelicka, K., Demmel, J.: Optimization of sparse matrix-vector multiplication on emerging multicore platforms. Parallel Computing 35(3), 178–194 (2009)
9. Ning, Z., Jung, Y., Jin, Y., Kim, K.: Route Optimization for GPSR in VANET. In: IEEE International Advance Computing Conference, pp. 569–573 (2009)
10. Wisitpongphan, N., Bai, F., Mudalige, P., Tonguz, O.K.: On the Routing Problem in Disconnected Vehicular Ad-hoc Networks. In: IEEE International Conference on Computer Communications, pp. 2291–2295 (2007)
11. Du, S., Liu, F.Q., Xu, S.Z., Wang, X.H.: On-demand power control algorithm based on vehicle ad hoc network. Jisuanji Gongcheng/ Computer Engineering 35(16), 97–100 (2009)
12. Fasolo, E., Zanella, A., Zorzi, M.: An effective broadcast scheme for alert message propagation in vehicular ad hoc networks. In: IEEE International Conference on Communications, pp. 3960–3965 (2006)

13. Li, L.J., Liu, H.F., Yang, Z.Y., Ge, L.J., Huang, X.Y.: Broadcasting methods in vehicular ad hoc networks. Ruan Jian Xue Bao (Journal of Software) 21(7), 1620–1634 (2010)
14. Harri, J., Filali, F., Bonnet, C.: Mobility models for vehicular ad hoc networks: a survey and taxonomy. IEEE Communications Surveys & Tutorials 11(4), 19–41 (2009)
15. Djenouri, D., Nekka, E., Soualhi, W.: Simulation of mobility models in vehicular ad hoc networks. In: Proceedings of First ICST International Conference on Ambient Media and Systems (2008)
16. Abboud, K., Weihua, Z.: Modeling and analysis for emergency messaging delay in vehicular ad hoc networks. In: IEEE Global Telecommunications Conference, pp. 1–6 (2009)
17. Fukuyama, J.: A delay time analysis for multi-hop V2V communications over a linear VANET. In: IEEE Vehicular Networking Conference, pp. 1–7 (2009)
18. Prasanth, K., Duraiswamy, K., Jayasudha, K., Chandrasekar, C.: Minimizing end-to-end delay in Vehicular Ad Hoc Network using Edge Node Based Greedy Routing. In: First International Conference on Advanced Computing, pp. 135–140 (2009)
19. Khabazian, M., Ali, M.: A performance modeling of connectivity in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology 57(4), 2440–2450 (2008)
20. Xie, B., Xiao, X.Q.: Survivability model for vehicular Ad-Hoc network based on Markov chain. Jisuanji Yingyong/ Journal of Computer Applications (2008)
21. Glimm, J., Fenton, R.E.: An Accident-Severity Analysis for a Uniform-Spacing Headway Policy. IEEE Transactions on Vehicular Technology 29(1), 96–103 (1980)
22. Touran, A., Brackstone, M.A., McDonald, M.: A collision model for safety evaluation of autonomous intelligent cruise control. Accident Analysis and Prevention 31(5), 567–578 (1999)
23. Kim, T., Jeong, H.Y.: Crash Probability and Error Rates for Head-On Collisions Based on Stochastic Analyses. IEEE Transactions on Intelligent Transportation Systems 11(4), 896–904 (2010)
24. Cook, W., Rich, J.L.: A parallel cutting-plane algorithm for the vehicle routing problem with time windows. Technical Report TR99-04, Department of Computational and Applied Mathematics, Rice University (1999)
25. Ghiani, G., Guerriero, F., Laporte, G., Musmanno, R.: Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. European Journal of Operational Research 151(1), 1–11 (2003)
26. Bouthillier, A.L., Crainic, T.G.: A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. Computers & Operations Research 32(7), 1685–1708 (2005)
27. Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J., Menon, R.: Parallel Programming in OpenMP. Morgan Kaufmann (2001)
28. Garcia-Costa, C., Egea-Lopez, E., Tomas-Gabarron, J.B., Garcia-Haro, J.: A stochastic model for chain collisions of vehicles equipped with vehicular communications. In: IEEE Transactions on Intelligent Transportation Systems (in press), doi:10.1109/TITS.2011.2171336
29. The OpenMP official webpage, http://openmp.org
30. Amdahl, G.M.: Validity of the single processor approach to achieving large scale computing capabilities. In: Proceedings of the Spring Joint Computer Conference (1967)
31. The MPI official webpage, http://www.mpi-forum.org/

The Stability Box for Minimizing Total Weighted Flow Time under Uncertain Data

Yuri N. Sotskov[1], Tsung-Chyan Lai[2], and Frank Werner[3]

[1] United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganova Str 6, Minsk, Belarus
[2] Department of Business Administration, National Taiwan University, Roosevelt Rd 85, Taipei, Taiwan
[3] Faculty of Mathematics, Otto-von-Guericke-University, Magdeburg, Germany
sotskov@newman.bas-net.by, tclai@ntu.edu.tw, frank.werner@ovgu.de

Abstract. We consider an uncertain single-machine scheduling problem, in which the processing time of a job can take any real value from a given closed interval. The criterion is to minimize the sum of weighted completion times of the n jobs, a weight being associated with each job. For a job permutation, we study the stability box, which is a subset of the stability region. We derive an $O(n \log n)$ algorithm for constructing a job permutation with the largest dimension and volume of a stability box. The efficiency of such a permutation is demonstrated via a simulation on a set of randomly generated instances with $1000 \leq n \leq 2000$. If several permutations have the largest dimension and volume of a stability box, the developed algorithm selects one of them due to a mid-point heuristic.

Keywords: Single-machine scheduling, Uncertain data, Total weighted flow time, Stability analysis.

1 Introduction

In real-life scheduling, the numerical data are usually uncertain. A stochastic [6] or a fuzzy method [8] are used when the job processing times may be defined as random variables or as fuzzy numbers. If these times may be defined neither as random variables with known probability distributions nor as fuzzy numbers, other methods are needed to solve a scheduling problem under uncertainty [1,7,13]. The robust method [1,2,3] assumes that the decision-maker prefers a schedule hedging against the worst-case scenario. The stability method [4,5,10,11,12,13] combines a stability analysis, a multi-stage decision framework and the solution concept of a minimal dominant set of semi-active schedules.

In this paper, we implement the stability method for a single-machine problem with interval processing times of the n jobs (Section 2). In Section 3, we derive an $O(n \log n)$ algorithm for constructing a job permutation with the largest dimension and volume of a stability box. Computational results are presented in Section 4. We conclude with Section 5.

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 39–55.
DOI: 10.1007/978-3-642-34336-0_3

2 Problem Setting

The jobs $\mathcal{J} = \{J_1, J_2, ..., J_n\}$, $n \geq 2$, have to be processed on a single machine, a positive weight w_i being given for any job $J_i \in \mathcal{J}$. The processing time p_i of a job J_i can take any real value from a given segment $[p_i^L, p_i^U]$, where $0 \leq p_i^L \leq p_i^U$. The exact value $p_i \in [p_i^L, p_i^U]$ may remain unknown until the completion of the job $J_i \in \mathcal{J}$. Let $T = \{p \in R_+^n \mid p_i^L \leq p_i \leq p_i^U,\ i \in \{1, 2, \ldots, n\}\}$ denote the set of vectors $p = (p_1, p_2, \ldots, p_n)$ (scenarios) of the possible job processing times. $S = \{\pi_1, \pi_2, \ldots, \pi_{n!}\}$ denotes the set of permutations $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n})$ of the jobs $\mathcal{J}$. Problem $1|p_i^L \leq p_i \leq p_i^U| \sum w_i C_i$ is to find an optimal permutation $\pi_t \in S$:

$$\sum_{J_i \in \mathcal{J}} w_i C_i(\pi_t, p) = \gamma_p^t = \min_{\pi_k \in S} \left\{ \sum_{J_i \in \mathcal{J}} w_i C_i(\pi_k, p) \right\}. \tag{1}$$

Hereafter, $C_i(\pi_k, p) = C_i$ is the completion time of job $J_i \in \mathcal{J}$ in a semi-active schedule [6,13] defined by the permutation π_k.

Since a factual scenario $p \in T$ is unknown before scheduling, the completion time C_i of a job $J_i \in \mathcal{J}$ can be determined after the schedule execution. Therefore, one cannot calculate the value γ_p^k of the objective function

$$\gamma = \sum_{J_i \in \mathcal{J}} w_i C_i(\pi_k, p)$$

for a permutation $\pi_k \in S$ before the schedule realization.

However, one must somehow define a schedule before to realize it. So, the problem $1|p_i^L \leq p_i \leq p_i^U| \sum w_i C_i$ of finding an optimal permutation $\pi_t \in S$ defined in (1) is not correct. In general, one can find only a heuristic solution (a job permutation) to problem $1|p_i^L \leq p_i \leq p_i^U| \sum w_i C_i$ the efficiency of which may be estimated either analytically or via a simulation.

In the deterministic case, when a scenario $p \in T$ is fixed before scheduling (i.e., equalities $p_i^L = p_i^U = p_i$ hold for each job $J_i \in \mathcal{J}$), problem $1|p_i^L \leq p_i \leq p_i^U| \sum w_i C_i$ reduces to the classical problem $1||\sum w_i C_i$. In contrast to the uncertain problem $1|p_i^L \leq p_i \leq p_i^U| \sum w_i C_i$, problem $1||\sum w_i C_i$ is called deterministic. The deterministic problem $1||\sum w_i C_i$ is correct and can be solved exactly in $O(n \log n)$ time [9] due to the necessary and sufficient condition (2) for the optimality of a permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$:

$$\frac{w_{k_1}}{p_{k_1}} \geq \frac{w_{k_2}}{p_{k_2}} \geq \ldots \geq \frac{w_{k_n}}{p_{k_n}}, \tag{2}$$

where the strict inequality $p_{k_i} > 0$ holds for each job $J_{k_i} \in \mathcal{J}$. Using the sufficiency of condition (2), problem $1||\sum w_i C_i$ can be solved to optimality by the weighted shortest processing time rule: process the jobs $\mathcal{J}$ in non-increasing order of their weight-to-process ratios $\frac{w_{k_i}}{p_{k_i}}$, $J_{k_i} \in \mathcal{J}$.

3 The Stability Box

In [12], the stability box $\mathcal{SB}(\pi_k, T)$ within a set of scenarios T has been defined for a permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$. To present the definition of the stability box $\mathcal{SB}(\pi_k, T)$, we need the following notations.

We denote $\mathcal{J}(k_i) = \{J_{k_1}, J_{k_2}, \ldots, J_{k_{i-1}}\}$ and $\mathcal{J}[k_i] = \{J_{k_{i+1}}, J_{k_{i+2}}, \ldots, J_{k_n}\}$. Let S_{k_i} denote the set of permutations $(\pi(\mathcal{J}(k_i)), J_{k_i}, \pi(\mathcal{J}[k_i])) \in S$ of the jobs $\mathcal{J}$, $\pi(\mathcal{J}')$ being a permutation of the jobs $\mathcal{J}' \subset \mathcal{J}$. Let N_k denote a subset of set $N = \{1, 2, \ldots, n\}$. The notation $1|p|\sum w_iC_i$ will be used for indicating an instance with a fixed scenario $p \in T$ of the deterministic problem $1||\sum w_iC_i$.

Definition 1. [12] *The maximal closed rectangular box*

$$\mathcal{SB}(\pi_k, T) = \times_{k_i \in N_k}[l_{k_i}, u_{k_i}] \subseteq T$$

is a stability box of permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$, if permutation $\pi_e = (J_{e_1}, J_{e_2}, \ldots, J_{e_n}) \in S_{k_i}$ being optimal for the instance $1|p|\sum w_iC_i$ with a scenario $p = (p_1, p_2, \ldots, p_n) \in T$ remains optimal for the instance $1|p'|\sum w_iC_i$ with a scenario $p' \in \{\times_{j=1}^{i-1}[p_{k_j}, p_{k_j}]\} \times [l_{k_i}, u_{k_i}] \times \{\times_{j=i+1}^{n}[p_{k_j}, p_{k_j}]\}$ for each $k_i \in N_k$. If there does not exist a scenario $p \in T$ such that permutation π_k is optimal for the instance $1|p|\sum w_iC_i$, then $\mathcal{SB}(\pi_k, T) = \emptyset$.

The maximality of the closed rectangular box $\mathcal{SB}(\pi_k, T) = \times_{k_i \in N_k}[l_{k_i}, u_{k_i}]$ in Definition 1 means that the box $\mathcal{SB}(\pi_k, T) \subseteq T$ has both a maximal possible dimension $|N_k|$ and a maximal possible volume.

For any scheduling instance, the stability box is a subset of the stability region [13,14]. However, we substitute the stability region by the stability box, since the latter is easy to compute [11,12]. In [11], a branch-and-bound algorithm has been developed to select a permutation in the set S with the largest volume of a stability box. If several permutations have the same volume of the stability box, the algorithm from [11] selects one of them due to simple heuristics. The efficiency of the constructed permutations has been demonstrated on a set of randomly generated instances with $5 \leq n \leq 100$.

In [12], an $O(n \log n)$ algorithm has been developed for calculating a stability box $\mathcal{SB}(\pi_k, T)$ for the fixed permutation $\pi_k \in S$ and an $O(n^2)$ algorithm has been developed for selecting a permutation in the set S with the largest dimension and volume of a stability box. The efficiency of these algorithms was demonstrated on a set of randomly generated instances with $10 \leq n \leq 1000$. All algorithms developed in [11,12] use the precedence-dominance relation on the set of jobs $\mathcal{J}$ and the solution concept of a minimal dominant set $S(T) \subseteq S$ defined as follows.

Definition 2. [10] *The set of permutations $S(T) \subseteq S$ is a minimal dominant set for a problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_iC_i$, if for any fixed scenario $p \in T$, the set $S(T)$ contains at least one optimal permutation for the instance $1|p|\sum w_iC_i$, provided that any proper subset of set $S(T)$ loses such a property.*

Definition 3. [10] *Job J_u dominates job J_v, if there exists a minimal dominant set $S(T)$ for the problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_iC_i$ such that job J_u precedes job J_v in every permutation of the set $S(T)$.*

Theorem 1. [10] *For the problem* $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$, *job* J_u *dominates job* J_v *if and only if inequality* (3) *holds:*

$$\frac{w_u}{p_u^U} \ge \frac{w_v}{p_v^L}. \tag{3}$$

Due to Theorem 1 proven in [10], we can obtain a compact presentation of a minimal dominant set $S(T)$ in the form of a digraph $(\mathcal{J}, \mathcal{A})$ with the vertex set $\mathcal{J}$ and the arc set $\mathcal{A}$. To this end, we can check inequality (3) for each pair of jobs from the set $\mathcal{J}$ and construct a dominance digraph $(\mathcal{J}, \mathcal{A})$ of the precedence-dominance relation on the set of jobs $\mathcal{J}$ as follows. The arc (J_u, J_v) belongs to the set $\mathcal{A}$ if and only if inequality (3) holds. The construction of the digraph $(\mathcal{J}, \mathcal{A})$ takes $O(n^2)$ time.

3.1 Illustrative Example

For the sake of simplicity of the calculation, we consider a special case $1|p_i^L \le p_i \le p_i^U|\sum C_i$ of the problem $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ when each job $J_i \in \mathcal{J}$ has a weight w_i equal to one. From condition (2), it follows that the deterministic problem $1||\sum C_i$ can be solved to optimality by the shortest processing time rule: process the jobs in non-decreasing order of their processing times p_{k_i}, $J_{k_i} \in \mathcal{J}$.

A set of scenarios T for Example 1 of the uncertain problem $1|p_i^L \le p_i \le p_i^U|\sum C_i$ is defined in columns 1 and 2 in Table 1.

Table 1. Data for calculating $\mathcal{SB}(\pi_1, T)$ for Example 1

	1	2	3	4	5	6	7	8
i	p_i^L	p_i^U	$\frac{w_i}{p_i^U}$	$\frac{w_i}{p_i^L}$	d_i^-	d_i^+	$\frac{w_i}{d_i^+}$	$\frac{w_i}{d_i^-}$
1	2	3	$\frac{1}{3}$	0.5	1	0.5	2	1
2	1	9	$\frac{1}{9}$	1	$\frac{1}{6}$	$\frac{1}{3}$	3	6
3	8	8	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{1}{9}$	9	6
4	6	10	0.1	$\frac{1}{6}$	0.1	$\frac{1}{9}$	9	10
5	11	12	$\frac{1}{12}$	$\frac{1}{11}$	0.1	$\frac{1}{11}$	11	10
6	10	19	$\frac{1}{19}$	0.1	$\frac{1}{15}$	$\frac{1}{12}$	12	15
7	17	19	$\frac{1}{19}$	$\frac{1}{17}$	$\frac{1}{15}$	$\frac{1}{19}$	19	15
8	15	20	$\frac{1}{20}$	$\frac{1}{15}$	$\frac{1}{20}$	$\frac{1}{19}$	19	20

In [12], formula (9) has been proven. To use it for calculating the stability box $\mathcal{SB}(\pi_k, T)$, one has to define for each job $J_{k_i} \in \mathcal{J}$ the maximal range $[l_{k_i}, u_{k_i}]$ of possible variations of the processing time p_{k_i} preserving the optimality of permutation π_k (see Definition 1). Due to the additivity of the objective function $\gamma = \sum_{J_i \in \mathcal{J}} w_iC_i(\pi_k, p)$, the lower bound $d_{k_i}^-$ on the maximal range of possible variations of the weight-to-process ratio $\frac{w_{k_i}}{p_{k_i}}$ preserving the optimality of the permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ is calculated as follows:

$$d^-_{k_i} = \max\left\{\frac{w_{k_i}}{p^U_{k_i}},\ \max_{i<j\le n}\left\{\frac{w_{k_j}}{p^L_{k_j}}\right\}\right\}, i \in \{1,2,\ldots,n-1\}, \tag{4}$$

$$d^-_{k_n} = \frac{w_{k_n}}{p^U_{k_n}}. \tag{5}$$

The upper bound $d^+_{k_i}, J_{k_i} \in \mathcal{J}$, on the maximal range of possible variations of the weight-to-process ratio $\frac{w_{k_i}}{p_{k_i}}$ preserving the optimality of the permutation π_k is calculated as follows:

$$d^+_{k_i} = \min\left\{\frac{w_{k_i}}{p^L_{k_i}},\ \min_{1\le j<i}\left\{\frac{w_{k_j}}{p^U_{k_j}}\right\}\right\}, i \in \{2,3,\ldots,n\}, \tag{6}$$

$$d^+_{k_1} = \frac{w_{k_1}}{p^L_{k_1}}. \tag{7}$$

For Example 1, the values $d^-_{k_i}, i \in \{1,2,\ldots,8\}$, defined in (4) and (5) are given in column 5 of Table 1. The values $d^+_{k_i}$ defined in (6) and (7) are given in column 6. In [12], the following claim has been proven.

Theorem 2. [12] *If there is no job J_{k_i}, $i \in \{1,2,\ldots,n-1\}$, in permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ such that inequality*

$$\frac{w_{k_i}}{p^L_{k_i}} < \frac{w_{k_j}}{p^U_{k_j}} \tag{8}$$

holds for at least one job J_{k_j}, $j \in \{i+1, i+2, \ldots, n\}$, then the stability box $\mathcal{SB}(\pi_k, T)$ is calculated as follows:

$$\mathcal{SB}(\pi_k, T) = \times_{d^-_{k_i} \le d^+_{k_i}} \left[\frac{w_{k_i}}{d^+_{k_i}}, \frac{w_{k_i}}{d^-_{k_i}}\right]. \tag{9}$$

Otherwise, $\mathcal{SB}(\pi_k, T) = \emptyset$.

Using Theorem 2, we can calculate the stability box $\mathcal{SB}(\pi_1, T)$ of the permutation $\pi_1 = (J_1, J_2, \ldots, J_8)$ in Example 1. First, we convince that there is no job J_{k_i}, $i \in \{1,2,\ldots,n-1\}$, with inequality (8). Due to Theorem 2, $\mathcal{SB}(\pi_1, T) \neq \emptyset$.

The bounds $\frac{w_{k_i}}{d^+_{k_i}}$ and $\frac{w_{k_i}}{d^-_{k_i}}$ on the maximal possible variations of the processing times p_{k_i} preserving the optimality of the permutation π_1 are given in columns 7 and 8 of Table 1. The maximal ranges (segments) of possible variations of the job processing times within the stability box $\mathcal{SB}(\pi_1, T)$ are dashed in a coordinate system in Fig. 1, where the abscissa axis is used for indicating the job processing times and the ordinate axis for the jobs from set $\mathcal{J}$.

Using formula (9), we obtain the stability box for permutation π_1 as follows:

$$\begin{aligned}\mathcal{SB}(\pi_1, T) &= \left[\frac{w_2}{d^+_2}, \frac{w_2}{d^-_2}\right] \times \left[\frac{w_4}{d^+_4}, \frac{w_4}{d^-_4}\right] \times \left[\frac{w_6}{d^+_6}, \frac{w_6}{d^-_6}\right] \times \left[\frac{w_8}{d^+_8}, \frac{w_8}{d^-_8}\right] \\ &= [3,6] \times [9,10] \times [12,15] \times [19,20].\end{aligned}$$

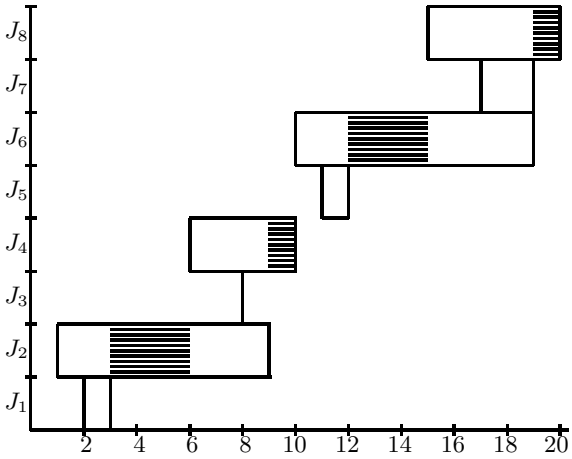


Fig. 1. The maximal ranges $[l_i, u_i]$ of possible variations of the processing times $p_i, i \in \{2, 4, 6, 8\}$, within the stability box $\mathcal{SB}(\pi_1, T)$ are dashed

Each job J_i, $i \in \{1, 3, 5, 7\}$, has an empty range of possible variations of the time p_i preserving the optimality of permutation π_1 since $d_i^- > d_i^+$ (see columns 5 and 6 in Table 1). The dimension of the stability box $\mathcal{SB}(\pi_1, T)$ is equal to $4 = 8 - 4$. The volume of this stability box is equal to $9 = 3 \cdot 1 \cdot 3 \cdot 1$.

In [12], an $O(n \log n)$ algorithm STABOX has been developed for calculating the stability box $\mathcal{SB}(\pi_k, T)$ for a fixed permutation $\pi_k \in S$.

For practice, the value of the relative volume of a stability box is more useful than its absolute value. The relative volume of a stability box is defined as the product of the fractions

$$\left(\frac{w_i}{d_i^-} - \frac{w_i}{d_i^+}\right) : \left(p_i^U - p_i^L\right) \tag{10}$$

for the jobs $J_i \in \mathcal{J}$ having non-empty ranges $[l_i, u_i]$ of possible variations of the processing time p_i (inequality $d_i^- \leq d_i^+$ must hold for such a job $J_i \in \mathcal{J}$).

The relative volume of the stability box for permutation π_1 in Example 1 is calculated as follows: $\frac{3}{8} \cdot \frac{1}{4} \cdot \frac{3}{9} \cdot \frac{1}{5} = \frac{1}{160}$. The absolute volume of the whole box of the scenarios T is equal to $2880 = 1 \cdot 8 \cdot 4 \cdot 1 \cdot 9 \cdot 2 \cdot 5$. The relative volume of the rectangular box T is defined as 1.

3.2 Properties of a Stability Box

A job permutation in the set S with a larger dimension and a larger volume of the stability box seems to be more efficient than one with a smaller dimension and (or) a smaller volume of stability box.

We investigate properties of a stability box, which allow us to derive an $O(n \log n)$ algorithm for choosing a permutation $\pi_t \in S$ which has

(a) the largest dimension $|N_t|$ of the stability box $\mathcal{SB}(\pi_t, T) = \times_{t_i \in N_t}[l_{t_i}, u_{t_i}] \subseteq T$ among all permutations $\pi_k \in S$ and

(b) the largest volume of the stability box $\mathcal{SB}(\pi_t, T)$ among all permutations $\pi_k \in S$ having the largest dimension $|N_k| = |N_t|$ of their stability boxes $\mathcal{SB}(\pi_k, T)$.

Definition 1 implies the following claim.

Property 1. For any jobs $J_i \in \mathcal{J}$ and $J_v \in \mathcal{J}$, $v \neq i$,

$$(w_i/u_i, w_i/l_i) \bigcap \left[w_v/p_v^U, w_v/p_v^L\right] = \varnothing.$$

Let S^{max} denote the subset of all permutations π_t in the set S possessing properties (a) and (b). Using Property 1, we shall show how to define the relative order of a job $J_i \in \mathcal{J}$ with respect to a job $J_v \in \mathcal{J}$ for any $v \neq i$ in a permutation $\pi_t = (J_{t_1}, J_{t_2}, \ldots, J_{t_n}) \in S^{max}$. To this end, we have to treat all three possible cases (I)–(III) for the intersection of the open interval $\left(\frac{w_i}{p_i^U}, \frac{w_i}{p_i^L}\right)$ and the closed interval $\left[\frac{w_v}{p_v^U}, \frac{w_v}{p_v^L}\right]$. The order of the jobs J_i and J_v in the desired permutation $\pi_t \in S^{max}$ may be defined in the cases (I)–(III) using the following rules.

Case (I) is defined by the inequalities

$$\frac{w_v}{p_v^U} \leq \frac{w_i}{p_i^U}, \quad \frac{w_v}{p_v^L} \leq \frac{w_i}{p_i^L} \tag{11}$$

provided that at least one of inequalities (11) is strict.

In case (I), the desired order of the jobs J_v and J_i in permutation $\pi_t \in S^{max}$ may be defined by a strict inequality from (11): job J_v proceeds job J_i in permutation π_t.

Indeed, if job J_i proceeds job J_v, then the maximal ranges $[l_i, u_i]$ and $[l_v, u_v]$ of possible variations of the processing times p_i and p_v preserving the optimality of $\pi_k \in S$ are both empty (it follows from equalities (4) – (7) and (9)). Thus, the following property is proven.

Property 2. For case (I), there exists a permutation $\pi_t \in S^{max}$, in which job J_v proceeds job J_i.

Case (II) is defined by the equalities

$$\frac{w_v}{p_v^U} = \frac{w_i}{p_i^U}, \quad \frac{w_v}{p_v^L} = \frac{w_i}{p_i^L}. \tag{12}$$

Property 3. For case (II), there exists a permutation $\pi_t \in S^{max}$, in which the jobs J_i and J_v are located adjacently: $i = t_r$ and $v = t_{r+1}$.

Proof. The maximal ranges $[l_i, u_i]$ and $[l_v, u_v]$ of possible variations of the processing times p_i and p_v preserving the optimality of $\pi_k \in S$ are both empty.

If job J_i and job J_v are located adjacently, then the maximal range $[l_u, u_u]$ of possible variations of the processing time p_u for any job $J_u \in \mathcal{J} \setminus \{J_i, J_v\}$ preserving the optimality of the permutation π_k is no less than that if at least one job $J_w \in \mathcal{J} \setminus \{J_i, J_v\}$ is located between job J_i and job J_v. ■

If equalities (12) hold, one can restrict the search for a permutation $\pi_t \in S^{max}$ by a subset of permutations in set S with the adjacently located jobs J_i and J_v (Property 3). Moreover, the order of such jobs $\{J_i, J_v\}$ does not influence the volume of the stability box and its dimension.

Remark 1. Due to Property 3, while looking for a permutation $\pi_t \in S^{max}$, we shall treat a pair of jobs $\{J_i, J_v\}$ satisfying (12) as one job (either job J_i or J_v).

Case (III) is defined by the strict inequalities

$$\frac{w_v}{p_v^U} > \frac{w_i}{p_i^U}, \quad \frac{w_v}{p_v^L} < \frac{w_i}{p_i^L}. \tag{13}$$

For a job $J_i \in \mathcal{J}$ satisfying case (III), let $\mathcal{J}(i)$ denote the set of all jobs $J_v \in \mathcal{J}$, for which the strict inequalities (13) hold.

Property 4. (i) For a fixed permutation $\pi_k \in S$, job $J_i \in \mathcal{J}$ may have at most one maximal segment $[l_i, u_i]$ of possible variations of the processing time $p_i \in [p_i^L, p_i^U]$ preserving the optimality of permutation π_k.

(ii) For the whole set of permutations S, only in case (III), a job $J_i \in \mathcal{J}$ may have more than one (namely: $|\mathcal{J}(i)|+1 > 1$) maximal segments $[l_i, u_i]$ of possible variations of the time $p_i \in [p_i^L, p_i^U]$ preserving the optimality of this or that particular permutation from the set S.

Proof. Part *(i)* of Property 4 follows from the fact that a non-empty maximal segment $[l_i, u_i]$ (if any) is uniquely determined by the subset $\mathcal{J}^-(i)$ of jobs located before job J_i in permutation π_k and the subset $\mathcal{J}^+(i)$ of jobs located after job J_i. The subsets $\mathcal{J}^-(i)$ and $\mathcal{J}^+(i)$ are uniquely determined for a fixed permutation $\pi_k \in S$ and a fixed job $J_i \in \mathcal{J}$.

Part *(ii)* of Property 4 follows from the following observations. If the open interval $\left(\frac{w_i}{p_i^U}, \frac{w_i}{p_i^L}\right)$ does not intersect with the closed interval $\left[\frac{w_v}{p_v^U}, \frac{w_v}{p_v^L}\right]$ for each job $J_v \in \mathcal{J}$, then there exists a permutation $\pi_t \in S^{max}$ with a maximal segment $[l_i, u_i] = \left[w_i/p_i^U, w_i/p_i^L\right]$ preserving the optimality of permutation π_t.

Each job $J_v \in \mathcal{J}$ with a non-empty intersection $\left(\frac{w_i}{p_i^U}, \frac{w_i}{p_i^L}\right) \bigcap \left[\frac{w_v}{p_v^U}, \frac{w_v}{p_v^L}\right] \neq \emptyset$ satisfying inequalities (11) (case (I)) or equalities (12) (case (II)) may shorten the above maximal segment $[l_i, u_i]$ and cannot generate a new possible maximal segment. In case (III), a job J_v satisfying inequalities (13) may generate a new possible maximal segment $[l_i, u_i]$ just for job J_i satisfying the same strict inequalities (13) as job J_v does. So, the cardinality $|\mathcal{L}(i)|$ of the whole set $\mathcal{L}(i)$ of such segments $[l_i, u_i]$ is not greater than $|\mathcal{J}(i)| + 1$. ■

Let $\mathcal{L}$ denote the set of all maximal segments $[l_i, u_i]$ of possible variations of the processing times p_i for all jobs $J_i \in \mathcal{J}$ preserving the optimality of a permutation $\pi_t \in S^{max}$. Using Property 4 and induction on the cardinality $|\mathcal{J}(i)|$, we proved

Property 5. $|\mathcal{L}| \leq n$.

3.3 A Job Permutation with the Largest Volume of a Stability Box

The above properties allows us to derive an $O(n \log n)$ algorithm for calculating a permutation $\pi_t \in S^{max}$ with the largest dimension $|N_t|$ and the largest volume of a stability box $\mathcal{SB}(\pi_t, T)$.

Algorithm. MAX-STABOX

Input: Segments $[p_i^L, p_i^U]$, weights w_i, $J_i \in \mathcal{J}$.
Output: Permutation $\pi_t \in S^{max}$, stability box $\mathcal{SB}(\pi_t, T)$.

Step 1: Construct the list $\mathcal{M}(U) = (J_{u_1}, J_{u_2}, \ldots, J_{u_n})$ and the list $\mathcal{W}(U) = (\frac{w_{u_1}}{p^U_{u_1}}, \frac{w_{u_2}}{p^U_{u_2}}, \ldots, \frac{w_{u_n}}{p^U_{u_n}})$ in non-increasing order of $\frac{w_{u_r}}{p^U_{u_r}}$. Ties are broken via decreasing $\frac{w_{u_r}}{p^L_{u_r}}$.

Step 2: Construct the list $\mathcal{M}(L) = (J_{l_1}, J_{l_2}, \ldots, J_{l_n})$ and the list $\mathcal{W}(L) = (\frac{w_{l_1}}{p^L_{l_1}}, \frac{w_{l_2}}{p^L_{l_2}}, \ldots, \frac{w_{l_n}}{p^L_{l_n}})$ in non-increasing order of $\frac{w_{l_r}}{p^L_{l_r}}$. Ties are broken via decreasing $\frac{w_{l_r}}{p^U_{l_r}}$.

Step 3: FOR $j = 1$ to $j = n$ DO
compare job J_{u_j} and job J_{l_j}.

Step 4: IF $J_{u_j} = J_{l_j}$ THEN job J_{u_j} has to be located in position j in permutation $\pi_t \in S^{max}$ GOTO step 8.

Step 5: ELSE job $J_{u_j} = J_i$ satisfies inequalities (13). Construct the set $\mathcal{J}(i) = \{J_{u_{r+1}}, J_{u_{r+2}}, \ldots, J_{l_{k+1}}\}$ of all jobs J_v satisfying inequalities (13), where $J_i = J_{u_j} = J_{l_k}$.

Step 6: Choose the largest range $[l_{u_j}, u_{u_j}]$ among those generated for the job $J_{u_j} = J_i$.

Step 7: Partition the set $\mathcal{J}(i)$ into the subsets $\mathcal{J}^-(i)$ and $\mathcal{J}^+(i)$ generating the largest range $[l_{u_j}, u_{u_j}]$. Set $j = k + 1$ GOTO step 4.

Step 8: Set $j := j + 1$ GOTO step 4.
END FOR

Step 9: Construct the permutation $\pi_t \in S^{max}$ via putting the jobs $\mathcal{J}$ in the positions defined in steps 3 – 8.

Step 10: Construct the stability box $\mathcal{SB}(\pi_t, T)$ using algorithm STABOX derived in [12]. STOP.

Steps 1 and 2 of algorithm MAX-STABOX are based on Property 3 and Remark 1. Step 4 is based on Property 2. Steps 5 – 7 are based on Property 4, part *(ii)*. Step 9 is based on Property 6 which follows.

To prove Property 6, we have to analyze algorithm MAX-STABOX. In steps 1, 2 and 4, all jobs $\mathcal{J}^t = \{J_i \mid J_{u_j} = J_i = J_{l_j}\}$ having the same position in both lists $\mathcal{M}(U)$ and $\mathcal{M}(L)$ obtain fixed positions in the permutation $\pi_t \in S^{max}$.

The positions of the remaining jobs $\mathcal{J} \setminus \mathcal{J}^t$ in the permutation π_t are determined in steps 5 – 7. The fixed order of the jobs $\mathcal{J}^t$ may shorten the original segment $[p_i^L, p_i^U]$ of a job $J_i \in \mathcal{J} \setminus \mathcal{J}^t$. We denote such a reduced segment as $[\widehat{p}_i^L, \widehat{p}_i^U]$. So, in steps 5 – 7, the reduced segment $[\widehat{p}_i^L, \widehat{p}_i^U]$ has to be considered instead of original segment $[p_i^L, p_i^U]$ for a job $J_i \in \mathcal{J} \setminus \mathcal{J}^t$.

Let $\mathcal{L}'$ denote the maximal subset of set $\mathcal{L}$ (see Property 5) including exactly one element from each set $\mathcal{L}(i)$, for which job $J_i \in \mathcal{J}$ satisfies the strict inequalities (13).

Property 6. There exists a permutation $\pi_t \in S$ with the set $\mathcal{L}' \subseteq \mathcal{L}$ of maximal segments $[l_i, u_i]$ of possible variations of the processing time $p_i, J_i \in \mathcal{J}$, preserving the optimality of the permutation π_t.

Proof. Due to Property 2 and steps 1 – 4 of algorithm MAX-STABOX, the maximal segments $[l_i, u_i]$ and $[l_v, u_v]$ (if any) of jobs J_i and J_v satisfying (11) preserve the optimality of the permutation $\pi_t \in S^{max}$.

Let $\mathcal{J}^*$ denote the set of all jobs J_i satisfying (13). It is easy to see that

$$\bigcap_{J_i \in \mathcal{J}} (\widehat{p}_i^L, \widehat{p}_i^U] = \varnothing.$$

Therefore,

$$\bigcap_{J_i \in \mathcal{J}} \mathcal{J}(i) = \varnothing.$$

Hence, step 9 is correct: putting the set of jobs $\mathcal{J}$ in the positions defined in steps 3 – 8 does not cause any contradiction of the job orders. ■

Obviously, steps 1 and 2 take $O(n \log n)$ time. Due to Properties 4 and 5, steps 6, 7 and 9 take $O(n)$ time. Step 10 takes $O(n \log n)$ time since algorithm STABOX derived in [12] has the same complexity. Thus, the whole algorithm MAX-STABOX takes $O(n \log n)$ time. It is easy to convince that, due to steps 1 – 5, the permutation π_t constructed by algorithm MAX-STABOX possesses property (a) and, due to steps 6, 7 and 9, this permutation possesses property (b).

Remark 2. Algorithm MAX-STABOX constructs a permutation $\pi_t \in S$ such that the dimension $|N_t|$ of the stability box $\mathcal{SB}(\pi_t, T) = \times_{t_i \in N_t}[l_{t_i}, u_{t_i}] \subseteq T$ is the largest one for all permutations S, and the volume of the stability box $\mathcal{SB}(\pi_t, T)$ is the largest one for all permutations $\pi_k \in S$ having the largest dimension $|N_k| = |N_t|$ of their stability boxes $\mathcal{SB}(\pi_k, T)$.

Returning to Example 1, one can show (using Algorithm MAX-STABOX) that permutation $\pi_1 = (J_1, J_2, \ldots, J_8)$ has the largest dimension and volume of a stability box. Next, we compare $\mathcal{SB}(\pi_1, T)$ with the stability boxes calculated for the permutations obtained by the three heuristics defined as follows.

The lower-point heuristic generates an optimal permutation $\pi_l \in S$ for the instance $1|p^L|\sum w_i C_i$ with

$$p^L = (p_1^L, p_2^L, \ldots, p_n^L) \in T. \tag{14}$$

The upper-point heuristic generates an optimal permutation $\pi_u \in S$ for the instance $1|p^U|\sum w_i C_i$ with

$$p^U = (p_1^U, p_2^U, \ldots, p_n^U) \in T. \tag{15}$$

The mid-point heuristic generates an optimal permutation $\pi_m \in S$ for the instance $1|p^M|\sum w_i C_i$ with

$$p^M = \left(\frac{p_1^U - p_1^L}{2}, \frac{p_2^U - p_2^L}{2}, \ldots, \frac{p_n^U - p_n^L}{2}\right) \in T. \tag{16}$$

We obtain the permutation $\pi_l = (J_2, J_1, J_4, J_3, J_6, J_5, J_8, J_7)$ with the stability box

$$\mathcal{SB}(\pi_l, T) = \left[\frac{w_2}{d_2^+}, \frac{w_2}{d_2^-}\right] \times \left[\frac{w_6}{d_6^+}, \frac{w_6}{d_6^-}\right] = [1, 2] \times [10, 11].$$

The volume of the stability box $\mathcal{SB}(\pi_l, T)$ is equal to 1. We obtain the permutation $\pi_u = (J_1, J_3, J_2, J_4, J_5, J_7, J_6, J_8)$ and the permutation $\pi_m = (J_1, J_2, J_4, J_3, J_5, J_6, J_8, J_7)$. The volume of the stability box

$$\mathcal{SB}(\pi_u, T) = \left[\frac{w_4}{d_4^+}, \frac{w_4}{d_4^-}\right] \times \left[\frac{w_8}{d_8^+}, \frac{w_8}{d_8^-}\right] = [9, 10] \times [19, 20]$$

is equal to 1. The volume of the stability box

$$\mathcal{SB}(\pi_m, T) = \left[\frac{w_2}{d_2^+}, \frac{w_2}{d_2^-}\right] \times \left[\frac{w_6}{d_6^+}, \frac{w_6}{d_6^-}\right] = [3, 6] \times [12, 15]$$

is equal to $9 = 3 \cdot 3$. It is the same volume of the stability box as that of permutation π_1. Note, however, that the dimension $|N_m|$ of the stability box $\mathcal{SB}(\pi_m, T)$ is equal to 2, while the dimension $|N_1|$ of the stability box $\mathcal{SB}(\pi_1, T)$ of the permutation $\pi_1 \in S^{max}$ is equal to 4. Thus, $\pi_m \notin S^{max}$ since permutation π_m does not possess property (a).

4 Computational Results

There might be several permutations with the largest dimension and relative volume of a stability box $\mathcal{SB}(\pi_t, T)$ since several consecutive jobs in a permutation $\pi_t \in S^{max}$ may have an empty range of possible variations of their processing times preserving the optimality of the permutation π_t. In the computational experiments, we break ties in ordering such jobs by adopting the mid-point heuristic which generates a subsequence of these jobs as a part of an optimal permutation $\pi_m \in S$ for the instance $1|p^M|\sum w_i C_i$ with the scenario $p^M \in T$ defined by (16).

Our choice of the mid-point heuristic is based on the computational results of the experiments conducted in [12] for the problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$ with $10 \leq n \leq 1000$. In those computational results, the subsequence of a permutation $\pi_m \in S$ outperformed both the corresponding subsequence of the permutation $\pi_l \in S$ and that of the permutation $\pi_u \in S$ defined by (14) and (15), respectively.

We coded the algorithm MAX-STABOX combined with the mid-point heuristic for ordering consecutive jobs having an empty range of their processing times preserving the optimality of the permutation $\pi_t \in S^{max}$ in C++. This algorithm was tested on a PC with AMD Athlon (tm) 64 Processor 3200+, 2.00 GHz, 1.96 GB of RAM. We solved (exactly or approximately) a lot of randomly generated instances. Some of the computational results obtained are presented in Tables 2 – 4 for randomly generated instances of the problem $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$ with the number $n \in \{1000, 1100, \ldots, 2000\}$ of jobs.

Each series presented in Tables 2 – 4 contains 100 solved instances with the same combination of the number n of jobs and the same maximal possible error $\delta\%$ of the

random processing times $p_i \in [p_i^L, p_i^U]$. The integer center C of a segment $[p_i^L, p_i^U]$ was generated using the uniform distribution in the range $[L, U]$: $L \leq C \leq U$. The lower bound p_i^L for the possible job processing time $p_i \in [p_i^L, p_i^U]$ was defined as $p_i^L = C \cdot (1 - \frac{\delta}{100})$, the upper bound p_i^U of $p_i \in [p_i^L, p_i^U]$ was defined as $p_i^U = C \cdot (1 + \frac{\delta}{100})$. The same range $[L, U]$ for the varying center C of the segment $[p_i^L, p_i^U]$ was used for all jobs $J_i \in \mathcal{J}$, namely: $L = 1$ and $U = 100$. In Tables 2 – 4, we report computational results for the series of instances of the problem $1|p_i^L \leq p_i \leq p_i^U| \sum w_i C_i$ with the maximal possible errors $\delta\%$ of the job processing times from the set $\{0.25\%, 0.4\%, 0.5\%, 0.75\%, 1\%, 2.5\%, 5\%, 15\%, 25\%\}$.

For each job $J_i \in \mathcal{J}$, the weight $w_i \in R_+^1$ was uniformly distributed in the range $[1, 50]$. It should be noted that the job weights w_i were assumed to be known before scheduling (in contrast to the actual processing times p_i^* of the jobs $J_i \in \mathcal{J}$, which were assumed to be unknown before scheduling).

The number n of jobs in each instance of a series is given in column 1 of Table 2, Table 3 and Table 4. The maximum possible error $\delta\%$ of the random processing times $p_i \in [p_i^L, p_i^U]$ is given in column 2. Column 3 represents the average relative number $|\mathcal{A}|$ of the arcs in the dominance digraph $(\mathcal{J}, \mathcal{A})$ constructed using condition (3) of Theorem 1. The relative number $|\mathcal{A}|$ is calculated in percentages of the number of arcs in the complete circuit-free digraph of order n as follows: $\left(|\mathcal{A}| : \frac{n(n-1)}{2}\right) \cdot 100\%$. Column 4 represents the average dimension $|N_t|$ of the stability box $\mathcal{SB}(\pi_t, T)$ of the permutation π_t with the largest relative volume of a stability box. $|N_k|$ is equal to the number of jobs with a non-zero maximal possible variation of the processing time preserving the optimality of permutation $\pi_t \in S^{max}$. Column 5 represents the average relative volume of the stability box $\mathcal{SB}(\pi_t, T)$ of the permutations π_t with the largest dimension and relative volume of a stability box. If $\mathcal{SB}(\pi_t, T) = T$ for all instances in the series, then column 5 contains the number one.

In the experiments, we answered the question of how large the relative error Δ of the objective function $\gamma = \sum_{i=1}^n w_i C_i$ was for the permutation $\pi_t \in S^{max}$ with the largest dimension and relative volume of a stability box $\mathcal{SB}(\pi_t, T)$:

$$\Delta = \frac{\gamma_{p^*}^t - \gamma_{p^*}}{\gamma_{p^*}},$$

where p^* is the actual scenario (unknown before scheduling), γ_{p^*} is the optimal objective function value for the scenario $p^* \in T$ and $\gamma_{p^*}^t = \sum_{i=1}^n w_i C_i(\pi_t, p^*)$.

Column 6 represents the number of instances (among the 100 instances in a series) for which a permutation π_t with the largest dimension and relative volume of the stability box $\mathcal{SB}(\pi_t, T)$ provides an optimal solution for the instance $1|p^*| \sum w_i C_i$ with the actual processing times $p^* = (p_1^*, p_2^*, \ldots, p_n^*) \in T$.

From the experiments, it follows that, if the maximal possible error of the processing times is not greater than 0.4%, then the dominance digraph $(\mathcal{J}, \mathcal{A})$ is a complete circuit-free digraph. Therefore, the permutation $\pi_t \in S^{max}$ provides an optimal solution for such an instance $1|p^*| \sum w_i C_i$.

The average (maximum) relative error Δ of the objective function value $\gamma_{p^*}^t$ calculated for the permutation $\pi_t \in S^{max}$ constructed by the algorithm MAX-STABOX with

Table 2. Randomly generated instances with $[L, U] = [1, 100]$, $w_i \in [1, 50]$ and $n \in \{1000, 1100, 1200, 1300\}$

Number of jobs n	Maximal error of p_i $\delta\%$	Relative arc number $\lvert\mathcal{A}\rvert$ (in %)	Average dimension $\lvert N_t\rvert$	Relative volume of $\mathcal{SB}(\pi_t, T)$	Number of exact solutions	Average error Δ	Maximal error Δ	CPU time (in s)
1	2	3	4	5	6	7	8	9
1000	0.25%	100	1000	1	100	0	0	8.62
1000	0.4%	100	1000	1	100	0	0	8.56
1000	0.5%	100	989.61	0.227427	11	≈ 0	≈ 0	8.69
1000	0.75%	99.545177	451.29	≈ 0	0	0.000023	0.000031	8.98
1000	1%	99.192559	330.65	≈ 0	0	0.000042	0.000051	8.96
1000	2.5%	97.591726	124	0.000001	0	0.000157	0.000181	8.9
1000	5%	94.889794	54.86	0.001976	0	0.000526	0.000614	8.84
1000	15%	84.39185	12.29	0.011288	0	0.004309	0.004858	8.86
1000	25%	73.954372	4.71	0.09081	0	0.012045	0.013303	8.89
1100	0.25%	100	1100	1	100	0	0	11.51
1100	0.4%	100	1100	1	100	0	0	11.46
1100	0.5%	99.997839	1087.27	0.200252	11	≈ 0	≈ 0	11.51
1100	0.75%	99.539967	478.35	≈ 0	0	0.000023	0.00003	12.1
1100	1%	99.188722	349.3	≈ 0	0	0.000043	0.000049	12.05
1100	2.5%	97.611324	131.01	0.000001	0	0.000155	0.000175	11.8
1100	5%	94.862642	57.35	0.006242	0	0.000528	0.000593	11.79
1100	15%	84.288381	11.46	0.017924	0	0.004371	0.004899	11.76
1100	25%	74.076585	4.29	0.133804	0	0.01189	0.013289	11.8
1200	0.25%	100	1200	1	100	0	0	15.4
1200	0.4%	100	1200	1	100	0	0	15.12
1200	0.5%	99.998	1185.27	0.174959	5	≈ 0	0.000001	15.42
1200	0.75%	99.540619	515.8	≈ 0	0	0.000023	0.000029	16
1200	1%	99.190977	375.34	≈ 0	0	0.000042	0.000051	16.06
1200	2.5%	97.581479	138.75	0.000002	0	0.000156	0.000177	15.81
1200	5%	94.88253	62.06	0.006396	0	0.000534	0.000596	15.51
1200	15%	84.376763	12.88	0.042597	0	0.004332	0.004733	15.33
1200	25%	74.100395	5.01	0.08078	0	0.011872	0.01351	15.21
1300	0.25%	100	1300	1	100	0	0	19.75
1300	0.4%	100	1300	1	100	0	0	19.38
1300	0.5%	99.997583	1280.26	0.084004	2	≈ 0	≈ 0	19.54
1300	0.75%	99.549162	543.2	≈ 0	0	0.000023	0.000026	20.3
1300	1%	99.199789	400.41	≈ 0	0	0.000042	0.000053	20.32
1300	2.5%	97.602491	148.41	0.000004	0	0.000157	0.000186	20.01
1300	5%	94.877326	65.23	0.019927	0	0.000532	0.000588	19.95
1300	15%	84.388473	13.47	0.024207	0	0.004364	0.004758	19.52
1300	25%	73.975873	5.5	0.08254	0	0.011962	0.013812	19.52

respect to the optimal objective function value γ_{p^*} defined for the actual job processing times is given in column 7 (in column 8, respectively).

Table 3. Randomly generated instances with $[L, U] = [1, 100]$, $w_i \in [1, 50]$ and $n \in \{1400, 1500, 1600, 1700\}$

Number of jobs n	Maximal error of p_i $\delta\%$	Relative arc number $\lvert\mathcal{A}\rvert$ (in %)	Average dimension $\lvert N_t\rvert$	Relative volume of $\mathcal{SB}(\pi_t, T)$	Number of exact solutions	Average error Δ	Maximal error Δ	CPU time (in s)
1	2	3	4	5	6	7	8	9
1400	0.25%	100	1400	1	100	0	0	24.92
1400	0.4%	100	1400	1	100	0	0	24.8
1400	0.5%	99.997556	1377.21	0.078809	1	≈ 0	0.000001	24.97
1400	0.75%	99.539142	575.2	≈ 0	0	0.000023	0.000029	25.67
1400	1%	99.198461	422.65	≈ 0	0	0.000042	0.00005	25.63
1400	2.5%	97.594897	154.9	0.000001	0	0.000157	0.000178	25.1
1400	5%	94.869044	70.36	0.002356	0	0.000533	0.000615	25.29
1400	15%	84.364242	14.35	0.029338	0	0.004339	0.004841	24.72
1400	25%	74.096446	5.18	0.14077	0	0.011998	0.013041	24.27
1500	0.25%	100	1500	1	100	0	0	31.44
1500	0.4%	100	1500	1	100	0	0	31.08
1500	0.5%	99.997493	1474.09	0.070241	0	≈ 0	0.000001	31.64
1500	0.75%	99.544441	607.5	≈ 0	0	0.000042	0.000052	32.39
1500	1%	99.193199	444.29	≈ 0	0	0.000042	0.000052	32.39
1500	2.5%	97.61593	167.25	0.000005	0	0.000155	0.000171	31.43
1500	5%	94.861654	71.34	0.00282	0	0.000533	0.000582	31.36
1500	15%	84.409904	14.93	0.05372	0	0.004394	0.00492	30.46
1500	25%	74.281235	5.46	0.148403	0	0.011936	0.013685	30.33
1600	0.25%	100	1600	1	100	0	0	38.63
1600	0.4%	100	1600	1	100	0	0	38.67
1600	0.5%	99.997452	1569.35	0.046151	0	≈ 0	0.000001	38.8
1600	0.75%	99.54273	638.18	≈ 0	0	0.000023	0.00003	39.76
1600	1%	99.192323	464.89	≈ 0	0	0.000042	0.000048	40.04
1600	2.5%	97.601128	174.91	0.000004	0	0.000157	0.000177	38.71
1600	5%	94.861356	76.990000	0.003505	0	0.000532	0.000581	38.46
1600	15%	84.343239	14.75	0.036278	0	0.004341	0.004811	37.34
1600	25%	74.123830	5.75	0.087651	0	0.011899	0.013192	36.34
1700	0.25%	100	1700	1	100	0	0	47.29
1700	0.4%	100	1700	1	100	0	0	47.18
1700	0.5%	99.997432	1665.41	0.034556	1	≈ 0	0.000001	47.12
1700	0.75%	99.544993	671.09	≈ 0	0	0.000023	0.000027	48.25
1700	1%	99.203930	495.13	≈ 0	0	0.000041	0.000049	48.47
1700	2.5%	97.598734	180.99	0.000072	0	0.000156	0.000172	46.88
1700	5%	94.852439	80.53	0.001601	0	0.000533	0.000585	46.33
1700	15%	84.358524	17.27	0.028854	0	0.004379	0.0049	45.26
1700	25%	74.030579	6.03	0.082325	0	0.012069	0.013255	44.24

For all series presented in Tables 2 – 4, the average (maximum) error Δ of the value $\gamma^t_{p^*}$ of the objective function $\gamma = \sum_{i=1}^{n} w_i C_i$ obtained for the permutation $\pi_t \in S^{max}$

Table 4. Randomly generated instances with $[L, U] = [1, 100]$, $w_i \in [1, 50]$ and $n \in \{1800, 1900, 2000\}$

Number of jobs n	Maximal error of p_i $\delta\%$	Relative arc number $\lvert\mathcal{A}\rvert$ (in %)	Average dimension $\lvert N_t\rvert$	Relative volume of $\mathcal{SB}(\pi_t, T)$	Number of exact solutions	Average error Δ	Maximal error Δ	CPU time (in s)
1	2	3	4	5	6	7	8	9
1800	0.25%	100	1800	1	100	0	0	56.18
1800	0.4%	100	1800	1	100	0	0	56.27
1800	0.5%	99.99761	1764.02	0.02624	0	≈ 0	0.000001	56.72
1800	0.75%	99.547537	706.21	≈ 0	0	0.000023	0.000028	57.38
1800	1%	99.193797	517.06	≈ 0	0	0.000042	0.000049	57.33
1800	2.5%	97.600247	190.97	0.000042	0	0.000156	0.000177	55.81
1800	5%	94.899074	84.82	0.007274	0	0.000529	0.000602	55.27
1800	15%	84.408342	17.67	0.040758	0	0.004348	0.004723	53.42
1800	25%	74.162869	6.38	0.126377	0	0.011981	0.013095	51.86
1900	0.25%	100	1900	1	100	0	0	65.65
1900	0.4%	100	1900	1	100	0	0	66.81
1900	0.5%	99.997533	1858.51	0.018832	0	≈ 0	0.000001	66.69
1900	0.75%	99.54191	733.81	≈ 0	0	0.000023	0.000028	67.75
1900	1%	99.189512	534.79	≈ 0	0	0.000042	0.000049	68.58
1900	2.5%	97.596318	199.82	0.000022	0	0.000156	0.000173	66.36
1900	5%	94.856400	89.93	0.002011	0	0.000534	0.000596	65.68
1900	15%	84.331351	17.61	0.048813	0	0.004372	0.004844	62.97
1900	25%	74.188836	6.82	0.092068	0	0.011965	0.013234	60.74
2000	0.25%	100	2000	1	100	0	0	78.41
2000	0.4%	100	2000	1	100	0	0	78.93
2000	0.5%	99.997489	1953.88	0.017798	2	≈ 0	≈ 0	79.06
2000	0.75%	99.542435	764.35	≈ 0	0	0.000023	0.000027	78.83
2000	1%	99.197383	565.09	≈ 0	0	0.000042	0.000048	78.1
2000	2.5%	97.605895	210.17	0.000035	0	0.000156	0.000173	75.8
2000	5%	94.867102	93.63	0.014015	0	0.000535	0.000606	75.02
2000	15%	84.412199	17.95	0.040101	0	0.004339	0.004751	74.08
2000	25%	73.977021	6.64	0.147426	0	0.01203	0.013046	71.22

with the largest dimension and relative volume of a stability box was not greater than 0.012069 (not greater than 0.013812).

The CPU-time for an instance of a series is presented in column 5. This time includes the time for the realization of the $O(n^2)$ algorithm for constructing the dominance digraph $(\mathcal{J}, \mathcal{A})$ using condition (3) of Theorem 1 and the time for the realization of the $O(n \log n)$ algorithm MAX-STABOX for constructing the permutation $\pi_t \in S^{max}$ and the stability box $\mathcal{SB}(\pi_t, T)$. This CPU-time grows rather slowly with n, and it was not greater than 79.06 s for each instance.

5 Conclusions

In [12], an $O(n^2)$ algorithm has been developed for calculating a permutation $\pi_t \in S$ with the largest dimension and volume of a stability box $\mathcal{SB}(\pi_t, T)$. In Section 3, we proved Properties 1 – 6 of a stability box allowing us to derive an $O(n \log n)$ algorithm for calculating such a permutation $\pi_t \in S^{max}$. The dimension and volume of a stability box are efficient invariants of the uncertain data T, as it was shown in simulation experiments on a PC reported in Section 4.

The results that we presented may be generalized to the problem $1|prec, p_i^L \leq p_i \leq p_i^U| \sum w_i C_i$, where the precedence constraints are given a priori on the set of jobs. If the deterministic problem $1|prec| \sum w_i C_i$ for a particular type of precedence constraints is polynomially solvable, then the above results may be used for the uncertain counterpart $1|prec, p_i^L \leq p_i \leq p_i^U| \sum w_i C_i$. In the latter problem, the dominance digraph $(\mathcal{J}, \mathcal{A})$ contains the arc (J_u, J_v) only if this arc does not violate the precedence constraint given between the jobs J_u and J_v a priori.

Acknowledgements. The first and second authors were supported in this research by National Science Council of Taiwan. The authors are grateful to Natalja G. Egorova for coding algorithm MAX-STABOX and other contributions.

References

1. Daniels, R., Kouvelis, P.: Robust scheduling to hedge against processing time uncertainty in single stage production. Management Science 41(2), 363–376 (1995)
2. Kasperski, A.: Minimizing maximal regret in the single machine sequencing problem with maximum lateness criterion. Operations Research Letters 33, 431–436 (2005)
3. Kasperski, A., Zelinski, P.: A 2-approximation algorithm for interval data minmax regret sequencing problem with total flow time criterion. Operations Research Letters 36, 343–344 (2008)
4. Lai, T.-C., Sotskov, Y.: Sequencing with uncertain numerical data for makespan minimization. Journal of the Operations Research Society 50, 230–243 (1999)
5. Lai, T.-C., Sotskov, Y., Sotskova, N., Werner, F.: Optimal makespan scheduling with given bounds of processing times. Mathematical and Computer Modelling 26(3), 67–86 (1997)
6. Pinedo, M.: Scheduling: Theory, Algorithms, and Systems. Prentice-Hall, Englewood Cliffs (2002)
7. Sabuncuoglu, I., Goren, S.: Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. International Journal of Computer Integrated Manufacturing 22(2), 138–157 (2009)
8. Slowinski, R., Hapke, M.: Scheduling under Fuzziness. Physica-Verlag, Heidelberg (1999)
9. Smith, W.: Various optimizers for single-stage production. Naval Research Logistics Quarterly 3(1), 59–66 (1956)
10. Sotskov, Y., Egorova, N., Lai, T.-C.: Minimizing total weighted flow time of a set of jobs with interval processing times. Mathematical and Computer Modelling 50, 556–573 (2009)
11. Sotskov, Y., Egorova, N., Werner, F.: Minimizing total weighted completion time with uncertain data: A stability approach. Automation and Remote Control 71(10), 2038–2057 (2010)

12. Sotskov, Y., Lai, T.-C.: Minimizing total weighted flow time under uncertainty using dominance and a stability box. Computers & Operations Research 39, 1271–1289 (2012)
13. Sotskov, Y., Sotskova, N., Lai, T.-C., Werner, F.: Scheduling under Uncertainty. Theory and Algorithms. Belorusskaya nauka, Minsk (2010)
14. Sotskov, Y., Wagelmans, A., Werner, F.: On the calculation of the stability radius of an optimal or an approximate schedule. Annals of Operations Research 83, 213–252 (1998)

Numerical Modelling of Nonlinear Diffusion Phenomena on a Sphere

Yuri N. Skiba and Denis M. Filatov

[1] Centro de Ciencias de la Atmósfera (CCA), Universidad Nacional Autónoma de México (UNAM), Av. Universidad #3000, Cd. Universitaria, C.P. 04510, México D.F., México
[2] Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional (IPN), Av. Juan de Dios Bátiz s/n, esq. Miguel Othón de Mendizábal, C.P. 07738, México D.F., México
skiba@unam.mx, denisfilatov@gmail.com

Abstract. A new method for the numerical modelling of physical phenomena described by nonlinear diffusion equations on a sphere is developed. The key point of the method is the splitting of the differential equation by coordinates that reduces the original 2D problem to a pair of 1D problems. Due to the splitting, while solving the 1D problems separately one from another we involve the procedure of map swap — the same sphere is covered by either one or another of two different coordinate grids, which allows employing periodic boundary conditions for both 1D problems, despite the sphere is, actually, *not* a doubly periodic domain. Hence, we avoid the necessity of dealing with cumbersome mathematical procedures, such as the construction of artificial boundary conditions at the poles, etc. As a result, second-order finite difference schemes for the one-dimentional problems implemented as systems of linear algebraic equations with tridiagonal matrices are constructed. It is essential that each split one-dimentional finite difference scheme keeps all the substantial properties of the corresponding differential problem: the spatial finite difference operator is negative definite, whereas the scheme itself is balanced and dissipative. The results of several numerical simulations are presented and thoroughly analysed. Increase of the accuracy of the finite difference schemes to the fourth approximation order in space is discussed.

1 Introduction

The diffusion equation is relevant to a wide range of important physical phenomena and has many applications [1,2,3,8]. The classical problem is the heat or pollution transfer in the atmosphere. A more sophisticated example is the single-particle Schrödinger equation which, formally speaking, is diffusion-like.

In many practical applications the diffusion equation has to be studied on a sphere, and besides, in the most general case it is nonlinear. Hence, consider the nonlinear diffusion equation on a sphere $S = \{(\lambda, \varphi) : \lambda \in [0, 2\pi), \varphi \in (-\frac{\pi}{2}, \frac{\pi}{2})\}$

$$\frac{\partial T}{\partial t} = \frac{1}{R\cos\varphi}\left[\frac{\partial}{\partial\lambda}\left(\frac{\mu T^{\alpha}}{R\cos\varphi}\frac{\partial T}{\partial\lambda}\right) + \frac{\partial}{\partial\varphi}\left(\frac{\mu T^{\alpha}\cos\varphi}{R}\frac{\partial T}{\partial\varphi}\right)\right] + f \qquad (1)$$

equipped with a suitable initial condition. Here $T = T(\lambda, \varphi, t) \geq 0$ is the unknown function, μT^{α} is the diffusion coefficient, $\mu = \mu(\lambda, \varphi, t) \geq 0$ is the amplification

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 57–70.
DOI: 10.1007/978-3-642-34336-0_4

factor, $f = f(\lambda, \varphi, t)$ is the source function, R is the radius of the sphere with λ as the longitude (positive eastward) and φ as the latitude (positive northward). The parameter α is usually a positive integer number that determines the degree of nonlinearity of the diffusion process; if $\alpha = 0$ then we are dealing with linear diffusion.

The diffusion equation has a few important properties retaled to the physics of the diffusion process [9]. First, integration of (1) over the sphere results in the balance equation

$$\frac{d}{dt}\int_S T dS = \int_S f dS \,, \tag{2}$$

which under $f = 0$ provides the mass conservation law

$$\frac{d}{dt}\int_S T dS = \text{const} \,; \tag{3}$$

second, multiplication of (1) by T and integration over S yields

$$\frac{1}{2}\frac{d}{dt}\int_S T^2 dS \le \int_S f T dS \,, \tag{4}$$

from where under $f = 0$ we obtain the solution's dissipation in the $L_2(S)$-norm

$$\frac{d}{dt}\int_S T^2 dS \le 0 \,. \tag{5}$$

Obviously, equation (1) does not admit the analytical solution unless a simple particular case (e.g., the diffusion coefficient is constant and the sources are absent) is being studied. Therefore, our aim is to develop an accurate and computationally efficient method for the numerical simulation of nonlinear diffusion processes. This problem is complicated by two things. First, one has to treat somehow the poles, where the differential equation becomes meaningless. Second, although the sphere can be considered as a periodic domain in the longitude, it is *not* periodic in the latitude — again, due to the presence of the poles, — so, the question of constructing suitable boundary conditions arises.

2 Mathematical Foundations

In this Section we provide the mathematical foundation of the new method for the numerical solution of nonlinear diffusion equations on a sphere. First, involving the operator splitting by coordinates (also known as dimensional splitting) and afterward inventing the procedure of map swap, we shall develop a couple of finite difference schemes of the second approximation order both in time and in space. Then we shall study the properties of the resulting schemes.

So, in the time interval (t_n, t_{n+1}) we linearise equation (1) as follows

$$\frac{\partial T}{\partial t} = \frac{1}{R\cos\varphi}\left[\frac{\partial}{\partial\lambda}\left(\frac{D}{R\cos\varphi}\frac{\partial T}{\partial\lambda}\right) + \frac{\partial}{\partial\varphi}\left(\frac{D\cos\varphi}{R}\frac{\partial T}{\partial\varphi}\right)\right] + f \,, \tag{6}$$

where

$$D = \mu \left(T^n\right)^{\alpha} \ , \tag{7}$$

while $T^n = T(\lambda, \varphi, t_n)$. Then we split equation (6) by coordinates as follows [4]

$$\frac{\partial T}{\partial t} = A_\lambda T + \frac{f}{2} \equiv \frac{1}{R\cos\varphi}\frac{\partial}{\partial\lambda}\left(\frac{D}{R\cos\varphi}\frac{\partial T}{\partial\lambda}\right) + \frac{f}{2} \ , \tag{8}$$

$$\frac{\partial T}{\partial t} = A_\varphi T + \frac{f}{2} \equiv \frac{1}{R\cos\varphi}\frac{\partial}{\partial\varphi}\left(\frac{D\cos\varphi}{R}\frac{\partial T}{\partial\varphi}\right) + \frac{f}{2} \ . \tag{9}$$

This means that in order to find the solution at a time moment t_{n+1} one has first to solve equation (8) in λ using the function $T(\lambda, \varphi, t)$ at t_n as the initial condition. Then, taking the resulting solution as the initial condition, one has to solve (9) in φ. The outcome will be the (approximate) solution to (1) at t_{n+1}. In the next time interval (t_{n+1}, t_{n+2}) the succession has to be repeated, and so on.

Two challenges are met here.

First, because the term $R\cos\varphi$ vanishes at $\varphi = \pm\pi/2$, equations (8)-(9) have no sense at the poles. Therefore, defining the grid steps $\Delta\lambda = \lambda_{k+1} - \lambda_k$ and $\Delta\varphi = \varphi_{l+1} - \varphi_l$, we create a half-step-shifted λ-grid

$$S^{(1)}_{\Delta\lambda,\Delta\varphi} = \left\{(\lambda_k, \varphi_l) : \lambda_k \in \left[\tfrac{\Delta\lambda}{2}, 2\pi + \tfrac{\Delta\lambda}{2}\right), \varphi_l \in \left[-\tfrac{\pi}{2} + \tfrac{\Delta\varphi}{2}, \tfrac{\pi}{2} - \tfrac{\Delta\varphi}{2}\right]\right\} \ . \tag{10}$$

The half step shift in φ allows excluding the pole singularities, and therefore the corresponding finite difference equation will have sense everywhere on $S^{(1)}_{\Delta\lambda,\Delta\varphi}$ (Fig. 1). The equation is enclosed with the periodic boundary condition, since the sphere is a periodic domain in λ.

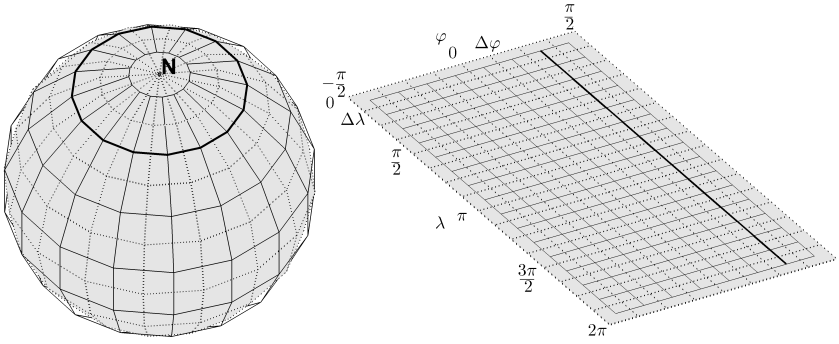


Fig. 1. The grid shown in the solid line is used while solving in λ. The semi-integer shift in φ allows excluding the pole singularities, which keeps the equation to have sense on the entire sphere.

Second, while solving in φ, equation (9) has to be enclosed with two boundary conditions at the poles. It is well known, however, that the construction of an appropriate boundary condition is always a serious problem, because a lot of undesired effects may

emerge therewith [7]. Leaving details, the boundary condition must lead to a well-posed problem, as well as be adequate to the phenomenon being modelled from the physical standpoint. To overcome this difficulty, we suggest it should be used the map swap — instead of λ_k being varied from the zero meridian around the sphere and φ_l from the South pole to the North, one should change the coordinate map from (10) to

$$S^{(2)}_{\Delta\lambda,\Delta\varphi} = \left\{ (\lambda_k, \varphi_l) : \lambda_k \in \left[\tfrac{\Delta\lambda}{2}, \pi - \tfrac{\Delta\lambda}{2} \right], \varphi_l \in \left[-\tfrac{\pi}{2} + \tfrac{\Delta\varphi}{2}, \tfrac{3\pi}{2} + \tfrac{\Delta\varphi}{2} \right) \right\} . \quad (11)$$

Obviously, both maps cover the entire sphere and consist of the same grid nodes. The use of map (11) allows treating the solution as periodic while computing in φ, similarly to how it is in λ (Fig. 2).

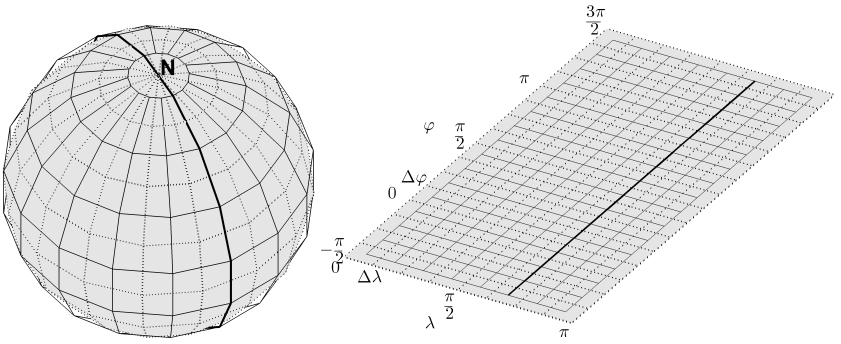


Fig. 2. The grid shown in the solid line is used while solving in φ. This allows considering the sphere as a periodic domain in the latitude, without the necessity of constructing boundary conditions at the poles.

Having armed ourselves with (10)-(11), now we are ready for the discretisation of the split 1D problems.

Typically, let $G_{kl} = G(\lambda_k, \varphi_l)$, where G is any of the functions $T, \overline{D}, f$. Approximate (8)-(9) as follows

$$\frac{T_k^{n+\frac{1}{2}} - T_k^n}{\tau} = \underbrace{\frac{1}{R^2 \cos^2\varphi} \frac{1}{\Delta\lambda} \left(\overline{D}_{k+1/2} \frac{T_{k+1} - T_k}{\Delta\lambda} - \overline{D}_{k-1/2} \frac{T_k - T_{k-1}}{\Delta\lambda} \right)}_{A_{\Delta\lambda} T_k} + \frac{f_k^{n+\frac{1}{2}}}{2} , \quad (12)$$

$$\frac{T_l^{n+1} - T_l^{n+\frac{1}{2}}}{\tau} = \underbrace{\frac{1}{R^2 |\cos\varphi_l|} \frac{1}{\Delta\varphi} \left(\overline{D}_{l+1/2} \frac{T_{l+1} - T_l}{\Delta\varphi} - \overline{D}_{l-1/2} \frac{T_l - T_{l-1}}{\Delta\varphi} \right)}_{A_{\Delta\varphi} T_l} + \frac{f_l^{n+\frac{1}{2}}}{2} , \quad (13)$$

where $\overline{D} = D$ in (12) and $\overline{D} = D|\cos\varphi|$ in (13), as well as

$$\overline{D}_{i\pm1/2} := \frac{\overline{D}_{i\pm1} + \overline{D}_i}{2} \,. \tag{14}$$

(We omitted the nonvarying index l in (12) and the nonvarying index k in (13) for clarity.) As for the function T_{kl}, we involve the Crank-Nicolson approximation

$$T_{kl} := \frac{T_{kl}^{n+\frac{p}{2}} + T_{kl}^{n+\frac{p-1}{2}}}{2} \,, \tag{15}$$

where $p = 1$ for (12) and $p = 2$ for (13). Substituting (15) into (12)-(13), we come to the systems of linear algebraic equations in λ

$$\begin{aligned} &- T_{k+1}^{n+\frac{1}{2}} m_{k+1} + T_k^{n+\frac{1}{2}} \left(\frac{1}{\tau} + m_k\right) - T_{k-1}^{n+\frac{1}{2}} m_{k-1} \\ &= T_{k+1}^{n} m_{k+1} + T_k^{n} \left(\frac{1}{\tau} - m_k\right) + T_{k-1}^{n} m_{k-1} + \frac{f_k^{n+\frac{1}{2}}}{2} \,, \end{aligned} \tag{16}$$

where

$$m_k = \frac{\overline{D}_{k+1/2} + \overline{D}_{k-1/2}}{2R^2 \cos^2\varphi_l \Delta\lambda^2} \,, \qquad m_{k+j} = \frac{\overline{D}_{k+j/2}}{2R^2 \cos^2\varphi_l \Delta\lambda^2} \,, \qquad j = \pm1 \,, \tag{17}$$

and in φ

$$\begin{aligned} &- T_{l+1}^{n+1} m_{l+1} + T_l^{n+1} \left(\frac{1}{\tau} + m_l\right) - T_{l-1}^{n+1} m_{l-1} \\ &= T_{l+1}^{n+\frac{1}{2}} m_{l+1} + T_l^{n+\frac{1}{2}} \left(\frac{1}{\tau} - m_l\right) + T_{l-1}^{n+\frac{1}{2}} m_{l-1} + \frac{f_l^{n+\frac{1}{2}}}{2} \,, \end{aligned} \tag{18}$$

where

$$m_l = \frac{\overline{D}_{l+1/2} + \overline{D}_{l-1/2}}{2R^2 |\cos\varphi_l| \Delta\varphi^2} \,, \qquad m_{l+j} = \frac{\overline{D}_{l+j/2}}{2R^2 |\cos\varphi_l| \Delta\varphi^2} \,, \qquad j = \pm1 \,. \tag{19}$$

Using the procedure of bicyclic splitting [4]

$$\frac{T_{kl}^{n+\frac{1}{4}} - T_{kl}^{n}}{\tau/2} = A_{\Delta\lambda} T_{kl} + \frac{f_k^{n+\frac{2}{4}}}{2} \,, \tag{20}$$

$$\frac{T_{kl}^{n+\frac{2}{4}} - T_{kl}^{n+\frac{1}{4}}}{\tau/2} = A_{\Delta\varphi} T_{kl} + \frac{f_l^{n+\frac{2}{4}}}{2} \,, \tag{21}$$

$$\frac{T_{kl}^{n+\frac{3}{4}} - T_{kl}^{n+\frac{2}{4}}}{\tau/2} = A_{\Delta\varphi} T_{kl} + \frac{f_l^{n+\frac{2}{4}}}{2} \,, \tag{22}$$

$$\frac{T_{kl}^{n+1} - T_{kl}^{n+\frac{3}{4}}}{\tau/2} = A_{\Delta\lambda} T_{kl} + \frac{f_k^{n+\frac{2}{4}}}{2} \,, \tag{23}$$

we increase the temporal approximation accuracy for the linearised problem in the time interval (t_n, t_{n+1}) up to the second order. Note that here

$$T_{kl} := \frac{T_{kl}^{n+\frac{p}{4}} + T_{kl}^{n+\frac{p-1}{4}}}{2}\,, \qquad p = \overline{1,4}\,. \tag{24}$$

Theorem 1. *Finite difference schemes (12)-(13) are balanced.*

Proof. Consider, e.g., (16). Multiplying the left-hand side by $\tau R \Delta\lambda$, we have

$$\left(-T_{k+1}^{n+\frac{1}{2}} m_{k+1} + T_k^{n+\frac{1}{2}}\left(\frac{1}{\tau} + m_k\right) - T_{k-1}^{n+\frac{1}{2}} m_{k-1}\right)\tau R\Delta\lambda = T_k^{n+\frac{1}{2}} R\Delta\lambda$$
$$- \left(T_{k+1}^{n+\frac{1}{2}} D_{k+1/2} + T_{k-1}^{n+\frac{1}{2}} D_{k-1/2} + T_k^{n+\frac{1}{2}}(D_{k+1/2} + D_{k-1/2})\right)\frac{\tau}{2R\cos^2\varphi_l \Delta\lambda} \tag{25}$$

Summing all over the k's, due to the periodicity in λ the terms with $D_{k\pm1/2}$ cancel. Doing in the same manner with the right-hand side of (16), we find

$$R\Delta\lambda \sum_k \left(T_k^{n+\frac{1}{2}} - T_k^n\right) = \frac{\tau}{2} R\Delta\lambda \sum_k f_k^{n+\frac{1}{2}}\,, \tag{26}$$

that is scheme (12) is balanced. In particular, under $f_k^{n+\frac{1}{2}} = 0$ we obtain the mass conservation law at a fixed latitude φ_l

$$\sum_k T_k^{n+\frac{1}{2}} = \sum_k T_k^n\,. \tag{27}$$

Calculations for (13) can be done in a similar way. □

Theorem 2. *The finite difference operators $A_{\Delta\lambda}$ and $A_{\Delta\varphi}$ in (12)-(13) are negative definite.*

Proof. Consider (13) on grid (11). Multiply the right-hand side by $T_l|\cos\varphi_l|$ and sum all over the l's. It holds

$$\sum_l \frac{1}{R^2|\cos\varphi_l|}\frac{1}{\Delta\varphi}\left(\overline{D}_{l+1/2}\frac{T_{l+1}-T_l}{\Delta\varphi} - \overline{D}_{l-1/2}\frac{T_l - T_{l-1}}{\Delta\varphi}\right) T_l|\cos\varphi_l|$$
$$= \frac{1}{R^2\Delta\varphi^2}\left(\sum_l \overline{D}_{l+1/2}(T_{l+1}-T_l)T_l - \sum_l \overline{D}_{l-1/2}(T_l - T_{l-1})T_l\right)$$
$$= \frac{1}{R^2\Delta\varphi^2}\left(\sum_l \overline{D}_{l+1/2}(T_{l+1}-T_l)T_l - \sum_{l'} \overline{D}_{l'+1/2}(T_{l'+1} - T_{l'})T_{l'+1}\right)$$
$$= \frac{1}{R^2\Delta\varphi^2}\sum_l \overline{D}_{l+1/2}(T_{l+1}-T_l)(T_l - T_{l+1})$$
$$= -\frac{1}{R^2\Delta\varphi^2}\sum_l \overline{D}_{l+1/2}(T_{l+1}-T_l)^2 \le 0\,. \tag{28}$$

Here $l' = l - 1$, and we used the periodicity of the solution in φ.

Calculations for (12) are similar. □

Theorem 3. *Finite difference schemes (12)-(13) are dissipative.*

Proof. Consider (13). Multiplying both sides by $R\Delta\varphi T_l|\cos\varphi_l|$ and summing all over the nodes l's, given the Crank-Nicolson approximation (15) we obtain

$$\frac{||T^{n+1}||^2 - ||T^{n+\frac{1}{2}}||^2}{2\tau} = \langle A_{\Delta\varphi}T_l, T_l\rangle + \left\langle \frac{f_l^{n+\frac{1}{2}}}{2}, T_l \right\rangle . \tag{29}$$

Here $\langle\cdot,\cdot\rangle$ denotes the scalar product on $S^{(2)}_{\Delta\lambda,\Delta\varphi}$ at a fixed λ_k. Due to the negative definiteness of the operator $A_{\Delta\varphi}$ (Theorem 2) the first summand on the right-hand side of (29) is less than or equal to zero. Consequently, if $f_l^{n+\frac{1}{2}} = 0$ then

$$||T^{n+1}||^2 \le ||T^{n+\frac{1}{2}}||^2 , \tag{30}$$

that is the solution's L_2-norm decays in time.

Analogous calculations can be performed for (12). □

Corollary 1. *Finite difference schemes (12)-(13) on grids (10)-(11) are absolutely stable in the corresponding L_2-norms.*

All the statements of the aforegiven theorems and corollary 1 are true for problem (20)-(23).

3 Numerical Experiments

Our purpose is now to test the developed method. For this we shall numerically simulate several diffusion phenomena. We shall start from the simplest linear model, then consider a little more complicated nonlinear case, and finally prove our schemes on a highly nonlinear diffusion model.

Experiment 1. First we have to verify how the idea of the map swap works — whether nonphysical (purely computational) effects appear near the poles due to the convergence of meridians. To do this, we set $\alpha = 0$, μ constant, $f = 0$ and take the initial condition in the form of a disc-like spot located near a pole. From the theory it is known that the spot should isotropically propagate from the centre in all possible directions without any disturbance in the disc's shape.

The numerical solution on the grid $6^\circ \times 6^\circ$ at a few time moments is shown in Fig. 3. In Fig. 4 we also plot the solution's L_2-norm in time. The solution is seen to be consistent with what we have been expecting — the spot is isotropically spreading over the sphere and no visual disturbance of the spot's shape is observed while passing over the North pole. Yet, the graph of the L_2-norm proves the dissipativity of the schemes (cf. Theorem 3).

Therefore, we may conclude that the procedure of the map swap is mathematically correct and the diffusion process through the poles is being simulated physically adequate on the shifted grids (10)-(11).

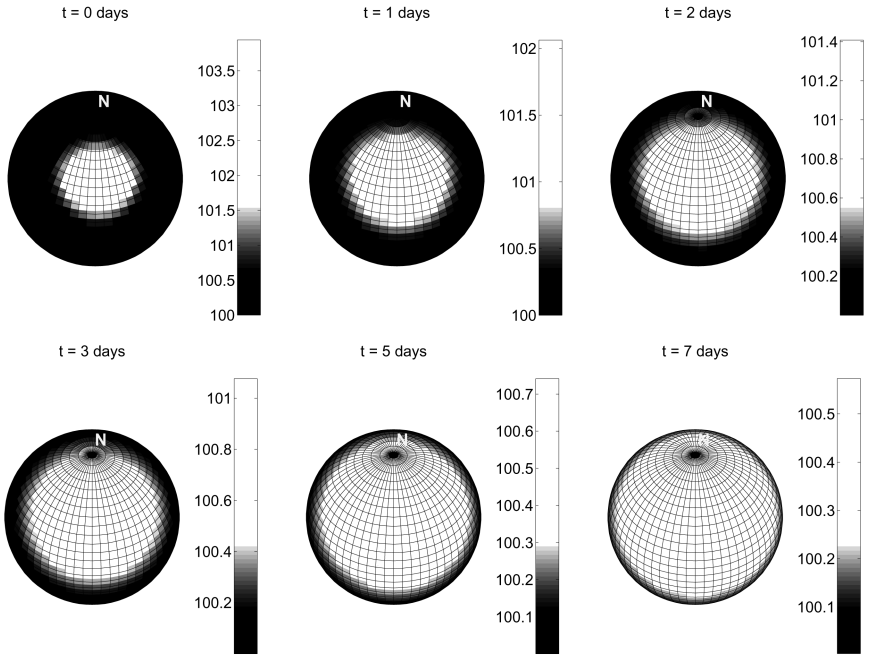


Fig. 3. Experiment 1: numerical solution at several time moments (a nonmonotonic colour map is used for better visualisation)

Experiment 2. Consider a more intricate problem, e.g., a nonlinear diffusion process in a heterogeneous medium. For this we set $\alpha = 1$ and complicate the problem, taking

$$\mu(\lambda, \varphi) \sim \sin^3 \frac{\lambda}{2} \sin^2 \varphi \, . \tag{31}$$

Let the initial condition be the same spot, but now placed on the North pole. The anisotropy of the medium occurs because the diffusion process is taking place in a longitudinal sector — since the asymmetry is concentrated in the diffusion factor μ, we are expecting intensive diffusion at those λ's (at a fixed latitude) where μ has a maximum, while poor diffusion is expected where μ is almost zero (Fig. 5).

The numerical solution shown in Fig. 6 confirms the expectations. Indeed, a strong diffusion process is being observed at $\lambda \approx \pi$, while weak diffusion is taking place at $\lambda \approx 0$, which is consistent with the profile of the diffusion factor (31).

Experiment 3. The aim of this experiment is to investigate the accuracy of the numerical solution to the split linearised problem that approximates the original unsplit nonlinear differential equation. Besides, of practical interest is the question of large gradients of the solutions that may appear in real physical problems. Therefore, we increase the nonlinearity of the problem up to $\alpha = 2$ and compare the numerical solution versus the analytical one

$$T(\lambda, \varphi, t) = c_1 \sin \xi \cos \varphi \cos^2 t + c_2 \, , \tag{32}$$

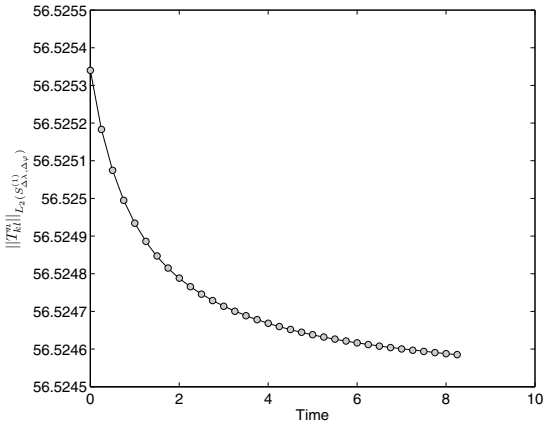


Fig. 4. Experiment 1: graph of the L_2-norm of the solution in time

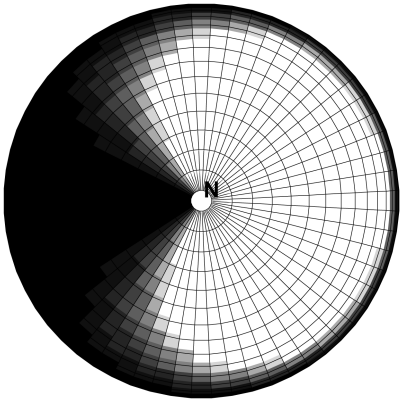


Fig. 5. Experiment 2: the profile of the diffusion factor μ (North pole view). The colour map ranges from white (maximum) to black (minimum).

where

$$\xi \equiv \omega(\lambda - \vartheta_1 \tan \kappa_1 \varphi) + \vartheta_2 \tan \kappa_2 \varphi \sin \gamma t \ . \tag{33}$$

In order for (32) to be the solution to (1), the sources must be

$$f(\lambda, \varphi, t) = \frac{\partial T}{\partial t} - \frac{\mu T^{\alpha-1}}{R \cos \varphi} \left[\frac{1}{R \cos \varphi} \left(\alpha \left(\frac{\partial T}{\partial \lambda} \right)^2 + T \frac{\partial^2 T}{\partial \lambda^2} \right) + \frac{\cos \varphi}{R} \left(T \left(\frac{\partial^2 T}{\partial \varphi^2} - \tan \varphi \frac{\partial T}{\partial \varphi} \right) + \alpha \left(\frac{\partial T}{\partial \varphi} \right)^2 \right) \right] , \tag{34}$$

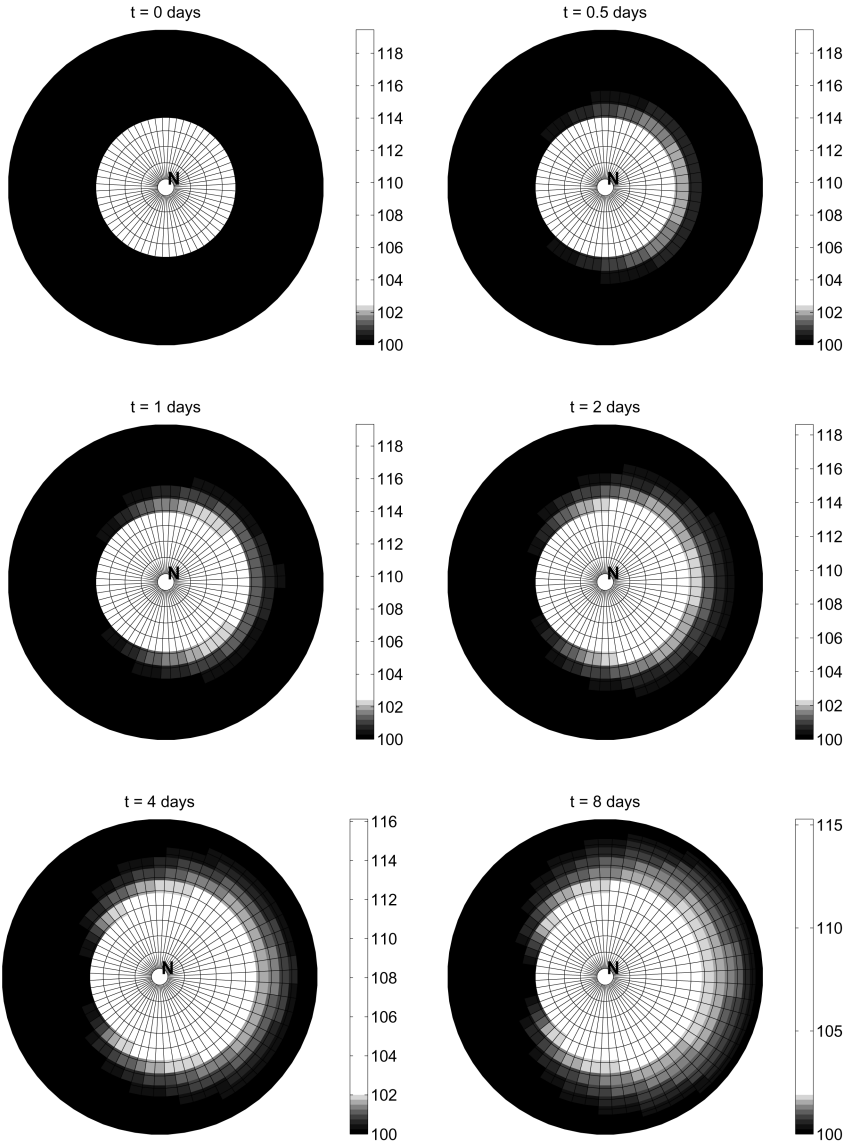


Fig. 6. Experiment 2: numerical solution at several time moments (North pole view)

where

$$\frac{\partial T}{\partial t} = c_1 \cos\varphi \left(\theta_2 \gamma \cos\xi \tan\kappa_2 \varphi \cos\gamma t \cos^2 t - \sin\xi \sin 2t\right),$$

$$\frac{\partial T}{\partial \lambda} = c_1 \omega \cos\varphi \cos^2 t \cos\xi \,,$$

$$\frac{\partial^2 T}{\partial \lambda^2} = -c_1 \omega^2 \cos\varphi \cos^2 t \sin\xi \,,$$

$$\frac{\partial T}{\partial \varphi} = c_1 \cos^2 t \left(\cos\varphi \cos\xi \frac{\partial \xi}{\partial \varphi} - \sin\xi \sin\varphi \right) ,$$

$$\frac{\partial^2 T}{\partial \varphi^2} = c_1 \cos^2 t \left(-2 \sin\varphi \cos\xi \frac{\partial \xi}{\partial \varphi} - \sin\xi \cos\varphi \left(1 + \left(\frac{\partial \xi}{\partial \varphi} \right)^2 \right) \right.$$
$$\left. + 2 \cos\xi \cos\varphi \left(-\omega \theta_1 \kappa_1^2 \frac{\sin \kappa_1 \varphi}{\cos^3 \kappa_1 \varphi} + \theta_2 \kappa_2^2 \sin \gamma t \frac{\sin \kappa_2 \varphi}{\cos^3 \kappa_2 \varphi} \right) \right) ,$$

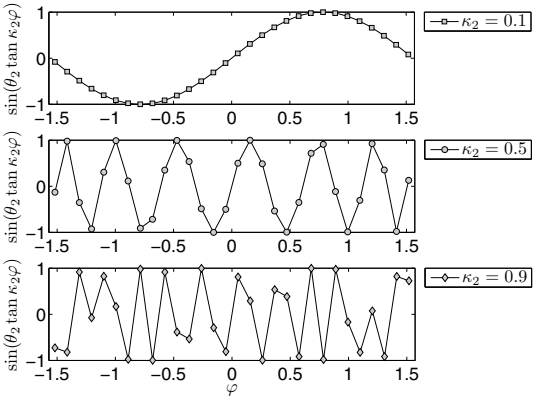


Fig. 7. Experiment 3: graphs of the function $\sin(\theta_2 \tan \kappa_2 \varphi)$ at $\theta_2 = 20$, $\kappa_2 = 0.1, 0.5, 0.9$

$$\frac{\partial \xi}{\partial \varphi} = -\omega \frac{\theta_1 \kappa_1}{\cos^2 \kappa_1 \varphi} + \sin \gamma t \frac{\theta_2 \kappa_2}{\cos^2 \kappa_2 \varphi} .$$

Because $\xi \sim \theta_2 \tan \kappa_2 \varphi$ while the time grows, the term $\sin \xi$ endeavours to simulate large gradients at the high latitudes. Indeed, while at nearly zero κ_2's this function is rather smooth, it becomes a saw-tooth wave as $\kappa_2 \to 1$ (Fig. 7).

This experiment was performed on a series of grids $6^\circ \times 6^\circ$, $4^\circ \times 4^\circ$ and $2^\circ \times 2^\circ$, with $c_1 = 10$, $c_2 = 100$, $\omega = 5$, $\vartheta_1 = 1$, $\kappa_1 = 0.5$, $\vartheta_2 = 20$, $\kappa_2 = 0.5$, $\gamma = 2$, $\mu = 1 \cdot 10^{-7}$. At each time moment t_n the numerical solution was compared with the analytical one in the relative error

$$\delta^n = \frac{||T^{num} - T^{exact}||_{L_2\left(S^{(1)}_{\Delta\lambda, \Delta\varphi}\right)}}{||T^{exact}||_{L_2\left(S^{(1)}_{\Delta\lambda, \Delta\varphi}\right)}} . \tag{35}$$

As it follows from (32)-(34), the forcing produces the spiral structure of the solution near the poles, which results in rather high solution's gradients. In addition, the direction of the spiral is getting changed in time due to the term $\sin \gamma t$. This is exactly what is reproduced by the numerical solution on the grid $6^\circ \times 6^\circ$ shown in Fig. 8. Yet, one can observe the cyclicity of the solution with the period $\frac{2\pi}{\gamma}$ due to the sinusoid $\sin \xi$'s behaviour (cf., e.g., $t = 0$ and $t = 3$; $t = 0.5$ and $t = 3.5$; etc.), consistent with the term

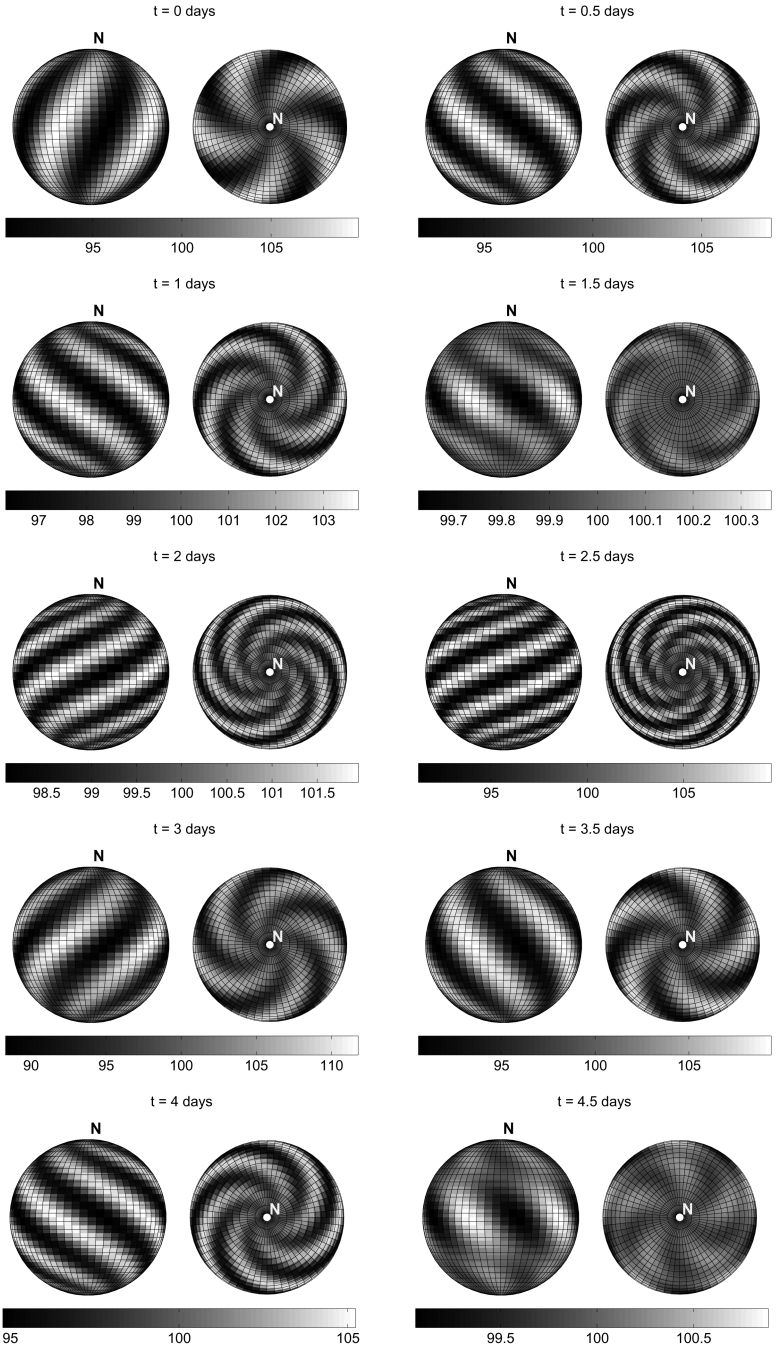


Fig. 8. Experiment 3: numerical solution at several time moments (the sphere is shown from two different angles, from the equator and from the North pole)

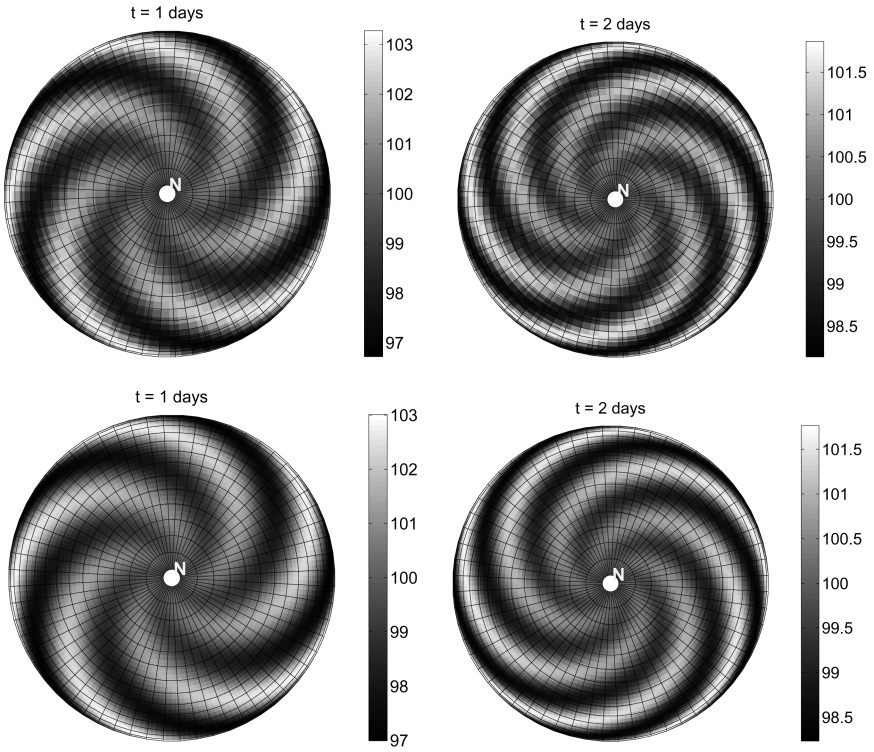


Fig. 9. Experiment 3: numerical solutions at the first two days, grid $4^\circ \times 4^\circ$ (top row) and $2^\circ \times 2^\circ$ (bottom row)

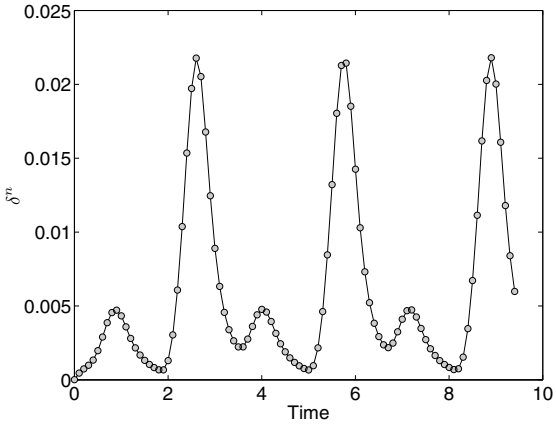


Fig. 10. Experiment 3: graph of the relative error in time

$\sin \gamma t$. In Fig. 9 we also plot the numerical solutions obtained on two finer grids, $4^\circ \times 4^\circ$ and $2^\circ \times 2^\circ$, at the first two days. It is seen that the finer the grid, the more accurate the

spatial approximation of the numerical solution. The maximum relative errors on the chosen grids are $2.20 \cdot 10^{-2}$, $8.91 \cdot 10^{-3}$ and $1.38 \cdot 10^{-3}$, respectively.

In Fig. 10 we also plot the graph of δ^n in time on the grid $6^\circ \times 6^\circ$. According to the periodical character of analytical solution (32) determined by forcing (34), the relative error periodically grows and decays too.

Acknowledgements. This research was partially supported by grants of the National System of Researchers (SNI) of Mexico, No. 14539 and 26073, and is part of the project PAPIIT-UNAM IN104811, Mexico.

References

1. Bear, J.: Dynamics of Fluids in Porous Media. Dover Publications, New York (1988)
2. Catté, F., Lions, P.-L., Morel, J.-M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. 29, 182–193 (1992)
3. Glicksman, M.E.: Diffusion in Solids: Field Theory, Solid-State Principles and Applications. John Wiley & Sons, New York (2000)
4. Marchuk, G.I.: Methods of Computational Mathematics. Springer, Berlin (1982)
5. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, Cambridge (2007)
6. Skiba, Y.N., Filatov, D.M.: On an efficient splitting-based method for solving the diffusion equation on a sphere. Numer. Meth. Part. Diff. Eq. (2011), doi: 10.1002/num.20622
7. Tsynkov, S.V.: Numerical solution of problems on unbounded domains. A review. Appl. Numer. Math. 27, 465–532 (1998)
8. Vorob'yov, A.K.: Diffusion Problems in Chemical Kinetics. Moscow University Press, Moscow (2003)
9. Wu, Z., Zhao, J., Yin, J., Li, H.: Nonlinear Diffusion Equations. World Scientific Publishing, Singapore (2001)

Appendix

As it was noticed in Section 2, this is exactly the dimensional splitting that allows employing periodic boundary conditions in both spatial coordinates, thereby providing a fast and simple numerical algorithm for finding the solution [5].

A not less important benefit of the coordinate splitting is that one may involve spatial stencils of high-order approximations without complicating the computer code. In particular, in [6], for the linear diffusion equation, we tested fourth-order schemes. The limits on the size of the present paper do not permit us to make a comprehensive analysis of the fourth-order schemes in the nonlinear case. This is a subject of a separate study. However, the approach is similar to [6], since due to (7) the nonlinear term gets linearised and the problem reduces to the linear case.

Simulation, Parameter Estimation and Optimization of an Industrial-Scale Evaporation System

Ines Mynttinen, Erich Runge, and Pu Li

Technische Universität Ilmenau, Helmholtzplatz 5, 98693 Ilmenau, Germany
{ines.mynttinen,erich.runge,pu.li}@tu-ilmenau.de

Abstract. Design and operation of complex industrial systems can be improved based on simulation and optimization using physical process models. However, this endeavor is particularly challenging for hybrid systems, where in addition to the continuous evolution described by differential algebraic equations the dynamic process shows instantaneous switches between different operating modes. In this study, we consider parameter estimation for an industrial evaporation system with discrete mode switches due to phase transitions. Simulation results of the hybrid evaporator model are compared with those of a smooth evaporator model. A smoothing approach is applied in order to modify the hybrid model such that the discrete transitions are integrated into the system of differential algebraic equations. This leads to exclusively smooth trajectories, making the model suitable for parameter estimation to be solved by means of gradient-based optimization methods. The dependence of the parameter estimation results on the smoothing parameter is investigated.

Keywords: Parameter estimation, Nonlinear dynamic optimization, Large-scale hybrid systems.

1 Introduction

Simulation and optimization based on physical models are state-of-the-art tools in design and operation of complex industrial systems. Optimization problems result from many tasks such as parameter estimation, data validation, safety verification and model predictive control. The underlying model is given by the dynamic equations of the process under consideration and possibly additional equality and inequality constraints resulting, e.g., from safety specifications. The objective function depends on the particular task. Several powerful methods and computer codes are available for optimization problems which include only continuous system models expressed as a set of differential algebraic equations (DAEs). Unfortunately, in many processes occurring in, e.g., chemical industries, power plants and oil refineries, continuous and discrete state dynamics are coupled strongly. Such systems with mixed continuous and discrete dynamics are called hybrid systems. The discrete dynamics can result from instantaneous autonomous or from controlled (externally triggered) transitions from one operating regime to another. In the time periods between these transition points, the state variables of the system evolve continuously according to the DAEs of the respective operation mode. The trajectories of the state variables are in general non-smooth or even

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 71–84.
DOI: 10.1007/978-3-642-34336-0_5

discontinuous due to the mixed discrete-continuous dynamics. This can, of course, be a major problem for any optimization task. Several approaches, e.g., mixed-integer programming, heuristic methods, relaxation and penalization strategies have been proposed to tackle this problem. The present work focuses on relaxation strategies because they are most promising with regard to the computation time. Hitherto, mostly relatively small systems have been studied using relaxation methods. The present study concerns a large-scale industrial evaporator with switching behavior as either a hybrid model or as a relaxed continuous model. Both models are simulated and parameter sensitivities are calculated over the whole time horizon. For the smooth model, the dependence of the solution on the reformulation parameter is considered in detail.

This paper is organized as follows. Section 2 starts with a discussion of the challenges and solution approaches to simulation and optimization of hybrid dynamic systems. Next, in Section 3, the evaporator model in its hybrid and relaxed form is presented. In Section 4, the simulation results of the relaxed (and consequently smooth continuous) model are compared with those of the original hybrid model. Section 5 studies for the evaporator model two fundamental tasks of process engineering, namely parameter estimation and sensitivity analysis. Section 6 summarizes the results and concludes the paper.

2 Simulation and Optimization of Hybrid Systems

Mathematically, discrete transitions in hybrid dynamic systems are often formulated in terms of complementarity conditions. In numerical simulation, discrete transitions are almost always handled through embedded logical statements. At the zero-crossing points of some switching function, the initial conditions are updated and the appropriate set of equations is solved restarting at this point in time [1,2]. Systems with so-called Filippov solutions that remain for a while at the zero-crossing require additional analysis. Since they do not pose a particular problem for our approach, we will not discuss them further here. A profound analysis and numerical simulation results of hybrid systems can be found in [3,4]. For optimization tasks, the hybrid simulation can be embedded into a heuristic search algorithm. For instance, an evolutionary algorithm was applied to the start-up of the evaporation system in [5] and particle swarm optimization to the unit commitment problem [6]. These methods suffer from high computational cost when many function evaluations are needed (i.e., in a high dimensional search space). Alternatively one can consider the problem as a constrained optimization problem subject to the dynamic model equations. This leads to a dynamic nonlinear program (NLP). In the so-called direct method, the DAE system is discretized resulting in a large-scale NLP with equality (and possibly inequality) constraints, which can be solved by means of a NLP solver with a gradient-based search. However this NLP-based optimization of hybrid systems is an computationally extremely challenging task due to the non-smoothness of the objective function or constraints which result from instantaneous mode transitions. As a consequence, NLP regularity cannot be presumed and NLP solvers may fail [7]. Essentially three different approaches can be used to overcome this difficulty. Mixed-integer methods have been applied successfully to optimal control problems in [8,9], where a graph search algorithm explores the state space

of the discrete variables. An embedded NLP is used to find the local optima in the continuous state space. The complexity study in [10] indicates that for systems with many decision variables solving the problem becomes computationally expensive. The second approach applied, e.g., in [11,12] comprises sequential optimization methods. Here, the optimization layer exclusively contains continuous variables. The hybrid system is put into the simulation layer and solved by any simulator which is capable to treat discontinuities. Again, the necessity of many simulation runs increases the computational cost. Reformulation strategies, which represent the third class of methods, introduce additional variables and parameters to remove the non-smoothness related to the complementarity conditions from the problem while retaining the system features. Reformulation strategies have been studied in [13,14,15]. Most reformulation strategies fall into one of the following two classes: (i) Relaxation methods transform the complementarities into a set of relaxed equality or inequality constraints, e.g., by the smoothing discussed in this contribution. A sequence of relaxed problems is solved in order to approach the solution of the original problem. (ii) Penalization methods introduce a penalization term into the objective function which measures the violation of the complementarity condition. A comparison of relaxation methods with the heuristic, simulation-based particle swarm optimization regarding the accuracy of the optimization result and the computation time can be found in [16].

3 Model of the Evaporator

The evaporation of volatile components to concentrate non-volatile components within a mixture is a common technology in process engineering. Usually multi-stage systems built up from several identical single evaporators are used. A single evaporator model is considered in this paper following [8].

The system consists of an evaporation tank and a heat exchanger (see Figure 1). The tank is fed through the valve V_1 with a mixture of three liquid components A, B, C with mass fractions w_A, w_B, w_C, where A is a hydrocarbon of high molar mass and thus has a very low vapor pressure (implemented as $P_A^0 = 0$ in the model) compared to water (B) and ethanol (C). Inside the tank, the volatile components are evaporated. Hence the mass fraction of the non-volatile component A in the liquid is increased. This product will be drained from the tank through the valve V_2 when the desired concentration of A is reached. The vapor which consists of B and C with the mass fractions ξ_B, ξ_C determined by the phase equilibrium escapes from the tank through the valve V_{v1}. In order to heat the tank, hot steam is supplied to the heat exchanger, where the steam condensates and leaves the heat exchanger as a liquid.

Depending on the pressure inside the evaporator and the temperature difference between the heat exchanger and the tank, 4 operating modes can be distinguished: If the temperature of the heat exchanger is higher than that of the tank, the heat exchanger operates in the mode 'heating' (H), otherwise 'non-heating' (NH). Inside the tank, the transition from the mode 'non-evaporating' (NE) to the mode 'evaporating' (E) occurs as soon as the pressure reaches a certain threshold. Hence, during operation the system

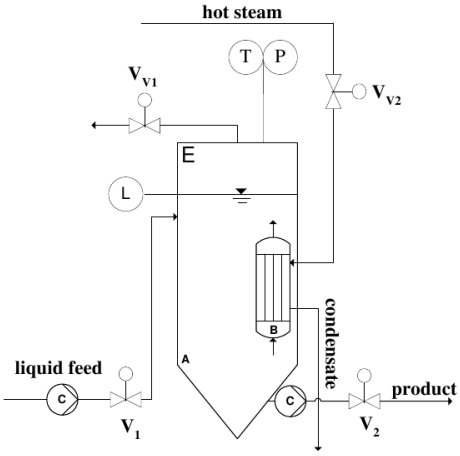


Fig. 1. Evaporator model [8]

may visit the four operating modes: NE/NH ($m = 1$), NE/H ($m = 2$), E/H ($m = 3$) and E/NH ($m = 4$). Thus, the evaporator model possesses the typical features of a hybrid dynamic system with autonomous mode transitions.

3.1 The Hybrid Evaporator Model

The hybrid model of the evaporator consists of $M = 4$ sets of DAEs $f^{(m)}(\dot{x}, x, p) = 0$, $m = 1, ..., M$. The differential equations represent the mass and energy balances inside the evaporator. The analytical expressions change when the pressure acting as a state-dependent switching function $\psi^{(1)}(p_{evap}) = p_{evap} - p_c$, $p_c = 0.4$ bar crosses 0:

$$\dot{m}_i = \begin{cases} F_{in,liq} \cdot w_{i,in} - F_{out,liq} \cdot w_i & \text{if } \psi^{(1)}(p_{evap}) < 0 \\ F_{in,liq} \cdot w_{i,in} - F_{out,liq} \cdot w_i - F_{out,vap} \cdot \xi_i & \text{if } \psi^{(1)}(p_{evap}) \geq 0 \end{cases} \tag{1}$$

$$i = A, B, C$$

$$\dot{U} = \begin{cases} \dot{U}_{in,liq} - \dot{U}_{out,liq} + Q & \text{if } \psi^{(1)}(p_{evap}) < 0 \\ \dot{U}_{in,liq} - \dot{U}_{out,liq} - \dot{U}_{out,vap} - H_{out,vap} + Q & \text{if } \psi^{(1)}(p_{evap}) \geq 0 \end{cases} . \tag{2}$$

F denotes the mass inflow/outflow of the liquid or vapor, respectively. Beside the energy transfer due to in- and outflows of liquid and vapor the energy balance $\dot{U}$ includes the heat transfer Q from the heatexchanger and the evaporation enthalpy $H_{out,vap}$. Algebraic equations for the thermodynamic relations describe the phase equilibrium between the liquid and the vapor components according to each mode. Furthermore, the operation of the heatexchanger switches between the heating and the non-heating mode at the zero-crossing points of the switching function $\psi^{(2)}(T_{heatex}, T_{evap}) = T_{heatex} - T_{evap}$ which leads for the heatexchanger to the conditional expressions

$$Q = \begin{cases} 0 & \text{if } \psi^{(2)}(T_{heatex}, T_{evap}) < 0 \\ \left[c_{p,vap}(\xi_{in}) \cdot T_{in} - c_{p,liq}(\xi_{in}) \cdot T_{heatex} + h_V(\xi_{in}) \right] F_{in,vap} & \\ & \text{if } \psi^{(2)}(T_{heatex}, T_{evap}) \geq 0 \end{cases} \tag{3}$$

$$T_{heatex} = \begin{cases} T_{evap} & \text{if } \psi^{(2)}(T_{heatex}, T_{evap}) < 0 \\ T_{evap} + Q/kA & \text{if } \psi^{(2)}(T_{heatex}, T_{evap}) \geq 0 \end{cases} \tag{4}$$

with the temperature of the incoming steam T_{in}, the temperature inside the heatexchanger T_{heatex}, the mass flow rate of the steam $F_{in,vap}$, the composition of the steam ξ_{in}, the specific evaporation enthalpy h_V, the heat capacities of liquid and vapor, the heat transfer coefficient k and the heating area A. $F_{in,vap}$ is calculated using an approximation of the Bernoulli equation [17].

At the zero-crossing points of $\psi^{(1)}$ and $\psi^{(2)}$, the state variables immediately before the switch x^- have to be mapped onto the state variables immediately after the switch x^+ using the so-called transition functions $x^+ = \mathcal{T}(x^-)$. For instance, for the vapor mass fraction ξ_B the transition function from the non-evaporating modes ($m = 1, 2$) to the evaporating modes ($m = 3, 4$) reads

$$\xi_B^+ = \xi_B^- + \frac{w_B P_B^0(T)}{w_A P_A^0(T) + w_B P_B^0(T) + w_C P_C^0(T)} \tag{5}$$

with the temperature $T^+ = T^- = T$ and the liquid mass fractions $w^+ = w^- = w$.

3.2 A Smooth Evaporator Model

State trajectories are in general non-smooth or even discontinuous at the transition points. If such a model is included into an optimization problem, these points are severe obstacles for gradient-based optimization algorithms. In order to make the optimization of hybrid systems accessible to NLP solvers, the complementarity condition of the original problem is relaxed, i.e., the strict complementarity conditions are fulfilled only approximately. In our smoothing approach, we replace the if-else-statement usually used to implement Eq. (1) and (2) by the smoothing function

$$\varphi(x) = \left(1 + \exp\left[-\frac{\psi(x)}{\tau} \right] \right)^{-1} \tag{6}$$

where $\tau > 0$ is the small smoothing parameter. The model equations are combined in one single set of equations according to

$$\begin{aligned} f(\dot{x}, x, p) = \varphi(x) f^{(1)}(\dot{x}, x, p) \\ + (1 - \varphi(x)) f^{(2)}(\dot{x}, x, p). \end{aligned} \tag{7}$$

Eq. 7 is expected to reproduce the switching behavior of the hybrid model in the limit $\tau \to 0$.

4 Simulation Results

Figure 2 shows that the trajectories of the smooth model with smoothing parameter $\tau = 0.002$ bar and of the original hybrid model deviate only marginally from each other. When the pressure meets the transition condition $p = p_c$ (see inset of Figure 2(a)) the evaporator switches from the non-evaporating mode to the evaporating mode. As a consequence, the mass fractions of the volatile components B and C jump according to Eq. 5 from 0 in the non-evaporating mode (no vapor is present) to the finite values given by the phase equilibrium (see Figure 2(c)) and vapor starts to escape from the tank (Figure 2(d)). The evaporation of the volatile components B and C leads to a decrease of their mass fractions w_B, w_C in the liquid. The decrease of w_C (ethanol) is more pronounced due to the higher vapor pressure and thus the higher outflow of C. Consequently, the vapor mass fractions cross each other near $t = 1150$ s (Figure 2(c)). Since the pressure in the evaporator and also the vapor outflow (Figure 2(a) and 2(d)) depend on the (temperature-dependent) vapor pressure and the mass fractions of all liquid components, both first increase due to the increasing temperature (Figure 2(b)) and later decrease due to the reduced mass fractions w_B, w_C in the liquid.

In the transition region, the dynamics is given by the linear combination (Eq. 7) of both operating modes involved. Figure 3 demonstrates that the smooth model approximates the hybrid model the better the smaller the smoothing parameter τ is chosen: The slope of the state trajectory ξ_C increases and the transition region narrows. The increasing slope of ξ_C is to lead back to the increasing slope of the smoothing function $\varphi(x)$ (Eq. 6). The width of the transition region can be estimated by looking at the variation of the smoothing function

$$\frac{\Delta\varphi}{\Delta t} \approx \frac{\partial\varphi}{\partial\psi}\frac{\partial\psi}{\partial x}\frac{\partial x}{\partial t} \tag{8}$$

linearized around the exact transition point $\psi(x^*) = \psi(x(t^*)) = 0$. With $\Delta\varphi = 1$, the estimate of the width of the transition region will be

$$\Delta t = \frac{4\tau}{\frac{\partial\psi}{\partial x}\big|_{x^*}\frac{\partial x}{\partial t}\big|_{t^*}}\,. \tag{9}$$

For the transition between the non-evaporating and the evaporating mode, this takes the form

$$\Delta t = \frac{4\tau}{\frac{\partial p}{\partial t}\big|_{t^*}} \tag{10}$$

with $\partial p/\partial t = 0.002$ bar/s (see Figure 2(a)). These estimates are shown in Figure 3. The actual transition width obey rather well the predicted ratio $\Delta t_1 : \Delta t_2 : \Delta t_3 = 5 : 2 : 1$. Closely related to the above estimation of the width of the transition region is the argument put forward by us in [18] that the approximate model matches the original model satisfactorily for practical use if

$$\left|\frac{d\varphi}{dt}\right|_{\psi\approx 0} \gg \left|\frac{dx_i}{dt}\right|_{\psi\approx 0} \quad \forall i,\ i = 1...n\,. \tag{11}$$

In this case the influence of the mixture of the modes in the transition region becomes negligible. The derived estimate for appropriate values of τ could be confirmed for the evaporator model.

It is important to note that the trajectories of the hybrid model and the smooth approximation are nearly identical outside the transition region. Obviously, the smoothing only extends the transition time but does not drive the system to a different region of the state space. From these results, we can conclude that our smoothing approach is well suited for the evaporator model.

For a more quantitative analysis of the convergence of the solutions of the relaxed model to that of the original model, we consider in Figure 4 and 5 the average squared deviation

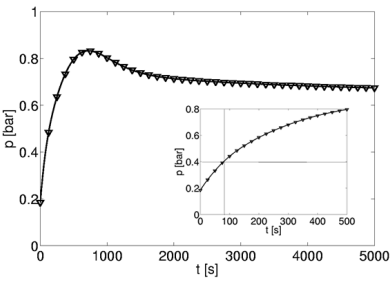
$$s = \frac{1}{N} \sum_{i=1}^{N} \left[\frac{x^{(hybrid)}(t_i) - x^{(smooth)}(t_i)}{\max[x^{(hybrid)}]} \right]^2 \tag{12}$$

for the state variables $x = \xi_C$ (Figure 4), T_{evap} (Figure 5(a)) and p_{evap} (Figure 5(b)) calculated with the hybrid and the smooth model, respectively. For the vapor mass fraction ξ_C the average squared deviation is found numerically to follow approximately $s = \alpha\tau + \epsilon$ with $\alpha = 0.35$ and $\epsilon = 1 \cdot 10^{-4}$. Theoretically we expect $\epsilon = 0$ for sufficiently well behaved functional dependencies and arbitrarily fine discretization ($N \to \infty$). In this case, the dependence of the deviation of the discontinuous state on the smoothing parameter τ is dominated by the finite width of the transition region. For continuous state variables such as T_{evap} and p_{evap} the contribution of the transition region is expected to be small and to vanish superlinearly for $\tau \to 0$. This is confirmed by Figure 5.

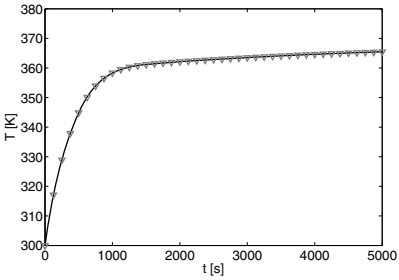
5 Parameter Estimation and Sensitivity Analysis

A fundamental task frequently occurring in process engineering is parameter estimation. Parameter estimation in general aims at extracting the best guesses of the parameters determining the dynamics of the system under consideration based on a series of measurements $x_{ij}^{(m)}$ of several state variables x_i, $i = 1, ..., M$ at different points in time t_j, $j = 1, ..., N$. It is useful to combine the parameter estimation of a hybrid dynamic system with the sensitivity analysis for at least two reasons: First, the sensitivity with respect to the smoothing parameter is needed to predict the suitability of the smooth model for parameter estimation. Second, the sensitivities of the measured state variables with respect to the parameters to be estimated allow to evaluate whether certain data can be used in a specific parameter estimation problem.

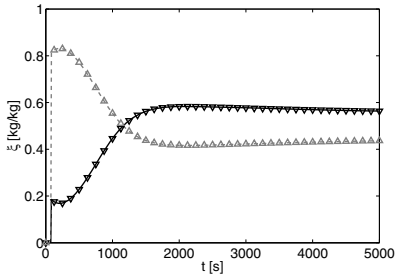
Figure 6 shows the sensitivities of the state variables ξ_C, T_{evap} and p_{evap} with respect to the smoothing parameter τ. The sensitivity of the state variable ξ_C (solid line) quantifies the observation already stated qualitatively in Section 4 that the vapor mass fraction is influenced by the smoothing only in the transition region, i.e., $\frac{d\xi}{d\tau} \approx 0$ outside the transition region. The shape of $\frac{d\xi_C}{d\tau}$ is easily understood in view of the trajectory shown in Figure 3. As τ increases, i.e., $\Delta\tau = \tau_2 - \tau_1 > 0$, the curve $\xi_C(\tau_2)$ lies above that of $\xi_C(\tau_1)$ as long as $p < p_c$ and thus $\Delta\xi_C = \xi_C(\tau_2) - \xi_C(\tau_1)$ is negative, whereas



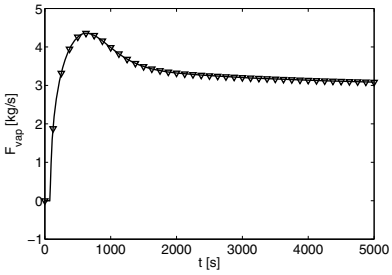
(a) Pressure inside the evaporator for the hybrid model (solid) and the smooth model (triangles).



(b) Temperature inside the evaporator for the hybrid model (solid) and the smooth model (triangles).



(c) Mass fractions of the volatile components B (black, hybrid model: solid, smooth model: triangles) and C (grey, hybrid model: dashed, smooth model: triangles) in the vapor.



(d) Vapor flow from the evaporator for the hybrid model (solid) and the smooth model (triangles).

Fig. 2. Simulation results of the hybrid and the smooth model

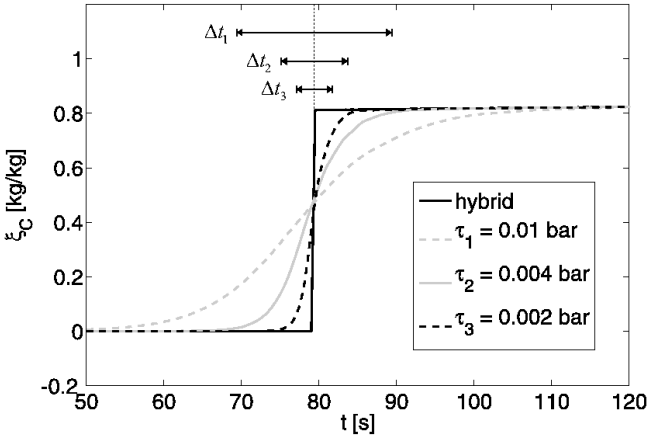


Fig. 3. Mass fraction ξ_C from simulations with several values of the smoothing parameter. Double headed arrows show the width of the transition region as estimated by Eq. 10.

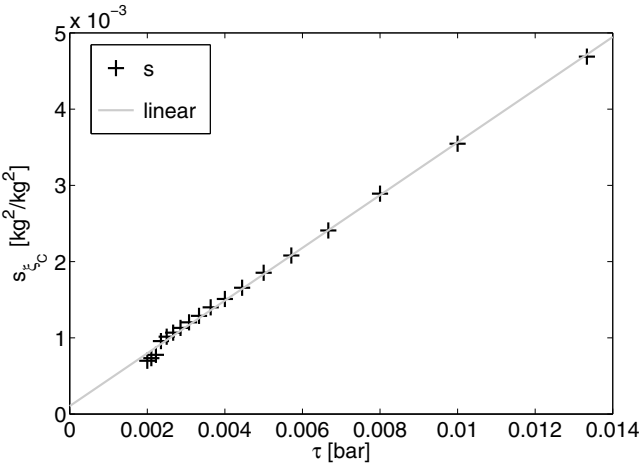
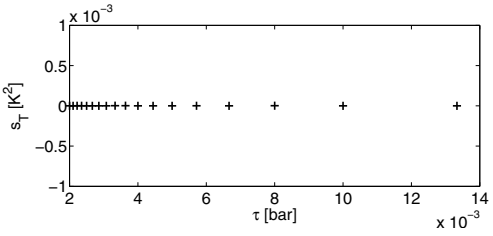
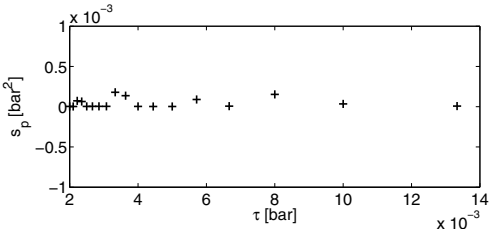


Fig. 4. Deviation between the trajectories of ξ_C calculated with the smooth model and the hybrid model (Eq. 12) as a function of the smoothing parameter



(a) Deviation between the trajectories of T_{evap}.



(b) Deviation between the trajectories of p_{evap}.

Fig. 5. Deviation between the trajectories calculated with the smooth model and the hybrid model (Eq. 12) as a function of the smoothing parameter

it is positive when $p > p_c$. In Figure 6, the sensitivity is calculated at the rather large parameter value $\tau = 0.01$ bar. A smaller τ yields a narrower transition region (not shown). The effect of the smoothing on the the pressure (dotdashed line) and the temperature (dashed line) can be neglected in accordance with the findings reported above in Figure 5.

As an illustrative example of parameter estimation we will estimate below the coefficient k of the heat transfer from the heat exchanger to the tank, the valve throughputs

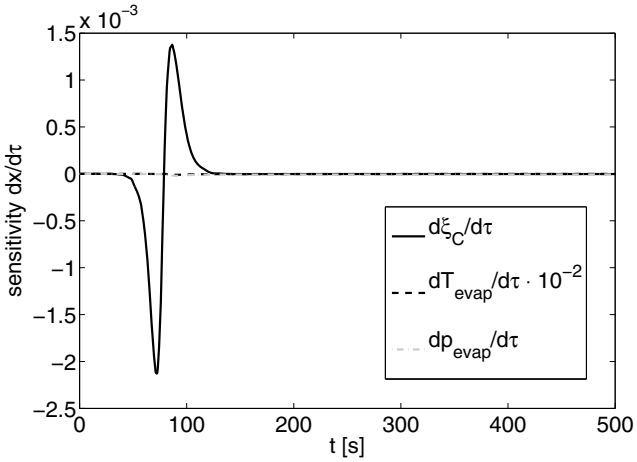


Fig. 6. Sensitivities of the states ξ_C (solid), T_{evap} (dashed) and p_{evap} (grey dotdashed) with respect to the smoothing parameter

of the liquid inflow (V_1) and the liquid outflow (V_2) based on measurements of several state variables (ξ_C, T_{evap} and p_{evap}). The vapor mass fraction ξ_C will be chosen as measured variable due to the fact that ξ_C is one of the quantities with the most significant hybrid behavior, which shows up as a jump at the mode transition (Figure 2(c)). The temperature and the pressure are chosen since they can be measured easily in real life. As can be seen in Figures 7 and 8, the sensitivities of ξ_C are relatively large around the mode transition. The most dominant influence of the heat transfer coefficient k and the liquid inflow V_1, i.e., the mass of liquid to be heated, is indirect via the transition time from NE to E. An increase of the heat transfer coefficient or a decrease of the liquid inflow shifts the curve $\xi_C(t)$ to the left, i.e., accelerates the process. Hence the respective sensitivities are positive ($d\xi_C/dk$) and negative ($d\xi_C/dV_1$) in the first time period and change sign when $\xi_C(t)$ exceeds its maximum. The sensitivities of the pressure and the temperature with respect to the model parameters k and V_1 become more and more important with time and reach higher absolute values than the sensitivity of the vapor mass fraction ξ_C outside the transition region. The sensitivities of all three state variables with respect to the liquid outflow have opposite sign compared to that of the liquid inflow and the absolute values are smaller (not shown). Based on Figures 7 and 8, one can expect to find the correct parameter values by means of parameter estimation, if ξ_C, T_{evap} or p_{evap} measurement data are available.

The 'measurement data' for our parameter estimation have been generated by simulation of the hybrid system ($k^{(sim)} = 4.0\,kW/m^2K$, $V_1^{(sim)} = 7.0\,kg/s$ and $V_2^{(sim)} = 2.0\,kg/s$) with added Gaussian-distributed measurement error. We use series of 51 equidistant data points within the time horizon $t \in [0, 500]$ s. As usual, the parameter estimation problem is formulated as least-square optimization.

Table 1 shows the results of the parameter estimation based on different data sets. The heat transfer coefficient can be determined accurately using measurements of T_{evap} and p_{evap} whereas the ξ_C data do not lead to such a good estimate. The use of the two

Table 1. Parameter estimation result for different series of measurement

Parameter	$\xi_C^{(meas)}$	$T_{evap}^{(meas)}$	$p_{evap}^{(meas)}$	$T_{evap}^{(meas)}, p_{evap}^{(meas)}$
V_1	8.50	7.15	6.45	6.23
V_2	1.96	2.42	0.12	1.49
k	3.50	3.97	4.04	3.93

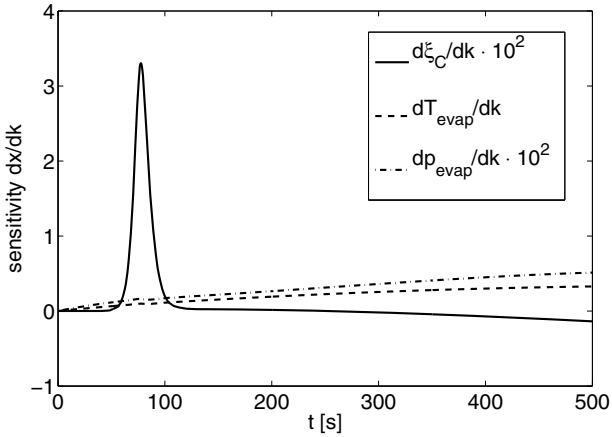


Fig. 7. Sensitivities of the states ξ_C (solid), T_{evap} (dashed) and p_{evap} (dotdashed) with respect to the heat transfer coefficient

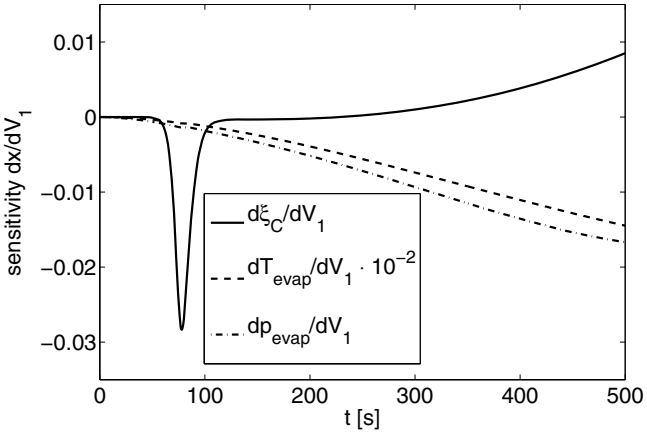


Fig. 8. Sensitivities of the states ξ_C (solid), T_{evap} (dashed) and p_{evap} (dotdashed) with respect to the valve throughputs

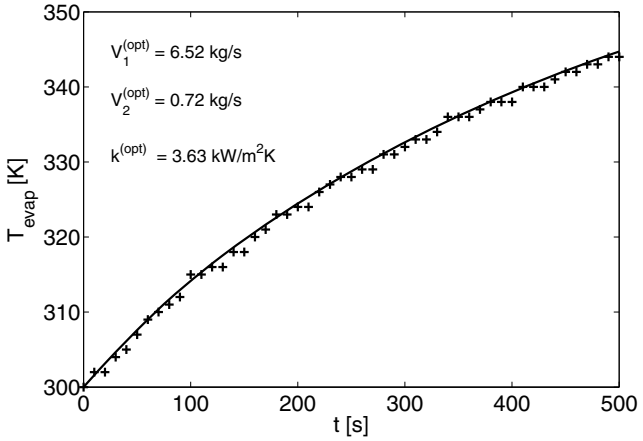


Fig. 9. Parameter estimation result (solid) and measurement (cross)

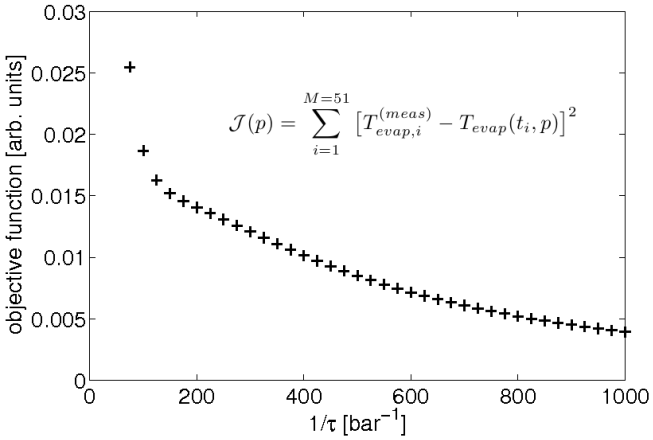


Fig. 10. Objective function values at the solution as a function of the smoothing parameter

series T_{evap} and p_{evap} together does not improve the result. Figure 9 shows the optimization result for a particular realization of 'measurement data' of the temperature with $\sigma = 0.5\,K$. The optimal parameter values are found to be $k^{(opt)} = 3.63\,kW/m^2K$, $V_1^{(opt)} = 6.52\,kg/s$ and $V_2^{(opt)} = 0.72\,kg/s$.

Finally, we investigate the dependence of the optimization result on the smoothing parameter based on the measurement of the temperature. In order to avoid the interference with other effects, the measurement error is set to $\sigma = 0$ and only the valve for the liquid inflow V_1 is estimated. In this situation the parameter value is found very precisely ($V_1 = 6.98\,kg/s$). The objective function values at the solution clearly decrease

with the smoothing parameter τ (see Figure 10) even though the continuous trajectory of T_{evap} is only marginally modified by the smoothing method. Similarly and importantly, the optimal parameters estimated are quite robust regarding the smoothing parameter.

6 Conclusions

In summary, we carried out the parameter estimation accompanied by sensitivity analysis for a hybrid evaporating system using a smoothing approach. We first investigated a smooth approximated model by means of simulation of the evaporator dynamics for different values of the smoothing parameter. Performing the sensitivity analysis with respect to the parameters to be estimated we could evaluate the usability of the measurement of a certain variable for the parameter estimation. The sensitivity with respect to the smoothing parameter was studied to evaluate the suitability of the smooth model for the purpose of parameter estimation. The proposed method allowed to successfully identify the parameter values. The results turned out to be quite robust against the variation of the smoothing parameter. In the future, the extension of the model will be made to optimize the operations of the evaporation system.

References

1. Barton, P.I., Lee, C.K.: Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. ACMT. Model. Comp. S. 12, 256–289 (2002)
2. Bahl, V., Linninger, A.A.: Modeling of Continuous-Discrete Processes. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 387–402. Springer, Heidelberg (2001)
3. Mehrmann, V., Wunderlich, L.: Hybrid systems of differential-algebraic equations - analysis and numerical solution. J. Process Contr. 19, 1218–1228 (2009)
4. Goebel, R., Sanfelice, R.G., Teel, A.R.: Hybrid dynamical systems. IEEE Control Syst. Mag., 28–93 (2009)
5. Sonntag, C., Su, W., Stursberg, O., Engell, S.: Optimized start-up control of an industrial-scale evaporation system with hybrid dynamics. Control Eng. Pract. 16, 976–990 (2008)
6. Pappala, V.S., Erlich, I.: A new approach for solving the unit commitment problem by adaptive particle swarm optimization. In: 2008 IEEE Power and Energy Society General Meeting, pp. 1–6 (2008)
7. Biegler, L.T.: Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes. SIAM and MOS (2010)
8. Sonntag, C., Stursberg, O., Engell, S.: Dynamic optimization of an industrial evaporator using graph search with embedded nonlinear programming. In: 2nd IFAC Conference on Analysis and Design of Hybrid Systems, pp. 211–216 (2006)
9. Barton, P.I., Lee, C.K., Yunt, M.: Optimization of hybrid systems. Comp. Chem. Eng. 30, 1576–1589 (2006)
10. Till, J., Engell, S., Panek, S., Stursberg, O.: Applied hybrid system optimization: An empirical investigation of complexity. Control Eng. Pract. 12, 1291–1303 (2004)
11. de Prada, C., Cristea, S., Rosano, J.J.: Optimal start-up of an evaporation station. In: 8th International IFAC Symposium on Dynamics and Control of Process Systems, vol. 3, pp. 115–120 (2007)

12. Voelker, A., Sonntag, C., Lohmann, S., Engell, S.: Optimization-based safety analysis of an industrial-scale evaporation system with hybrid dynamics. In: 8th International IFAC Symposium on Dynamics and Control of Process Systems, vol. 1, pp. 117–122 (2007)
13. Baumrucker, B.T., Renfro, J.G., Biegler, L.T.: MPEC problem formulations and solution strategies with chemical engineering applications. Comp. Chem. Eng. 32, 2903–2913 (2008)
14. Sager, S.: Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control. J. Process Contr. 19, 1238–1247 (2009)
15. Ralph, D., Wright, S.J.: Some properties of regularization and penalization schemes for MPECs. Optim. Method. Softw. 19, 527–556 (2004)
16. Mynttinen, I., Li, P.: A reformulation scheme for parameter estimation of hybrid systems. In: 21st European Symposium on Computer-Aided Process Engineering, pp. 778–782 (2011)
17. Sonntag, C., Stursberg, O.: Safety verification of a discretely controlled evaporation system. Technical Report, HYCON, pp. 1–20 (2005)
18. Mynttinen, I., Hoffmann, A., Runge, E., Li, P.: Reformulation strategies for optimization of hybrid dynamic systems (2011) (submitted)

High Detailed Lava Flows Hazard Maps by a Cellular Automata Approach

William Spataro[1,⋆] , Rocco Rongo[2], Valeria Lupiano[2], Maria Vittoria Avolio[1], Donato D'Ambrosio[1] and Giuseppe A. Trunfio[3]

[1] Department of Mathematics and High Performance Computing Center, University of Calabria, via Pietro Bucci, I-87036 Rende, Italy
spataro@unical.it
[2] Department of Earth Sciences and High Performance Computing Center, University of Calabria, Italy
[3] Department of Architecture, Planning and Design, University of Sassari, Italy

Abstract. The determination of areas exposed to be interested by new eruptive events in volcanic regions is crucial for diminishing consequences in terms of human causalities and damages of material properties. In this paper, we illustrate a methodology for defining flexible high-detailed lava invasion hazard maps. Specific scenarios can be extracted at any time from the simulation database, for land-use and civil defence planning in the long-term, to quantify, in real-time, the impact of an imminent eruption, and to assess the efficiency of protective measures. Practical applications referred to some inhabited areas of Mt Etna (South Italy), Europe's most active volcano, show the methodology's appropriateness in this field.

Keywords: Cellular automata, Lava flows simulation, Hazard maps, Land use planning, Mt Etna.

1 Introduction

The use of thematic maps of volcanic hazard is of fundamental relevance to support policy managers and administrators in taking the most correct land use planning and proper actions that are required during an emergency phase. In particular, hazard maps are a key tool for emergency management, describing the threat that can be expected at a certain location for future eruptions. At Mt. Etna (Sicily, Italy), the most active volcano in Europe, the majority of events occurred in the last four centuries report damage to human properties in numerous towns on the volcano flanks [1]. In last decades, the risk of the Etnean area has increased due to continued urbanization [2], with the consequence that new eruptions may involve even greater risks. Different countermeasures based on embankments or channels were adopted in recent crises to stop or deflect lava ([3], [4]). However, such kinds of interventions are generally performed while the eruption is in progress, with the consequence of both not guarantying their effectiveness, besides inevitably putting into danger the safety of involved persons. For the purpose of individuating affected areas in advance, one response to such challenges is the numerical

⋆ Corresponding author.

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 85–99.
DOI: 10.1007/978-3-642-34336-0_6

simulation of lava flows (e.g., [5], [6], [7]) or the simple paths followed by future lava flows (e.g., [8], [9]). For instance, in 2001 the path of the eruption that threatened the town of Nicolosi on Mt Etna was correctly predicted by means of a lava flows simulation model [10], providing at that time useful information to local Civil Defense authorities. However, in order to be efficiently and correctly applied, the above approaches require an a priori knowledge of the degree of exposure of the volcano surrounding areas, to allow both the realization of preventive countermeasures, and a more rational land use planning. In the following, we illustrate a methodology for the definition of flexible high-resolution lava invasion hazard maps, based on an improved version of SCIARA, a reliable and efficient Cellular Automata lava flow model, and show some specific applications related to inhabited areas of Mt Etna, which demonstrate the validity of the application for civil defense purposes and land use planning.

2 Cellular Automata and the SCIARA Model for Lava Flows Simulation

The behavior of lava flows is difficult to be dealt with using traditional methods based on differential equation systems (e.g., cf. [11], [12], [13]). In fact, due to the complexities of its rheology, lava can range from fluids approximating Newtonian liquids to brittle solids while cooling, and thus it is difficult to solve the differential equations without making some simplifications. Nevertheless, many attempts of modelling real cases can be found in literature.

In order to be applied for land use planning and civil defense purposes in volcanic regions, a computational model for simulating lava flows should be well calibrated and validated against test cases to assess its reliability, cf. e.g. ([6], [14], [15]). Another desirable characteristic should be the model's efficiency since, depending on the extent of the considered area, a great number of simulations could be required ([16], [17]). A first computational model of basaltic lava flows, based on the Cellular Automata computational paradigm and, specifically, on the Macroscopic Cellular Automata approach for the modeling of spatially extended dynamical systems, was proposed in [18] called SCIARA. In the following years, the SCIARA family of lava flows simulation models have been improved and applied with success to the simulation of different Etnean cases of study, e.g. [10], [14].

Cellular Automata (CA) [19] were introduced in 1947 by Hungarian-born American mathematician John von Neumann in his attempt to understand and formalise the underling mechanisms that regulate the auto-reproduction of living beings. While initially studied from a theoretical point of view, CA are continuously gaining researchers attention also for their range of applicability in different fields, such as Physics, Biology, Earth Sciences and Engineering. However, researchers' major interest for CA regard their use as powerful parallel computational models and as convenient tools for modelling and simulating several types of complex physical phenomena (e.g. [7], [10], [20], [21], [22]).

Classical Cellular Automata can be viewed as an n-dimensional space, R, subdivided in cells of uniforms shape and size. Each cell embeds an identical finite automaton (fa), whose state accounts for the temporary features of the cell; Q is the finite set of states.

The fa input is given by the states of a set of neighbouring cells, including the central cell itself. The neighbourhood conditions are determined by a geometrical pattern, X, which is invariant in time and space. The fa have an identical state transition function $\tau : Q^{\sharp X} \rightarrow Q$, where $\sharp X$ is the cardinality of the set of neighbouring cells, which is simultaneously applied to each cell. At step $t = 0$, fa are in arbitrary states and the CA evolves by changing the state of all fa simultaneously at discrete times, according to τ.

While Cellular Automata models are suitable for describing fluid-dynamics from a microscopic point of view (e.g., Lattice Boltzmann models - [23]), many natural phenomena generally evolve on very large areas, thus needing a macroscopic level of description. Moreover, they may be also difficult to be modelled through standard approaches, such as differential equations [24], and Macroscopic Cellular Automata (MCA) [25] can represent a valid alternative. Macroscopic Cellular Automata (MCA) introduce some extensions to the classical CA formal definition. In particular, the Q of state of the cell is decomposed in r substates, Q_1, Q_2,..., Q_r, each one representing a particular feature of the phenomenon to be modelled (e.g. for lava flow models, cell temperature, lava content, outflows, etc). The overall state of the cell is thus obtained as the Cartesian product of the considered substates: $Q = Q_1 \times Q_2 \times ... \times Q_r$. A set of parameters, $P = \{p_1, p_2, ..., p_p\}$, is furthermore considered, which allow to "tune"the model for reproducing different dynamical behaviours of the phenomenon of interest (e.g. for lava flow models, the Stephan-Boltzmann constant, lava density, lava solidification temperature, etc). As the set of state is split in substates, also the state transition function τ is split in elementary processes, $\tau_1, \tau_2, ..., \tau_s$, each one describing a particular aspect that rules the dynamic of the considered phenomenon. Eventually, $G \subset R$ is a subset of the cellular space that is subject to *external influences* (e.g. for lava flow models, the crater cells), specified by the supplementary function γ. External influences are introduced in order to model features which are not easy to be described in terms of local interactions.

In the MCA approach, by opportunely discretizing the surface on which the phenomenon evolves, the dynamics of the system can be described in terms of flows of some quantity from one cell to the neighbouring ones. Moreover, as the cell dimension is a constant value throughout the cellular space, it is possible to consider characteristics of the cell (i.e. substates), typically expressed in terms of volume (e.g. lava volume), in terms of thickness. This simple assumption permits to adopt a straightforward but efficacious strategy that computes outflows from the central cell to the neighbouring ones in order to minimize the non-equilibrium conditions. Still, owing to their intrinsic parallelism, both CA and MCA models implementation can be easily and efficaciously implemented on parallel computers, and the simulation duration can be reduced almost proportionally to the number of available processors ([26], [27]).

In this work, the latest release of the SCIARA Cellular Automata model for simulating lava flows was adopted. Specifically, a Bingham-like rheology has been introduced for the first time as part of the Minimization Algorithm of the Differences [25], which is applied for computing lava outflows from the generic cell towards its neighbors. Besides, the hexagonal cellular space adopted in the previous releases [10] of the model for mitigating the anisotropic flow direction problem has been replaced by a square one, nevertheless by producing an even better solution for the anisotropic effect. The model

has been calibrated by considering three important real cases of studies, the 1981, 2001 and 2006 lava flows at Mt Etna (Italy), and on ideal surfaces in order to evaluate the magnitude of anisotropic effects. Even if major details of this advanced model can be found in [28], we briefly outline it's main specifications.

In formal terms, the SCIARA MCA model is defined as:

$$SCIARA =< R, L, X, Q, P, \tau, \gamma > \tag{1}$$

where:

- R is the set of square cells covering the bi-dimensional finite region where the phenomenon evolves;
- $L \subset R$ specifies the lava source cells (i.e. craters);
- $X = \{(0,0),(0,1),(-1,0),(1,0),(0,-1),(-1,1),(-1,-1),(1,-1),(1,1)\}$ identifies the pattern of cells (Moore neighbourhood) that influence the cell state change, referred to cells by indexes 0 (for the central cell) through 8;
- $Q = Q_z \times Q_h \times Q_T \times Q_f^8$ is the finite set of states, considered as Cartesian product of "substates". Their meanings are: cell elevation a.s.l. (above sea level), cell lava thickness, cell lava temperature, and lava thickness outflows (from the central cell toward the eight adjacent cells), respectively;
- $P = \{w, t, T_{sol}, T_{vent}, r_{Tsol}, r_{Tvent}, hc_{Tsol}, hc_{Tvent}, \delta, \rho, \epsilon, \sigma, c_\nu\}$ is the finite set of parameters (invariant in time and space) which affect the transition function (please refer to [28] for their specifications);
- $\tau : Q^9 \rightarrow Q$ is the cell deterministic transition function, applied to each cell at each time step, which describes the dynamics of lava flows, such as cooling, solidification and lava outflows from the central cell towards neighbouring ones;
- $\gamma : Q_h \times \mathbb{N} \rightarrow Q_h$ specifies the emitted lava thickness, h, from the source cells at each step $k \in \mathbb{N}$ ($\mathbb{N}$ is the set of natural numbers).

As stated before, the new SCIARA model introduces a rheology inspired to the Bingham model and therefore the concepts of critical height and viscosity are explicitly considered ([29], [11]). In particular, lava can flow out from a cell towards its neighbours if and only if its thickness overcomes a critical value (i.e. the critical height), so that the basal stress exceeds the yield strength. Moreover, viscosity is accounted in terms of flow relaxation rate, r, a the parameter of the distribution algorithm that influences the amount of lava that actually leaves the cell, according to a power law of the kind:

$$\log r = a + bT \tag{2}$$

where T is the lava temperature and a and b coefficients determined by solving the system:

$$\begin{cases} \log r_{Tsol} = a + bT_{sol} \\ \log r_{Tvent} = a + bT_{vent} \end{cases}$$

where T_{sol} and T_{vent} are the lava temperature at solidification and at the vents, respectively. Similarly, the critical height, hc, mainly depends on lava temperature according to a power law of the kind:

$$\log hc = c + dT \tag{3}$$

whose coefficients c and d are obtained by solving the system:

$$\begin{cases} \log hc_{Tsol} = c + dT_{sol} \\ \log hc_{Tvent} = c + dT_{vent} \end{cases}$$

It is known that, in general, deterministic CA for the simulation of macroscopic fluids present a strong dependence on the cell geometry and directions of the cellular space. In order to solve the problem, different solutions have been proposed in literature, such as the adoption of hexagonal cells ([10], [30], [31]) or Monte Carlo approaches ([32], [15]). The first solution, however, does not perfectly solve the problem on ideal surfaces, while the second one has the disadvantage of giving rise to non-deterministic simulation models. In order to solve the anisotropic problem, which is typical of deterministic Cellular Automata models for fluids on ideal surfaces, a fictitious topographic alteration along diagonal cells is considered with respect to those "individuated" by the DEM (Digital Elevation Model). As a matter of fact, in a standard situation of non-altered heights, cells along diagonals result in a lower elevation with respect to the remaining ones (which belong to the von Neumann neighborhood), even in case of constant slope. This is due since the distance between the central cell and diagonal neighbors is greater than of the distance between the central cell and orthogonal adjacent cells (cf. Figure 1). This introduces a side effect in the distribution algorithm, which operates on the basis of height differences. If the algorithm perceives a greater difference along diagonals, it will erroneously privilege them by producing greater outflows. In order to solve this problem, we consider the height of diagonal neighbors taken at the intersection between the diagonal line and the circle with radius equal to the cell side and centered in the central cell, so that the distance with respect to the centre of the central cell is constant for each cell of the Moore neighbourhood (Figure 1). Under the commonly assumed hypothesis of inclined plane between adjacent cells [15], this solution permits to have constant differences in level in correspondence of constant slopes, and the distribution algorithm can work "properly". Refer to [28] for other specifications on this issue.

3 A Methodology for Creating Hazard Maps

Volcanic hazard maps are fundamental for determining locations that are subject to eruptions and their related risk. Typically, a volcanic hazard map divides the volcanic area into a certain number of zones that are differently classified on the basis of the probability of being interested by a specific volcanic event in future. Mapping both the physical threat and the exposure and vulnerability of people and material properties to volcanic hazards can help local authorities to guide decisions about where to locate critical infrastructures (e.g. hospitals, power plants, railroads, etc) and human settlements and to devise mitigation measures that might be appropriate. This could be useful for avoiding the development of inhabited areas in high risk areas, thus controlling land use planning decisions.

While a reliable simulation model is certainly a valid instrument for analyzing volcanic risk in a certain area by simulating possible *single* episodes with different vent locations, e.g. [33], the methodology for defining high detailed hazard maps here presented is based on the application of the SCIARA lava flows computational model for simulating an *elevated* number of new events on topographic data. In particular, the

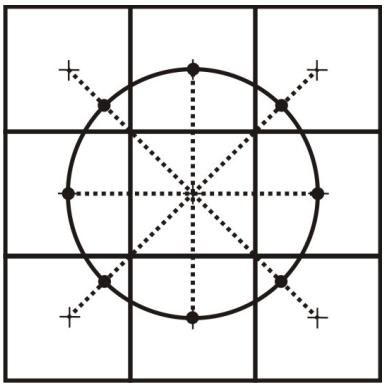


Fig. 1. Reference schema for cells altitude determination in the Moore neighbourhood. Altitudes of cells belonging to the von Neumann neighbourhood correspond to normal DEM values, while those along diagonals are taken at the intersection between the diagonal line and the circle with radius equal to the cell side, so that the distance with respect to the centre of the central cell is constant for each adjacent neighbour.

methodology requires the analysis of the past behavior of the volcano, for the purpose of classifying the events that historically interested the region. In such a way, a meaningful database of plausible simulated lava flows can be obtained, by characterizing the study area both in terms of areal coverage, and lava flows typologies. Once the simulation database has been completed (i.e., an adequate, usually elevated, number of simulations have been carried out), data is processed by considering a proper criterion of evaluation. A first solution could simply consist in considering lava flows overlapping, by assigning a greater hazard to those sites interested by a higher number of simulations. However, a similar choice could be misleading. In fact, depending on their particular traits (e.g., location of the main crater, duration and amount of emitted lava, or effusion rate trend), different events can occur with different probabilities, which should be taken into account in evaluating the actual contribution of performed simulations with respect to the definition of the overall hazard of the study area. In most cases, such probabilities can be properly inferred from the statistical analysis of past eruptions, allowing for the definition of a more refined evaluation criterion. Accordingly, in spite of a simple hitting frequency, a measure of lava invasion hazard can be obtained in probabilistic terms. In the following, we show how such approach was applied to Mt Etna.

3.1 Application of the Methodology to the Mt. Etna Volcano Area

By adopting a procedure well described in [17] and [34], which referred to the Eastern sector of Mt. Etna and applied by employing a previous version of the SCIARA CA model, we here show the application to the entire area of the volcano using the new SCIARA model described in Section 2. Firstly, based on documented past behavior of the volcano, the probability of new vents forming was determined, resulting in a characterization (thus, a Probability Density Function - PDF - map) of the study region

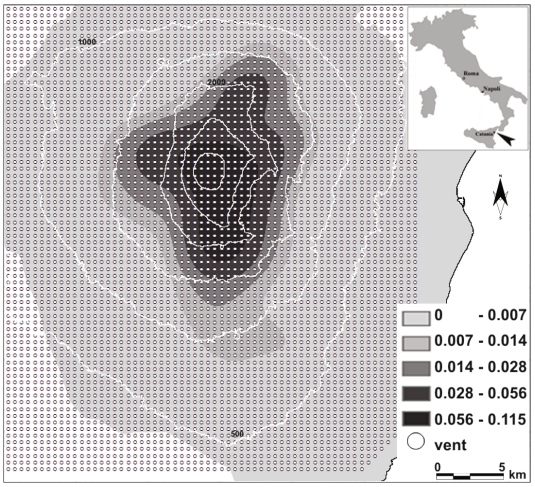


Fig. 2. The characterization of new vents forming of the study region on the basis of historical data (see text), representing different probabilities of activation, considered in this work, together with the grid of 4290 hypothetical vents defined as the source for the simulations to be carried

into areas (Figure 2), that represent different probabilities of new vents opening [35], assessed by employing a Poisson distribution which considers a spatial density and a temporal component. The spatial probability density function was estimated through a Gaussian kernel by considering the main volcanic structures at Mt Etna, while the temporal rate was evaluated by using an approach based on the "repose-time method" [36].

Subsequently, all flank eruptions of Etna since 1600 AD were classified according to duration and lava volume [17] and a representative effusion rate trend taken into account in order to characterize lava temporal distribution for the considered representative eruptions, basically reflecting the effusive mean behavior of Etnean lava flows. In fact, with the exception of few isolated cases, a typical effusive behavior was strongly evidenced by the analysis of the volcano past activity [37]. As a consequence, it is not a hasty judgment to suppose that such behavior will not dramatically change in the near future and thus that the SCIARA lava flows simulation model, calibrated and validated on a set of effusive eruptions, be adequate for simulating new events on Mt Etna. An overall probability of occurrence, p_e, was thus defined for each scenario, by considering the product of the individual probabilities of its main parameters:

$$p_e = p_s \cdot p_c \cdot p_t \tag{4}$$

where p_s denotes the probability of eruption from a given location (i.e., based on the PDF map), p_c the probability related to the event's membership class (i.e., emitted lava and duration), and p_t the probability related to its effusion rate trend.

Once representative lava flows were devised as above, a set of simulations were planned to be executed in the study area by means of the SCIARA lava flows simulation model. At this purpose, a grid composed by 4290 craters, equally spaced by $500m$,

was defined on the considered as a covering for Mt Etna, as shown in Figure 2. This choice allowed to both adequately and uniformly cover the study area, besides considering a relatively small number of craters. Specifically, a subset of event classes which define 6 different effusion rates probabilities, derived from historical events considered in [17], were taken into account for each crater, thus resulting in a total of 25740 different simulations to be carried out. Owing to the elevated number of SCIARA simulations to be carried out, thanks to the adoption of Parallel Computing each scenario was simulated for each of the vents of the grid. Simulations were performed on an 80-node Apple Xserve Xeon-based cluster and were performed in ca. 10 days.

Lava flow hazard was then punctually evaluated by considering the contributions of all the simulations which affected a generic cell in terms of their probability of occurrence. formally, if a given DEM cell (and thus, a CA cell) of co-ordinates (x, y) is affected by $n_{x,y} \leq N$ simulations, being N the overall number of performed simulations, its hazard $h_{x,y}$ can be defined as the sum of the probabilities of occurrence of involved lava flows, $p_e(i)$ (i=1, 2, ..., $n_{x,y}$):

$$h_{x,y} = \sum_{i=1}^{n_{x,y}} p_e(i) \tag{5}$$

The obtained lava flow hazard map resulting from these simulations and the application of equation 5 is presented in Figure 3, and represents the probability that future eruptions will affect the entire Etnean area.

Importantly, the methodology for the compilation of lava flows invasion hazard maps here proposed provides for, as integrant part, a process for the verification of results. A validation procedure was thus contemplated for the produced hazard map, consisting in a technique which produces statistical indicators on which one can quantify the reliability of the results and, therefore, assess whether the final product can be confidently used for Civil Protection, for example, for setting in safety particularly vulnerable areas, and for Land Use Planning. Refer to [17] for major details on the methodology validation process.

4 Applications for Civil Defense and Land Use Planning

As shown previously, the described methodology permits the definition of general hazard maps, as the one reported in Figure 3, which can give valuable information to Civil Defense responsible authorities. However, further, more specialized applications can be devised by considering that the SCIARA simulation model is integrated in a GIS Geographic Information System) application that permits to take also into account the effects of “virtual” embankments, channels, barriers, etc. In particular, the availability of a large number of lava flows of different eruption types, magnitudes and locations simulated for this study allows the instantaneous extraction of various scenarios. This is especially relevant once premonitory signs indicate the possible site of imminent eruptions, and thus permitting to consider hazard circumscribed to a smaller area. An important Civil Defense oriented application regards the possibility to identify all source areas of lava flows that are capable of affecting a given area of interest, such as a town or

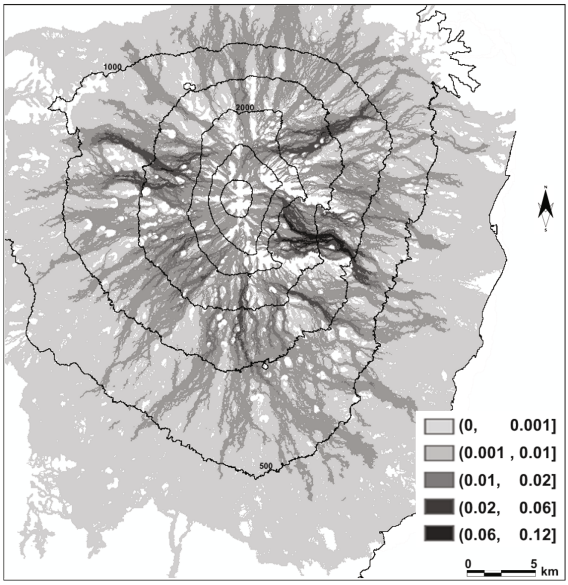


Fig. 3. Hazard map of the study area based on the 25740 simulations. As a compromise between map readability and accuracy, 5 classes are reported (grey colouring), in increasing order of susceptibility (probability of lava invasion).

a major infrastructure. In this case, this application is rapidly accomplished by querying the simulation database, by selecting the lava flows that affect the area of interest and by circumscribing their sources. For this application we have chosen the town of Belpasso, an important historical and cultural site, with many administrative buildings and tourist facilities. Figure 4 show vents which can originate eruptions capable of affecting the urban areas of Belpasso, together with the resulting hazard scenario, allowing to immediately assess the threat posed by an eruption exclusively on the basis of its source location. While the previous application localizes craters that can originate events that may interest an inhabited area, the one reported in Figure 5 can have even more impact in land use planning, referred for the entire town district of Zafferana Etnea, another important inhabited area of the volcano. This application is fundamental in understanding how local authorities can plan the future development of the city, avoiding it in elevated risk areas. Specifically, the figure shows how several areas of the entire municipality are at risk, especially to the North-West and South.

Etnean eruptions can even comprise complex events, which for Etna are fairly typical, such as lava emission from an extensive system of eruptive fissures propagating downslope over a length of several kilometers. We have performed an analysis for such an eruption on the east-northeast flank of Etna, not far from the 1928 eruption site, with lava emission from a fissure system about 7km long. The eruptive system was approximated by a subset of vents of the simulation grid and all lava flows originated from

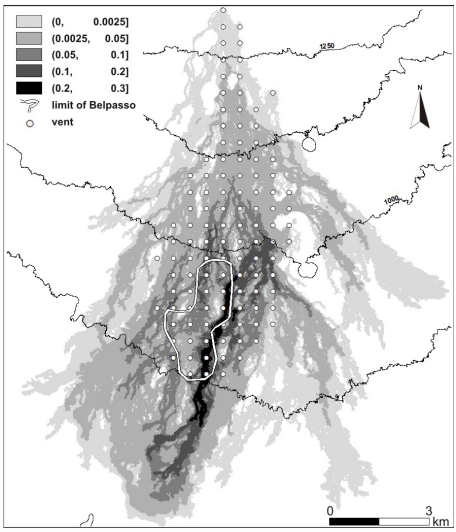


Fig. 4. Map showing vents, belonging to the simulation grid of Fig. 2, which can produce eruptions capable of affecting the urban area of the town of Belpasso, together with the resulting susceptibility scenario, allowing to immediately assess the threat posed by an eruption exclusively on the basis of its source location

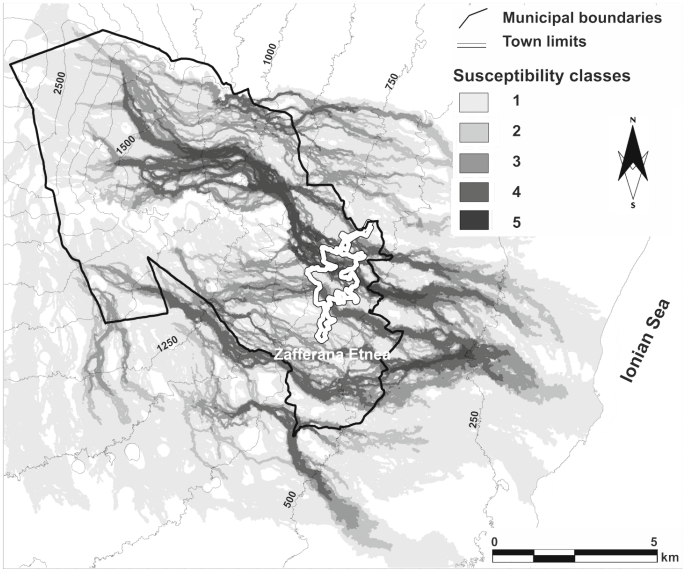


Fig. 5. A second example of application of hazard zonation referred to the entire town district of Zafferana Etnea. The town district boundaries are indicated by the black line, while the present inhabited area with white line. As shown, the majority of the municipal area is at risk.

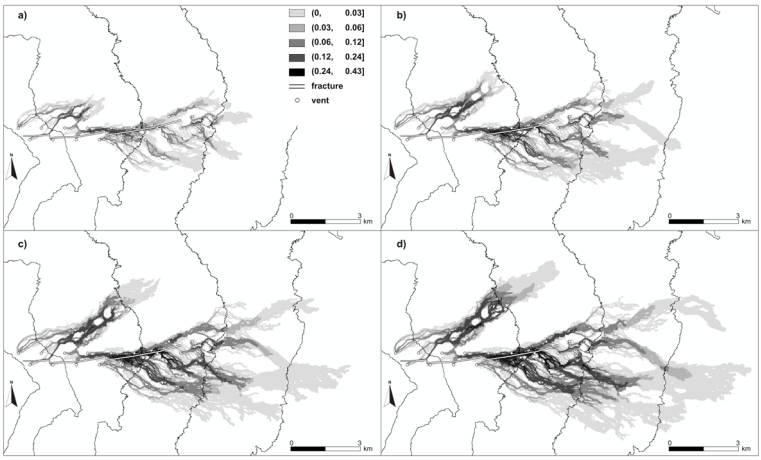


Fig. 6. A hypothetical hazard scenario for a system of eruptive fissures propagating downslope over a length of 7 km on the east-northeast flank of Mount Etna. The figure refers to lava hazard of the area after 1, 3, 6 and 9 days (a, b, c, d), respectively. Note that the same scenarios can be considered as temporal scenario for the area.

them selected from the simulation database (without needing to perform new simulations). For this application, the resulting map is shown in Figure 6. The same figure refers also to a further application regarding temporal hazard mapping, by evidencing the evolution of the involved area in time. This application could be of fundamental importance for assessing, from a temporal point of view, how hazard of a specific area evolves in time (e.g. day by day), so that more specific countermeasures can be considered by responsible authorities.

In particular, Figure 6 also shows the result relative to 1, 3, 6 and 9 days respectively, of the invaded areas, with relative probability values of occurrence, in the case of the activation of the considered fissure system. This application regards a real-time assessment of lava invasion in confined areas, since the produced map indicates a temporal evolution of hazard, in terms of probability, which can be useful in case of an imminent/new event to Civil Protection to monitor, and eventually intervene, in areas with higher values of lava hazard, without having information on the event's duration and emission rate that take place.

A further fundamental application category regards the assessment of protective measures, such as earth barriers or channel digging, for mitigating lava invasion susceptibility in given areas. To illustrate this kind of application, a northwest-southeast trending barrier, 2 km long and 20 m high, was considered along the northern margin of Nicolosi, an urban area with many administrative buildings and tourist facilities for diverting lava flows into a valley at the eastern margin of the town without, however, considering the legal and ethical aspects of such an operation. By querying the simulation database, all the lava flows that affected the barrier were selected and thus re-simulated on the modified topography which embeds the presence of the barrier. Similarly to the case of the applications shown in Figures 4 and 5, an ad hoc susceptibility scenario was extracted by considering these new simulations (Figure 7a).

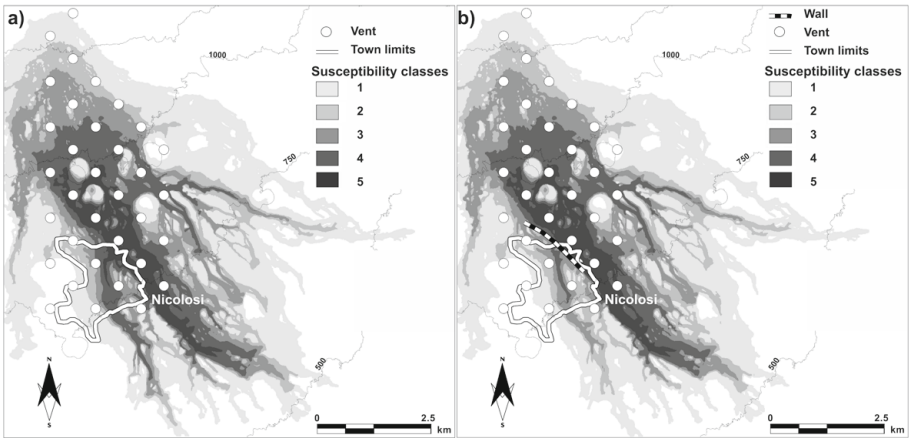


Fig. 7. (a) Map showing the location of a set of vents (white dots) which originate lava flows intersecting a hypothetical 2 km long and 20 m tall earth barrier (see b - right) to protect the centre of Nicolosi (white line); (b) The same area considered in (a), together with the scenario resulting from lava flows intersecting the barrier, which are re-simulated on a modified topography that embeds the presence of the barrier. As shown, the hazard decreases by two classes within the town limits (white line).

Results show that the barrier would indeed be necessary to effectively protect the town centre. The susceptibility here decreases by two classes (Figure 7b) and, at the same time, the areas invaded by diverted flows prove characterised by only a slightly higher susceptibility degree. In this specific case, the protective measure has a substantially positive effect. If this was not the case, further experiments with barriers of different positions and dimensions will reveal to what degree damage from lava flow invasion can be minimized, or whether it would be preferable to abandon any prospects of this kind of protective measure.

5 Conclusive Remarks

The fundamental problem of assessing the impact of future eruptions in a volcanic region lies mostly in the uncertainty concerning their duration, effusion rate, and location. A valid assessment tool relies on the adoption of volcanic hazard maps which, however, are usually based on the sole analysis of past events. Conversely, maps should represent the full range of probable hazards that can be expected to an agreed probability, considering thus all potential future scenarios. As a consequence, probabilistic hazard maps can provide a better base for planning mitigation strategies. We tackled this issue by an elaborate approach in the numerical simulation of a wide variety of lava flows, which are typical of Etna for duration and effusion rate, on a dense grid of vents, by attributing them a statistical likelihood. The methodology here presented deeply relies con the application of the latest SCIARA CA model release, which re-introduces a square tessellation of the cellular space instead of the previously adopted hexagonal one, solves the anisotropic flow direction problem. This result is particularly significant,

being SCIARA a deterministic model, as all the previously proposed solutions refer to probabilistic CA simulation models. This code has been used to simulate the 2001 and 2006 lava flows at Mt Etna (Italy) obtaining a high degree of overlapping between the real and the simulated event and a perfect fitting in terms of run-out were obtained.

A novelty of the presented methodology, besides the possibility of assessing the efficiency of protective measures for inhabited areas and/or major infrastructures, is that the simulation data permits to produce general susceptibility maps in unprecedented detail, and contains each single scenario out of a total of over thousands of simulated cases. It is therefore no longer necessary to wait for the next eruption and know its eruptive parameters and location in order to run ad-hoc simulations, as has been the practice until now. Instead, virtually all possible eruption scenarios can be simulated a priori, from as dense a network of hypothetical vent locations as possible, and extracted in real time as soon as the need arises, as in the case of an imminent or incipient eruptions. Since the obtained results are strongly related to the morphology of the study area, each new eruption will require the creation of an updated DEM incorporating the morphostructural changes induced by the eruption (e.g. [38]). However, re-simulation would be necessary only for those events affecting the modified area, and a new, updated hazard map can then be obtained by simply reprocessing the new set of simulations, which is a quite rapid procedure even on sequential computers. In general, the overall approach presented here can be applied to other volcanoes where a risk of lava flow inundation exists.

Acknowledgements. Authors thank Dr. B. Behncke and Dr. M. Neri from the INGV - Istituto Nazionale di Geofisica e Vulcanologia of Catania (Sicily, Italy), who provided topographic maps and the volcanological data. The authors are also grateful to Prof. G.M. Crisci and Prof. S. Di Gregorio for their precious comments and the common researches.

References

1. Behncke, B., Neri, M.: Cycles and Trends in the recent eruptive behaviour of Mount Etna (Italy). Can. J. Earth Sci. 40, 1405–1411 (2003)
2. Dibben, C.J.L.: Leaving the city for the suburbs - The dominance of 'ordinary' decision making over volcanic risk perception in the production of volcanic risk on Mt Etna, Sicily. J. Volcanol. Geotherm. Res. 172, 288–299 (2008)
3. Barberi, F., Carapezza, M.L., Valenza, M., Villari, L.: The control of lava flow during the 1991-1992 eruption of Mt. Etna. J. Volcanol. Geotherm. Res. 56, 1–34 (1993)
4. Barberi, F., Brondi, F., Carapezza, M.L., Cavarra, L., Murgia, C.: Earthen barriers to control lava flows in the 2001 eruption of Mt. Etna. J. Volcanol. Geotherm. Res. 123, 231–243 (2003)
5. Ishihara, K., Iguchi, M., Kamo, K.: Lava flows and domes: emplacement mechanisms and hazard implications. In: IAVCEI Proceedings, pp. 174–207. Springer, Heidelberg (1990)
6. Del Negro, C., Fortuna, L., Herault, A., Vicari, A.: Simulations of the 2004 lava flow at Etna volcano using the magflow cellular automata model. Bull. Volcanol. 70, 805–812 (2008)
7. Avolio, M.V., Crisci, G.M., Di Gregorio, S., Rongo, R., Spataro, W., D'Ambrosio, D.: Pyroclastic Flows Modelling using Cellular Automata. Comp. Geosc. 32, 897–911 (2006)
8. Felpeto, A., Arana, V., Ortiz, R., Astiz, M., Garcia, A.: Assessment and modelling of lava flow hazard on Lanzarote (Canary Islands). Nat. Hazards 23, 247–257 (2001)

9. Favalli, M., Tarquini, S., Fornaciai, A., Boschi, E.: A new approach to risk assessment of lava flow at Mount Etna. Geology 37, 1111–1114 (2009)
10. Crisci, G., Rongo, R., Di Gregorio, S., Spataro, W.: The simulation model SCIARA: the 1991 and 2001 lava flows at Mount Etna. J. Volcanol. Geotherm. Res. 132, 253–267 (2004)
11. Dragoni, M., Bonafede, M., Boschi, E.: Downslope flow models of a Bingham liquid: Implications for lava flows. J. Volc. Geoth. Res. 30(3-4), 305–325 (1986)
12. Crisp, J.A., Baloga, S.M.: A model for lava flows with two thermal components. J. Geophys. Res. 95, 1255–1270 (1990)
13. Longo, A., Macedonio, G.: Lava flow in a channel with a bifurcation. Phys. Chem. Earth Part A - Solid Earth and Geodesy 24(11-12), 953–956 (1999)
14. Rongo, R., Spataro, W., D'Ambrosio, D., Avolio, M.V., Trunfio, G.A., Di Gregorio, S.: Lava flow hazard evaluation through cellular automata and genetic algorithms: an application to Mt Etna volcano. Fund. Inform. 8, 247–268 (2008)
15. Vicari, A., Herault, A., DelNegro, C., Coltelli, M., Marsella, M., Proietti, C.: Modelling of the 2001 Lava Flow at Etna Volcano by a Cellular Automata Approach. Environ. Model. Soft. 22, 1465–1471 (2007)
16. D'Ambrosio, D., Rongo, R., Spataro, W., Avolio, M.V., Lupiano, V.: Lava Invasion Susceptibility Hazard Mapping Through Cellular Automata. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 452–461. Springer, Heidelberg (2006)
17. Crisci, G.M., Avolio, M.V., Behncke, B., D'Ambrosio, D., Di Gregorio, S., Lupiano, V., Neri, M., Rongo, R., Spataro, W.: Predicting the impact of lava flows at Mount Etna. J. Geophy. Res. 115(B0420), 1–14 (2010)
18. Crisci, G.M., Di Gregorio, S., Ranieri, G.: A cellular space model of basaltic lava flow. In: Proceedings Int. Conf. Applied Modelling and Simulation 1982, Paris-France, vol. 11, pp. 65–67 (1982)
19. Von Neumann, J.: Theory of self reproducing automata. Univ. Illinois Press, Urbana (1966)
20. Weimar, J.R.: Three-dimensional Cellular Automata for Reaction-Diffusion Systems. Fundam. Inform. 52(1-3), 277–284 (2002)
21. Succi, S.: The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford Univ. Press (2004)
22. Chopard, B., Droz, M.: Cellular Automata Modeling of Physical Systems. Cambridge University Press (1998)
23. McNamara, G.R., Zanetti, G.: Use of the Boltzmann equation to simulate lattice-gas automata. Phys. Rev. Lett. 61, 2332–2335 (1988)
24. McBirney, A.R., Murase, T.: Rheological properties of magmas. Ann. Rev. Ear. Planet. Sc. 12, 337–357 (1984)
25. Di Gregorio, S., Serra, R.: An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. Fut. Gener. Comp. Syst. 16, 259–271 (1999)
26. D'Ambrosio, D., Spataro, W.: Parallel evolutionary modelling of geological processes. Paral. Comp. 33(3), 186–212 (2007)
27. Oliverio, M., Spataro, W., D'Ambrosio, D., Rongo, R., Spingola, G., Trunfio, G.A.: OpenMP parallelization of the SCIARA Cellular Automata lava flow model: performance analysis on shared-memory computers. In: Proccedings of the International Conference on Computational Science, ICCS 2011, vol. 4, pp. 271–280 (2011)
28. Spataro, W., Avolio, M.V., Lupiano, V., Trunfio, G.A., Rocco, R., D'Ambrosio, D.: The latest release of the lava flows simulation model SCIARA: First application to Mt Etna (Italy) and solution of the anisotropic flow direction problem on an ideal surface. In: Proccedings of the International Conference on Computational Science, ICCS 2010, vol. 1(1), pp. 17–26 (2010)
29. Park, S., Iversen, J.D.: Dynamics of lava flow: Thickness growth characteristics of steady 2-dimensional flow. Geophys. Res. Lett. 11, 641–644 (1984)

30. Avolio, M.V., Di Gregorio, S., Rongo, R., Sorriso-Valvo, M., Spataro, W.: Hexagonal cellular automata model for debris flow simulation. In: Proceedings of IAMG, pp. 183–188 (1998)
31. D'Ambrosio, D., Di Gregorio, S., Iovine, G.: Simulating debris flows through a hexagonal Cellular Automata model: Sciddica S3-hex. Nat. Haz. Ear. Sys. Scien. 3, 545–559 (2003)
32. Miyamoto, H., Sasaki, S.: Simulating lava flows by an improved cellular automata method. Comp. Geosci. 23, 283–292 (1997)
33. Crisci, G.M., Di Gregorio, S., Nicoletta, F., Rongo, R., Spataro, W.: Analysing Lava Risk for the Etnean Area: Simulation by Cellular Automata Methods. Nat. Haz. 20, 215–229 (1999)
34. Avolio, M.V., D'Ambrosio, D., Lupiano, V., Rongo, R., Spataro, W.: Evaluating Lava Flow Hazard at Mount Etna (Italy) by a Cellular Automata Based Methodology. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2009, Part II. LNCS, vol. 6068, pp. 495–504. Springer, Heidelberg (2010)
35. Cappello, A., Vicari, A., DelNegro, C.: A retrospective validation of lava flow hazard map at Etna Volcano. Spec. Issue of Annals of Geophy. (2011) (to appear)
36. Ho, C.H., Smith, E.I., Feuerbach, D.L., Naumann, T.R.: Eruptive calculation for the Yucca Mountain site, USA: Statistical estimation of recurrence rates. Bull. Volcanol. 54, 50–56 (1991)
37. Behncke, B., Neri, M., Nagay, A.: Lava flow hazard at Mount Etna (Italy): New data from a GIS-based study. Spec. Pap. Geol. Soc. Am. 396, 187–205 (2005)
38. Tarquini, S., Favalli, M.: Changes of the susceptibility to lava flow invasion induced by morphological modifications of an active volcano: the case of Mount Etna, Italy. Nat. Hazards 54, 537–546 (2010)

A Consistent Preliminary Design Process for Mechatronic Systems

Jean-Yves Choley[1], Régis Plateaux[1], Olivia Penas[1], Christophe Combastel[2], and Hubert Kadima[3]

[1] SUPMECA, Lismma, 3 rue F. Hainaut, 93407 Saint Ouen Cedex, France
[2] ENSEA, ECS-lab, 6 avenue du Ponceau, 95014 Cergy-Pontoise Cedex, France
[3] EISTI, Laris, avenue du parc, 95011 Cergy-Pontoise Cedex, France
{jean-yves.choley,regis.plateaux,olivia.penas}@supmeca.fr
combastel@ensea.fr, hubert.kadima@eisti.eu

Abstract. In this paper, a consistent and collaborative preliminary design process for mechatronic systems is described. First, a functional analysis is carried out from user requirements with SysML. This allows one to define suitable architectures and associated test cases. Each of them has to be analysed and optimized separately in order to select the best architecture and the best set of key parameters. The next step of the preliminary design is a modelling of its architecture and its behaviour. In order to merge multi-physical and geometrical parameters, our generic method relies on a topological analysis of the system and generates a set of equations with physical and topological constraints previously defined. Finally, an interval analysis is implemented, allowing one to explore exhaustively the search space resulting from a declarative statement of constraints, in order to optimize the parameters under the constraint of the relevant test cases. An automotive power lift gate scenario has been chosen to test this design process.

Keywords: Mechatronic, preliminary design, SysML, CSP solver, topology.

1 Introduction

Nowadays, system engineering problems are solved using a wide range of domain-specific modelling languages and tools. Standards such as ISO 15288 detail the large number of system aspects and various components of multi-domain systems [1-2]. It is also not realistic to create an all-encompassing systems engineering language capable of modelling and simulating every aspect of a system. However, for multi-domain systems, a global approach is necessary. Indeed, each domain has its own methodologies and languages, thus impeding the consistency of the different modelling. Hence, a global optimization is difficult during the preliminary design process of these systems.

Mechatronic systems development involves considering the modelling of their components together with their interactions. Models can be used to formally represent all aspects of a systems engineering problem, including requirements, functional, structural, and behavioural modelling. Additionally, simulations can be performed on these models in order to verify and validate the effectiveness of design decisions.

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 101–112.
DOI: 10.1007/978-3-642-34336-0_7

This study covers the preliminary design phase of a mechatronic system, in order to verify that the chosen design is in accordance with the system requirements and to verify that this chosen design minimizes risks in further design phases. Following the recent advances in Model Based System Engineering [3], the preliminary design can be viewed as a model transformation process [4].

Based on the example of a power lift gate, our goal is to show how the engineering knowledge can be formalized and used all along the three following phases of the preliminary design process: requirements definition and functional analysis, geometrical and physical modelling, optimization.

Once the early design phases have been performed with SysML models, the physical modelling of the overall system has to be built, based on the topology of the system, in order to generate the equations required for the optimization phase. This being done, the Design Space Exploration can be executed in order to discover the optimal design solution from all functional and architectural specifications and constraints. Indeed, the most efficient way to explore this design space is to reason about previous SysML models, thus proving in a mathematically rigorous way that all required properties and constraints are met.

2 A Power Lift Gate Scenario

An automotive power lift gate (Fig. 1) includes a lift gate door hinged to a car body. This system moves the lift gate between its open and closed positions, thanks to electric cylinders (Fig. 2) that replace the usual gas struts in a classic manual lift gate. It includes a motor and a gearbox that are fixed to the base tube and a jackscrew that drives the upper tube, helped by a spring, in order to sustain static forces. Both electric cylinders are identical and are fixed to the car body and the lift gate.

In order to ensure that the main requirements are fulfilled, such as the opening duration and the power consumption, the electric cylinder has to be preliminarily designed, whatever its internal structure is, meaning that the fixing points on the carbody and the lift gate, the force needed to open and maintain the lift gate and its full length and rest length have to be determined.

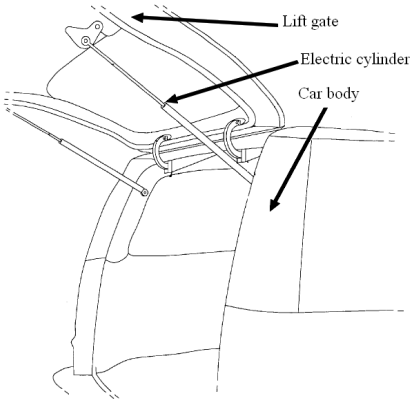


Fig. 1. The power lift gate location on a car body

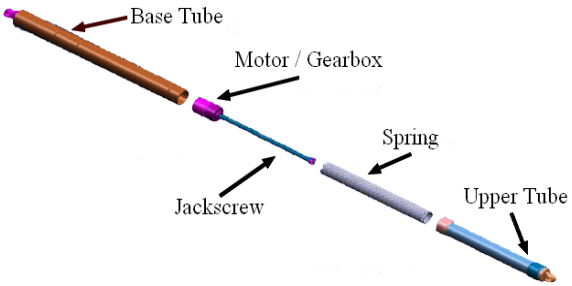


Fig. 2. The electric cylinder architecture

Table 1 summarizes some user requirements indispensable for our study:

Table 1. Power lift gate system user requirements

Id	Name	Definition of the requirement
U0	Usability	The system should be able to open/close the gate from any position, even in cold conditions or extreme angle conditions (car on a slope)
U1	Safety Usage	The system should be able to hold the gate in any medium position by itself without any external power (manual, electrical,…)
U2	Back drive ability with minimal manual force	U21 The system should include a manual function. The back drive ability of a lift gate means that the system should not block a movement created by an outside additional force on the gate.
		U22 The system should ensure that the force the user has to deploy without electrical assistance should not be higher than a maximum level in any position or condition (opened, closed, intermediate position, cold, hot, uphill, downhill)
U3	Minimal actuated force	The system should ensure that the force should not be higher than a maximum level in any position or condition (opened, closed, intermediate position, cold, hot, uphill, downhill).
U4	Minimal changes on the car design	U41 The system should be easily integrated in the car design. This means that, the number of modifications on the car design must be as low as possible (lower costs for the car manufacturer). U411 The fixation point between the car body and the actuator has to be within a specified area. U412 The fixation point between the lift gate and the actuator has to be within a specified area.
		U42 Ideally, the system should also be easily adaptable to different models of cars. Thus an enclosing box has to be respected.

3 A Preliminary Design Process

The proposed preliminary design process relies on a methodology that deals with different modelling, (SysML model, topological model) in order to provide consistent equations for an optimisation of the mechatronic system with interval analysis.

3.1 Modelling of the Power Lift Gate System with SysML

We propose a modelling of a power lift gate system by means of appropriate SysML models at the early stages of the technical engineering process. The different SysML diagrams make it possible for engineers from various disciplinary fields to share a common view about the system. First, we create an extended context diagram, in order to present the different interactions between the extended system (Lift gate with Electric cylinder) and its environment (Fig. 3).

Then a Use Case Diagram is defined to describe the system services (Fig. 4).

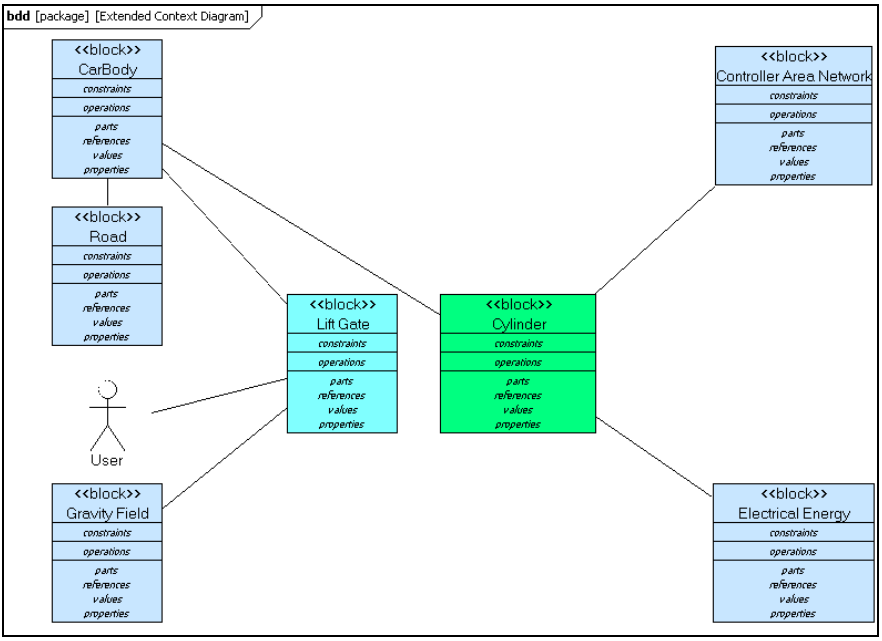


Fig. 3. Extended context diagram of the power lift gate system

SysML Requirement Diagram can be used to clearly organize user and derived system requirements (Fig. 5). By using a hierarchical representation of the requirements, clear gains can be made in the elaboration of requirements, in tradeoffs, as well as in the validation and the verification of requirements. Indeed, during design activities, verification activities need to be defined to satisfy system constraints and properties. Links between the Requirements Diagram and other models allow engineers to connect test criteria to test cases used throughout the development process.

During the architecture analysis, system synthesis by assigning functions to identified physical architecture elements (subsystems, components) is carried out (Fig. 6). Finally we create a kinematic joint diagram (Fig. 7) with connectors regarding to application points and with links representing the field or the type of joints between two elements.

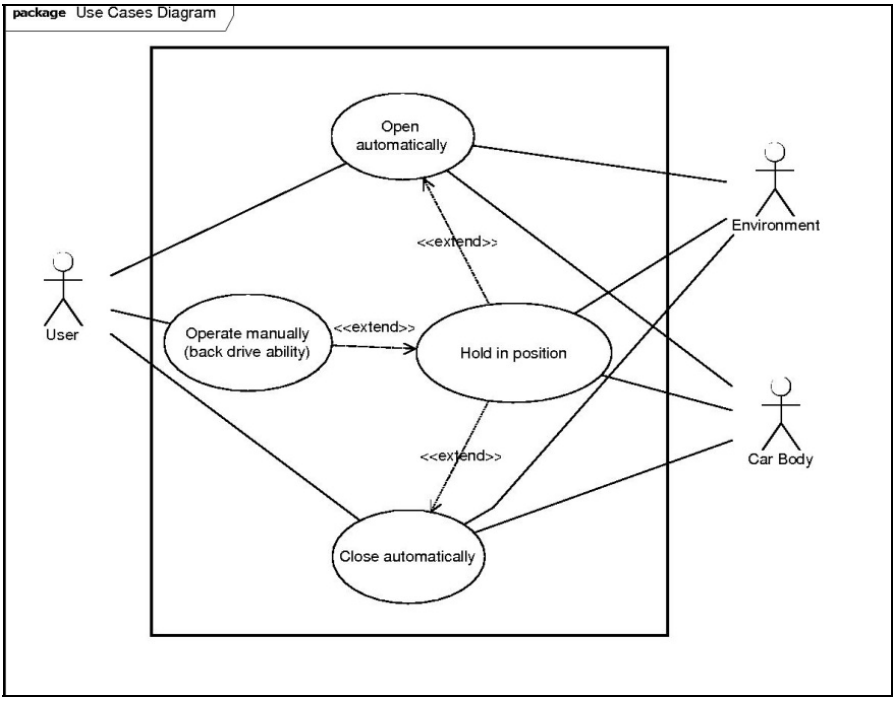


Fig. 4. Use cases diagram of the power lift gate system

3.2 Vector-Based Mechanical Modelling Derived from System Topology

The previous SysML diagrams bring to light the key parameters and the topology of the power lift gate system. In order to optimize these key parameters, this mechanical problem has to be translated into equations. We propose to use a highly suitable method [5] for multi-domain systems such as automotive mechatronic components. Based on a topological analysis of the system, this generic method delivers equations that can be processed by a solver. It relies on the works of Kron [6], Branin [7] and Björke [8]. Here, our method is restrained to the mechanical study of the static equilibrium of the lift gate but it may also be used to express the internal structure of the electric cylinder (screw and nut system, tubes, gearbox, spring, sensors, electrical engine and electronic components...).

The isolated system includes the lift gate with the electric cylinder between the points M and N, the car body being an external system. Let us assume that: the mechanical joints are perfect; points A, M and G belong to the system boundary; there is

neither external mechanical force nor torque on internal point N; P is the external force on the gravity centre G; F_C is the force created by the electrical cylinder, which corresponds to the internal force R_{MN}.

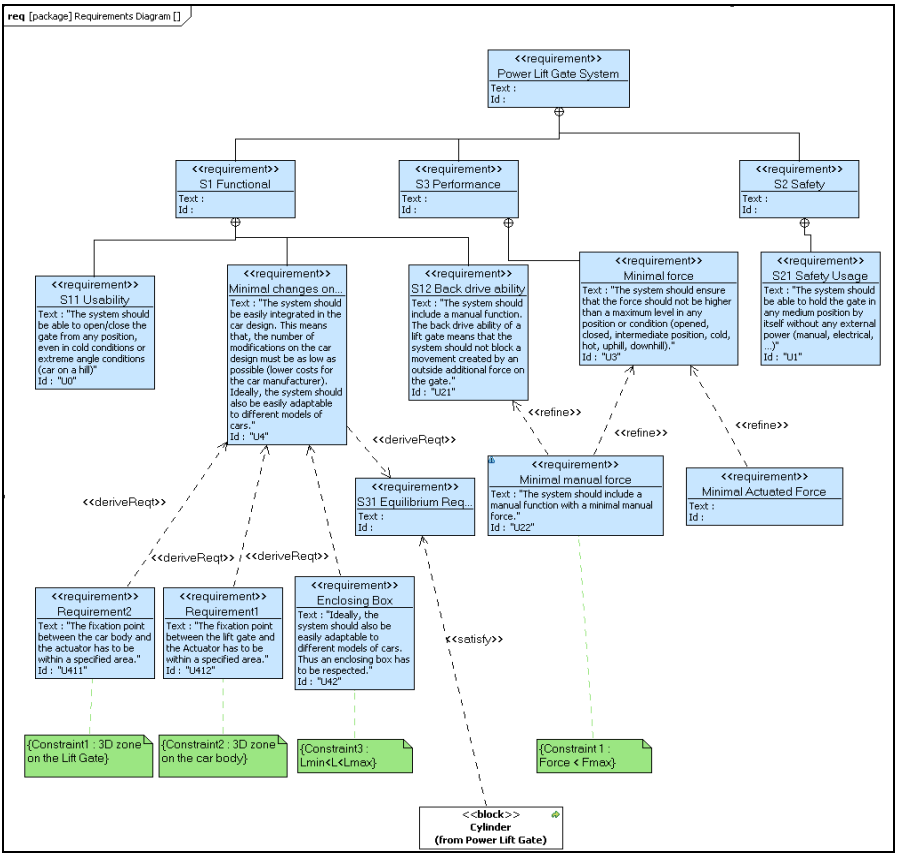


Fig. 5. Requirements diagram

In order to model the architecture of the system, a topological graph has first to be defined from geometrical and mechanical definitions of the problem (Fig. 8).

We use the kinematic joints diagram and the vectorial constraints between characteristic points of previous SysML diagrams to describe the topological structure. Indeed, each connector in the kinematic joints relations diagram represents a particular point, named “node” in the topological structure, and each link between two connectors gives the nature of the kinematic screw, dual of its static screw. The automation of this process between SysML diagrams and our topological representation is made through the analysis of a xml/xmi generated file from the SysML Kinematic Joints Relations Diagram (Fig. 7). So, the boundary of the system is expressed by means of labels attached to each node (boundary) named (A, G, ...), like the SysML connectors, and to each branch (internal), all of them inherited from SysML diagrams.

Then, the topology has to be mathematically expressed (Fig. 9) using a connection (or incidence) matrix named C and an algebraic graph that allows one to connect nodes and branches. The topological structure (graph) is overlaid with an algebraic

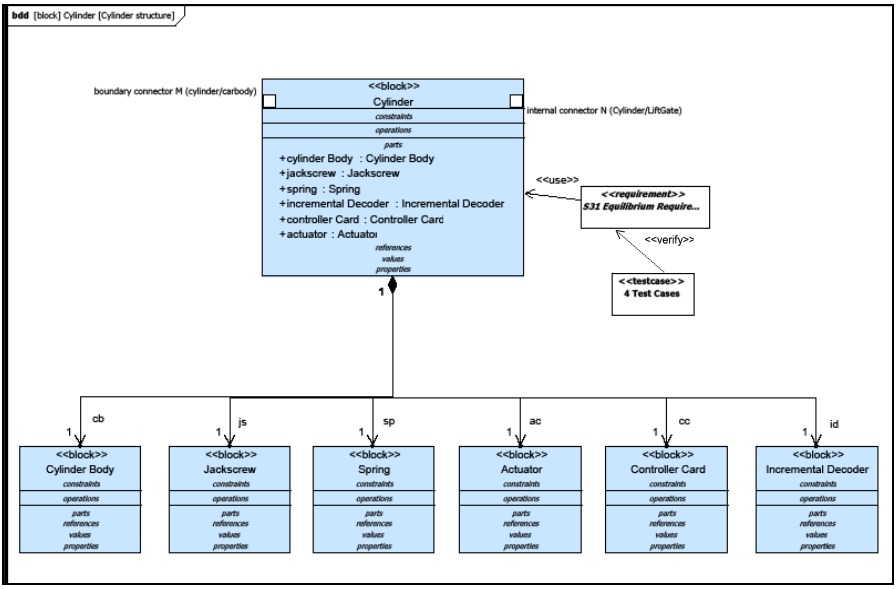


Fig. 6. Architecture, test cases and system requirements attachment

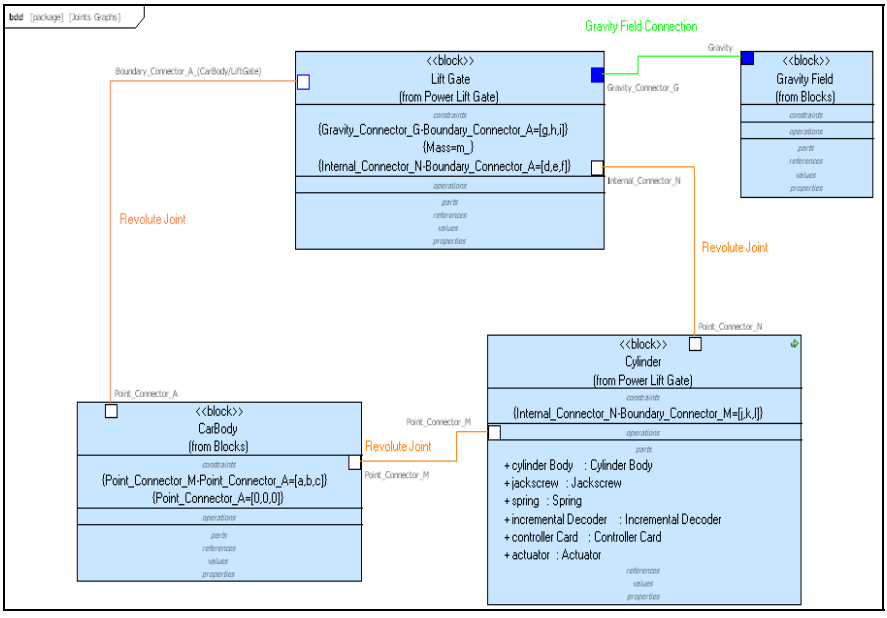


Fig. 7. Kinematic joints relations diagram in SysML

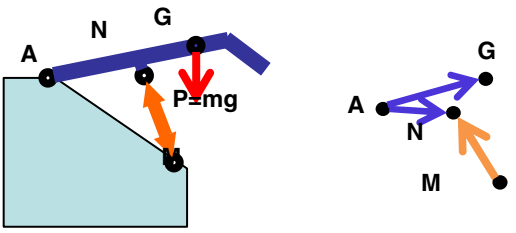


Fig. 8. Power lift gate topological graph

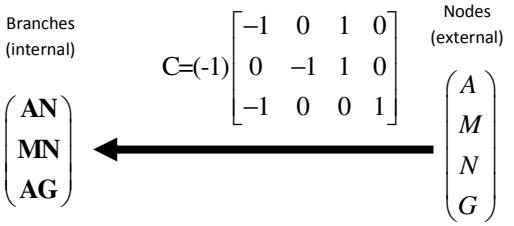


Fig. 9. Connection matrix and geometry (Nodes towards branches)

structure. This global structure connects nodes and branches of the graph, and may include physical parameters governing the behaviour of the system. This method has been thoroughly described in previous papers [5], [9].

Thus, the transposed matrix C^T can be used to express (Fig. 10) the connection between internal and external mechanical forces and moments, defined with their associated static screws, with T_A standing for "screw of external mechanical action on point A" and T_{AN} standing for "screw of internal mechanical action on AN structure":

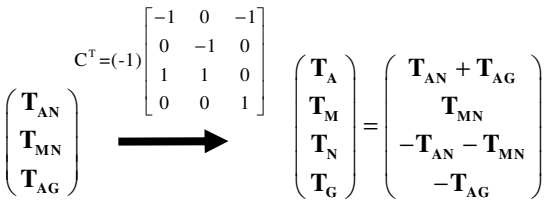


Fig. 10. Connection matrix and mechanical actions (Branches towards nodes)

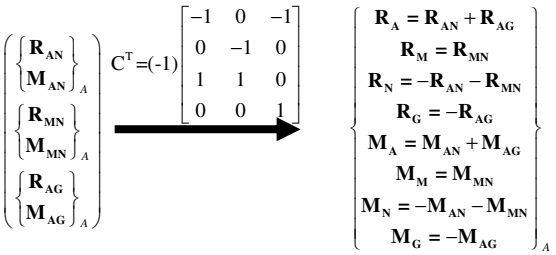


Fig. 11. Equations system

As a result, an equations system (Fig. 11) is obtained, with the decomposition of screws in 4 force equations and in 4 moment equations expressed in the arbitrarily chosen point A:

The equations system is solved and the static equilibrium of the lift gate system is expressed.

3.3 Computational Support for the Exploration of the Solution Space Based on Constraint Programming and Interval Analysis

Special emphasis is also placed on interval-based computational methods [10] allowing one to explore exhaustively the search space resulting from a declarative statement of constraints [11]. Given the previous high level vector model linked to a given topology, formal calculus and causal ordering based on bipartite graphs theory [12-14] can be used to avoid part of the tedious work consisting in giving the mathematical expressions of some constraints as required to run dedicated solvers. The use of interval computations within a constraint programming paradigm [15] also provides a computational support to quantify uncertainties and to detect inconsistencies. From a methodological point of view, the refinement inherent to the design process is underlined.

A Constraint Satisfaction Problem (CSP) is usually defined by (X, D, C) where $X = \{x_1, x_2, \ldots, x_n\}$ is a set of variables, $D = \{d_1, d_2, \ldots, d_n\}$ is a set of domains such that $\forall i \in \{1,\ldots, n\}$, $xi \in di$, and $C = \{C1, \ldots, Cm\}$ is a set of constraints depending on the variables in X. Each constraint includes information related to constraining the values for one or more variables. When continuous variables are considered, the use of interval analysis techniques naturally arises in order to represent the domains. Those methods make it possible to explicitly take uncertainties (in the sense of deterministic imprecision rather than probabilistic variability) into account in the preliminary design process. The use of an interval CSP solver (here, RealPaver) [16] allows an exhaustive search within the search space D which is partitioned into three sets, $D = D_0 \cup D_1 \cup D_?$, the latter two being described by a box paving: D_0 is a sub-domains of D where the constraints are never satisfied; D_1 is a sub-domains of D where the constraints are always satisfied; $D_?$ is a sub-domains of D where the satisfaction of the constraints has not been decided yet according to some stopping criterion (precision, for instance).

From an engineering design point of view, the variables in X can be a set of design parameters, the domains in D can be used to define the range of the search space of interest, and the constraints in C can be concurrently stated by several engineers in any order. Such a declarative modelling is a significant advantage of the CSP paradigm throughout the life cycle of a Computer Aided Engineering (CAE) application [17].

From a methodological point of view, the refinement inherent to the design process can be supported as follows: the poor initial knowledge results in a small number of constraints with few variables belonging to rather large intervals; then, the sequence of assumptions, trials and evaluations constituting the heart of an iteration within the design refinement loop allows the engineers to acquire knowledge, to organize it, and to gradually converge toward what will become the detailed solution [18].

In this paper, our case study is restricted to a few design parameters and focuses on the equilibrium requirement for the power lift gate. The design parameters are $X = [x_{MB}, y_{MB}, x_{NL}, y_{NL}]$ i.e. the 2D coordinates of the electric cylinder fixation points M (on the car body) and N (on the lift gate). The equilibrium requirement is related to four constraints previously identified in the analysis based on SysML:

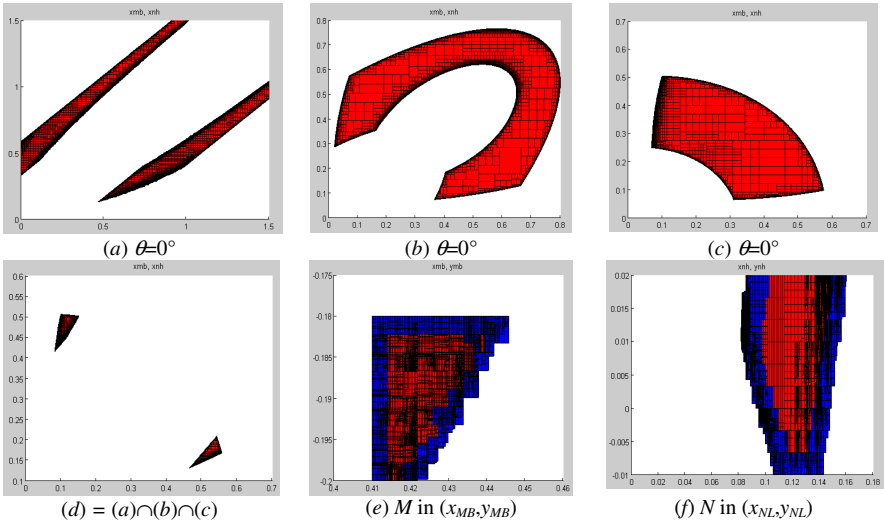
$C_{\Delta F}$: "The additional force value ΔF required to maintain the lift gate static equilibrium is inferior to some threshold level (ΔF_{max})". ΔF refers here to the force ΔF defined as $F_{cyl} = F_{spring} + \Delta F$, where F_{cyl} is the cylinder force required to maintain the static equilibrium and where F_{spring} is the force of the spring within the power cylinder used to reduce the power of the electrical motor;

C_L: "The electric cylinder length L is within the interval $[L_{min}, L_{max}]$ related to the aperture angle of the lift gate";

C_M: "The car body fixation point M is within a specified area";

C_N: "The lift gate fixation point N is within a specified area".

Following formal computations guided using causal ordering techniques, all the constraints are expressed as functions of the design parameters, and the text file required as input of the interval CSP solver is so obtained. The preliminary design of the power lift gate then consists in using the interval solver outputs to understand the influence of the opening angle on the position of fixation points and to perform a (possibly iterative) refinement by selecting an area in the solution space.



(*a*) θ=0° (*b*) θ=0° (*c*) θ=0°

(*d*) = (*a*)∩(*b*)∩(*c*) (*e*) M in (x_{MB},y_{MB}) (*f*) N in (x_{NL},y_{NL})

Fig. 12. Interval CSP outputs (a to f)

Fig. 12 (a-d) illustrates the influence of the opening angle on the solution set. This corresponds to a preliminary study before an exhaustive search for all the opening angles. Focusing on an area in the search space corresponds to the refinement related to the preliminary design process. The reduced search area allows a more precise exploration while preserving a reasonable computation time. The proposed refinement

iteration aims at being reproduced all along the preliminary design process in order to converge toward the solution set, Fig. 12 (e,f), that will be kept to initiate the detailed design of the power lift gate.

4 Conclusions

In this paper we have presented a proven solution for a global multi-domain constraints-based preliminary design supported by a robust design methodology in conformance with ISO IEEE 15288 System Engineering Standard. This solution, based on three interactive design environments (SysML, Topological modelling and Intervals analysis) and illustrated by a mechatronic example, processed with a unique set of key parameters, demonstrates the power of collaborative engineering in model-based design.

SysML allows one to define the high-level relationships between requirements and functional, structural and operational architectures of a system, but lacks detailed semantics to capture some domain-specific properties, for instance, geometry for mechanics.

For this reason, the chosen modelling method based on a topological representation of the whole system, allows one to generate all multi-physical equations, including geometrical parameters. This approach will improve the global optimization of both geometrical and physical parameters.

Then, a refinement methodology based on a sequential decision process and on a declarative statement of constraints is shown to be well supported by the interval CSP paradigm. The use of an interval solver illustrates both the methodological interest in using such tools for preliminary design purposes as well as the need to improve their scalability. This will be the subject of future works.

Although all those design activities may probably be conducted with a unique language such as Modelica, the interoperability of dedicated tools (Artisan or TopCased for SysML, RealPaver for CSP…) has been chosen for their efficiency.

As a result, Model-Based System Engineering simplifies the development of mechatronic and other multi-domain systems by providing a common approach for design and communication across different engineering disciplines.

References

1. ISO/IEC: ISO 15288, Systems lifecycle Processes, International Electro-technical commission, Brussels (2001)
2. Turki, S.: Ingénierie système guidée par les modèles: Application du standard IEEE 15288, de l'architecture MDA et du langage SysML à la conception des systèmes mécatroniques. Thesis of University of South Toulon, France (2008)
3. Estefan, J.: Survey of Candidate Model-Based Systems Engineering (MBSE) Methodologies. Rev. B (May 23, 2008), http://www.omgsysml.org/MBSE_Methodology_Survey_RevB.pdf
4. Hartman, A., Kreische, D. (eds.): Model Driven Architecture - Foundations and Applications. In: First European Conference, ECMDA-FA 2005. Proceedings Springer, Nuremberg (November 2005)

5. Plateaux, R., Choley, J.-Y., Pénas, O., Rivière, A., Cardon, F., Clément, A.: A piezoelectric mechatronic systems modelling based on a topological analysis. In: 7th France-Japan Congress, Mechatronics 2008, Le Grand Bornand, France, May 21-23 (2008)
6. Kron, G.: A short course in tensor analysis for electrical engineers. In: Tensor Analysis of Networks (John Wiley & Sons, Inc., New York, 1939); Wiley/Chapman & Hall, New York/London (1942); republished as Tensors for Circuits. With a new Introduction and List of Publications of Gabriel Kron. Dover, New York (1959)
7. Branin Jr., F.H.: The algebraic-topological basis for network analogies and the vector calculus. In: Proceedings of the Symposium on Generalized Networks, Polytechnic Institute of Brooklyn (1966)
8. Bjørke, Ø.: Manufacturing systems theory – a geometric approach to connection. Tapir publisher (1995) ISBN 82-519-1413-2
9. Plateaux, R., Penas, O., Rivière, A., Choley, J.-Y.: Need for the definition of a topological structure for the complex systems modeling. In: CPI 2007, Rabat, Morocco, October 22-24 (2007)
10. Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Applied interval analysis, with examples in parameter and state estimation, robust control and robotics. Springer (2001)
11. Yannou, B., Simpson, T.W., Barton, R.R.: Towards a computational design explorer using meta-modelling approaches and constraint programming. In: DETC/DAC: ASME Design Engineering Technical Conferences / Design Automation Conference, Chicago, Illinois, USA, September 2-6 (2003)
12. Duff, I.S.: On algorithms for obtaining a maximum transversal. ACM Association for Computing Machinery. Transactions on Mathematical Software 7(3), 315–330 (1981)
13. Duff, I.S.: Algorithm 575, Permutations for a zero-free diagonal. ACM Association for Computing Machinery, Transactions on Mathematical Software 7(3), 387–390 (1981)
14. Pothen, A., Chin-Ju, F.: Computing the Block Triangular Form of a Sparse Matrix. ACM Transactions on Mathematical Software 16(4), 303–324 (1990)
15. Bliek, C., Spellucci, P., Vicente, L.N., Neumaier, A., Granvilliers, L., Monfroy, E., Benhamou, F., Huens, E., Van Hentenryck, P., Sam-Haroud, D., Faltings, B.: Algorithms for solving non linear constrained and optimization problems: the state of the art. The Coconut Project, Deliverable D1 (June 8, 2001)
16. Granvilliers, L.: RealPaver user's manual: solving nonlinear constraints by interval computations. Edition 0.3, for RealPaver Version 0.3, Institut de Recherche en Informatique de Nantes (IRIN) (July 2003)
17. Raphael, B., Smith, I.F.C.: Fundamentals of Computer-Aided Engineering. John Wiley & Sons Ltd. (2003)
18. Aughenbaugh, J.M., Paredis, C.J.J.: Why are intervals and imprecision important in engineering design? In: NSF Workshop on Modelling Errors and Uncertainty in Engineering Computations, REC 2006, hosted by Georgia Tech Savannah, February 22-24 (2006)

A Process Based on the Model-Driven Architecture to Enable the Definition of Platform-Independent Simulation Models

Alfredo Garro*, Francesco Parisi, and Wilma Russo

Department of Electronics, Computer and System Sciences (DEIS),
University of Calabria, via P. Bucci 41C, Rende (CS), 87036, Italy
{garro,w.russo}@unical.it, fparisi@deis.unical.it

Abstract. Agent-Based Modeling and Simulation (ABMS) offers many advantages for dealing with and understanding a great variety of complex systems and phenomena in several application domains (e.g. financial, economic, social, logistics, chemical, engineering) allowing to overcome the limitations of the classical and analytical modelling techniques. However, the definition of agent-oriented models and the use of the existing agent-based simulation platforms often require advanced modelling and programming skills, thus hindering a wider adoption of the ABMS mainly in those domains that would benefit more from it. To promote and ease the exploitation of ABMS, especially among domain experts, the paper proposes the jointly exploitation of both Platform-Independent Metamodels and Model-Driven approaches by defining a Model-Driven process (MDA4ABMS) which conforms to the OMG Model-Driven Architecture (MDA) and enables the definition of Platform-Independent simulation Models from which Platform-Dependent simulation Models and the related code can be automatically obtained with significant reduction of programming and implementation efforts.

Keywords: Agent-based Modeling and Simulation, Model-driven Development, Model-driven Architecture, Platform-independent Simulation Models.

1 Introduction

Approaches which combine agent-based modeling with simulation make it possible to support not only the definition of the model of a system at different levels of complexity through the use of autonomous, goal-driven and interacting entities (agents) organized into societies which exhibit emergent properties, but also the execution of the obtained model to simulate the behavior of the complete system so that knowledge of the behaviors of the entities (micro-level) produces an understanding of the overall outcome at the system-level (macro-level).

Despite the acknowledged potential of Agent-Based Modeling and Simulation (ABMS) for analyzing and modeling complex systems in a wide range of application

* Corresponding author.

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 113–129.
DOI: 10.1007/978-3-642-34336-0_8

domains (e.g. financial, economic, social, logistics, chemical, engineering) [30], these approaches are slow to be widely used as the obtained agent-based simulation models are at a too low-abstraction level, strongly platform-dependent, and therefore not easy to verify, modify and update [17, 22, 28]; moreover, significant implementation efforts which are even more for domain experts, typically lacking of advanced programming skills, are required [30]. In particular, agent-based simulation models can be currently obtained mainly through either direct implementation or manual adaption of a conceptual system model for a specific ABMS platform. The former approach inevitably suffers from the limitations and specific features of the chosen platform, whereas the latter requires additional adaptation efforts, the magnitude of which increases depending on the gap between the conceptual and implementation models of the system.

To overcome these issues, solutions based on approaches well-established in contexts other than the ABMS can be exploited; in particular: (i) approaches based on Platform-Independent Metamodels, which enable the exploitation of more high-level design abstractions in the definition of Platform-Independent Models and the subsequent automatic code generation for different target platforms [1]; (ii) Model-Driven approaches, which enable the definition of a development process as a chain of model transformations [4]. Therefore, some solutions for the ABMS context currently exploit either the approach based on Platform-Independent Metamodels [3, 21, 30] or that based on Model-Driven [18, 22, 28]. The former approach makes available in this context the benefits of exploiting the high level abstraction typical of Platform-Independent Models which also enables the exchange of models regardless of the specific platform used for the simulation; in addition, Platform-Independent Models can be reviewed by domain experts working on different target platforms (possibly on the basis of the simulation result obtained), and then shared with other domain experts. The latter approach enables the definition of complete and integrated processes able to guide domain experts from the analysis of the system under consideration to its agent-based modeling and simulation. In fact, according to the Model-Driven paradigm, the phases which compose a process, the work-products of each phase and the transitions among the phases in terms of model transformations are fully specified; in addition, as the Model-Driven paradigm makes it possible the automatic code generation from a set of (visual) models of the system, the focus can be geared to system modeling and simulation analysis rather than to programming and implementation issues.

Under these considerations, this paper proposes the jointly exploitation of both Platform-Independent Metamodels and Model-Driven approaches as a viable solution able to fully address the highlighted issues so to promote a wider adoption of the ABMS especially in those domains that would benefit more from it. In particular, the paper proposes a Model-Driven process [4] able to guide and support ABMS practitioners in the definition of Platform-Independent Models starting from a conceptual and domain-expert-oriented modeling of the system without taking into account simulation configuration details. The proposed process conforms to the OMG Model-Driven Architecture (MDA) [32] and then allows to (automatically) produce *Platform-Specific simulation Models* (PSMs) starting from a *Platform-Independent simulation Model* (PIM) obtained on the basis of a preliminary *Computation Independent Model* (CIM).

The remainder of this paper is organized as follows: available ABMS languages, methodologies and tools are briefly discussed along with the main drawbacks which still hinder their wider adoption (Section 2), the proposed MDA-based process for ABMS (MDA4ABMS) is presented (Section 3) and then exemplified (Section 4) with reference to a popular problem (the *Demographic Prisoner's Dilemma*) able to represent several social and economic scenarios. Finally, conclusions are drawn and future works delineated.

2 ABMS: Languages, Methodologies and Tools

Several approaches have been proposed to support the definition of agent-based models and/or their implementation for specific simulation platforms; in the following, these approaches, grouped on the basis of the main features they provide, are briefly discussed and their main drawbacks, which still hinder their wider adoption, are highlighted.

1. *Agent-based Modeling and Simulation Platforms.* ABMS platforms, which also provide a visual editor for defining simulation models and, in particular, for specifying agent behaviours, as well as semi-automatic code generation capabilities, are currently available, e.g. Repast for Python Scripting (RepastPy) [11], Repast Simphony (Repast S) [29], the Multi-Agent Simulation Suite (MASS) [19], Ascape [35], SeSAm [25], and Escape [3].

Although the existing ABMS tools attempt to offer *comfortable* modeling and simulation environments, their exploitation is *comfortable* only when used for simple models. In fact, to model complex systems where basic behavior templates provided by the tools must be extended, significant programming skills are essential. Moreover, as these tools do not refer to any specific ABMS process, their use is mainly based on the extension and refinement of the examples and case studies provided, thus limiting such platform-dependent models to lower levels of abstraction and flexibility. Finally, the agent models adopted are often purely reactive and do not take into account organizational issues.

2. *Agent-based Modeling Languages.* Agent modeling languages, mainly coming from the Agent-Oriented Software Engineering (AOSE) domain, can be exploited for a clear, high level and often semantically well-founded definition of ABMS models; some of the wider adopted proposals are the Agent-Object-Relationship (AOR) Modeling [42], the Agent UML (AUML) [5], the Agent Modeling Language (AML) [10] and the Multi-Agent Modelling Language (MAML) [20].

These languages, which do not refer to a specific modeling process, are high-level languages based on graphical and, in some cases, easily customizable notations. Their capabilities make them more suitable as languages for depicting models than as programming languages. Moreover, compared to the models offered by agent-based simulation toolkits, the agent models expressed by these languages are richer, both at micro (agent) and macro (organization) levels. However, the definition of these agent models often requires advanced modeling skills and the transition from the produced design models to specific operational models must be often manually performed; this

task, in absence of tools enabling (semi)automatic transitions, can be quite difficult due to the consistent gap between the design and the operational model of the system.

3. *AOSE Processes and Methods for Agent-based Modeling and Simulation.* Processes and methodologies for the analysis, design and implementation of agent-based models, can be derived from the AOSE domain and possibly adapted for ABMS. Specifically, among the several available AOSE methodologies (such as PASSI [13], PASSIM [14], ADELFE [7], GAIA [43] and GAIA2 [44], TROPOS [8], SONIA [2], SODA+zoom [27], MESSAGE [9], INGENIAS [36], O-MaSE [16], SADDE [39], and Prometheus [34]), some of these, such as GAIA2 [44], SODA+zoom [27] and MESSAGE [9], provide processes, techniques and/or abstractions which are particularly suited for the ABMS context; moreover, specific ABMS extensions of AOSE methodologies can be found in [23, 37, 40].

Although these proposals can represent reference methods for guiding domain experts through the different phases of an ABMS process, only few of them go beyond the high level design phase and deal with detailed design and model implementation issues. As a consequence, they fail in supporting domain experts in the definition of agent-based models which can be directly and effortless executed on ABMS platforms able to fully handle the phases of simulation and result analysis. In fact, the adaptation between the models obtained and the target simulation models requires significant efforts which are time-consuming, error-prone and demands advanced programming skills.

4. *Model-driven Approaches for ABMS.* To fully support and address not only the design but also the implementation of simulation models on available ABMS platforms, some Model-Driven approaches for ABMS have been proposed [18, 22, 28]. However, as they refer to specific ABMS platforms, their exploitation is strongly related to the adoption of these platforms (e.g. Repast Simphony for [18], BoxedEconomy for [22], MASON for [28]).

With reference to other MDA-based approaches, which aim to provide a methodological support for the design of agent-based distributed simulations compliant to the High Level Architecture (HLA) [12, 41], in the ABMS context a still debated issue [26] concerns the trade-off between the overhead which the HLA layer introduces and the provided distribution and interoperability benefits. Specifically, some approaches conceive HLA as the PSM level of an MDA Architecture and provide a process for transforming a System PIM, based on UML, in a HLA-based System PSM [12]. On the contrary, HLA is conceived as the PIM level in [41] where the Federation Architecture Metamodel (FAMM) for describing the architecture of a HLA compliant federation is proposed to enable the definition of platform-independent simulation models (based on HLA) and the subsequent code generation phase.

3 The MDA4ABMS Process

This section describes the proposed MDA4ABMS process which combines the Model-Driven approach and the exploitation of Platform-Independent Metamodels so making available in the ABMS context the benefits of both exploited approaches. The MDA4ABMS process relies on the Model-Driven Architecture (MDA) [32] and the

Agent Modeling Framework (AMF) which is proposed in [3] for supporting the platform-independent modeling.

MDA, which is the most mature Model-Driven proposal launched by the Object Management Group (OMG), is based on the process depicted in Figure 1 where three main abstraction levels of a system and the resulting model transformations are introduced; in particular, the models related to the provided abstraction levels are the following:

- a *Computation Independent Model* (CIM) which describes context, requirements and organization of a system at a conceptual level;
- a *Platform-Independent Model* (PIM) which specifies architectural and behavioral aspects of the system without referring to any specific software platform;
- the *Platform-Specific Models* (PSMs) which describe the realization of the system for specific software platforms and from which code and other development artifacts can be straightforwardly derived.

Transformations between these models (M1 Layer) are enabled by both the corresponding *metamodels* in the M2 Layer and the mappings among metamodels. Each metamodel is defined as instance of the *meta-metamodel* represented in the M3 Layer by the Meta Object Facility (MOF) [31].

The MDA process provides the reference architecture for supporting the generation of target models given a source model as well as the mapping between its metamodel and the target metamodels. To exploit the MDA process in the ABMS domain and obtain agent-based models for specific platforms starting from a platform-independent model, the basic MDA concepts, which have been specifically conceived for the Software Engineering domain, have to be mapped into the ABMS counterparts.

To address these issues, the proposed MDA4ABMS process characterizes the following items (which are highlighted in Figure 1): (i) a reference CIM metamodel for the definition of CIMs which supports the agent-based conceptual system modeling carried out through both abstract and domain-expert oriented concepts (see Section 3.1); (ii) a PIM metamodel for the definition of Platform-Independent ABMS Models (See Section 3.2); (iii) mappings among these metamodels so to enable ABMS model transformations (see Section 3.3). The solution identified for the PIM level allows the

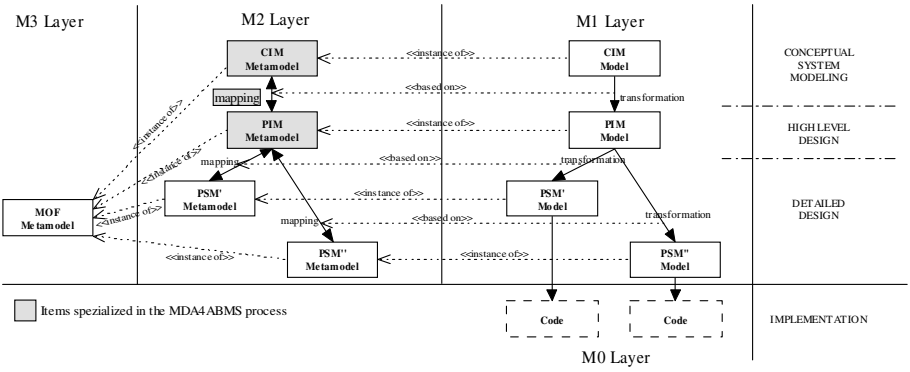


Fig. 1. The MDA-based process

automatic generation of PSMs and the related code for several popular ABMS platforms [3, 29, 35]; on the basis of the provided PIM metamodel, other PSM metamodels for the definition of PSM models can be defined for other and new simulation platforms.

3.1 The CIM Metamodel

The CIM metamodel of the MDA4ABMS process is defined by adopting for the behavior of agents a light and task-based model which combines the strengths of several well-known, task-based agent models [6]. This metamodel is quite general and plain, as required by the abstraction level for which it has been conceived, but powerful enough for representing, at a conceptual level, a great variety of systems in typical ABMS domains.

In particular, the CIM metamodel reported in Figure 2 is centered on the concept of *Agent*. An *Agent*, which is situated in an *Environment* constituted by *Resources,* is characterized by a *Behavior* and a set of *Properties*. *Agents* can be organized into *Societies* which in turn can be organized in *sub-societies*. A *Behavior* is composed by a set of *Tasks* organized according to *Composition Task Rules* which define precedence relations between *Tasks*. Each *Task*, which can act on a set of environment's *Resources*, is structured as an UML 2.0 Activity Diagram which consists of a set of linked *Actions* that can be either *Control Flow* (pseudo) actions (i.e. *start*, *end*, *split*, *join*, *decision*, *merge*, *sequence*) or *Computation* and *Interaction* actions (i.e. *outgoing* or *incoming signals*).

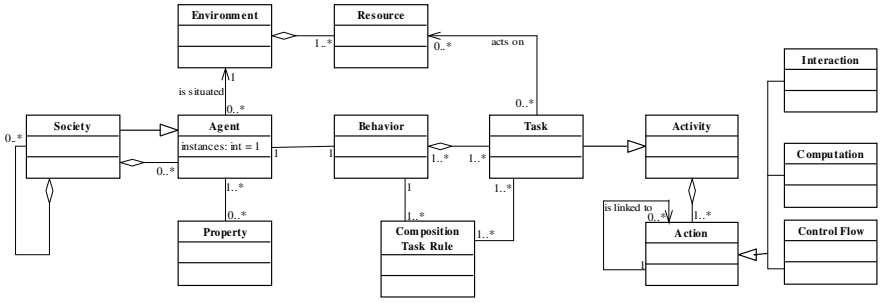


Fig. 2. The CIM metamodel

3.2 The PIM Metamodel

The definition of a PIM metamodel, able to represent a reference metamodel for the definition of Platform-Independent ABMS Models from which different Platform Specific Models (PSMs) can be derived, results in a challenging, long-term standardization process which should also take into account the features of the main ABMS platforms. A more practical solution can be based on the exploitation of the Agent Modeling Framework (AMF) [3] which is meant to provide a reference representation of platform-independent models that can be used to generate simulation models for widely adopted ABMS platforms. In particular, by using the AMF approach, PIM

models can be defined through a hierarchical visual editor and represented by XML documents [38] which are exploited for the generation of PSMs and related code.

Starting from the AMF proposal, the PIM metamodel of the MDA4ABMS process (see Figure 3) has been effortlessly defined. This metamodel is centered on the concept of (Simulation) *Context* (*SContext*) which represents an abstract environment in which (Simulation) *Agents* (*SAgents*) can act. An *SAgent* is provided with an internal *state* consisting of a set of *SAttributes*, a visualization style *SStyle*, and a group of *AActs* (*AGroup*) which constitute its behavior. An *AAct* is characterized by an *Execution Setting* which establishes when its execution can start, its periodicity and its priority.

SContexts, which are themselves *SAgents*, can be organized hierarchically and contain *sub-SContexts*. *SAgents* in an *SContext* can be organized by using *SProjections* which are structures designed to define and enforce relationships among *SAgents* in the *SContext*. In particular, a *SNetwork* projection defines the relationships of both acquaintance and influence between *SAgents* whereas *SGrid, SSpace, SGeography* and *SValueLayer* projections define either the physical space or logical structures in which the agents can be situated.

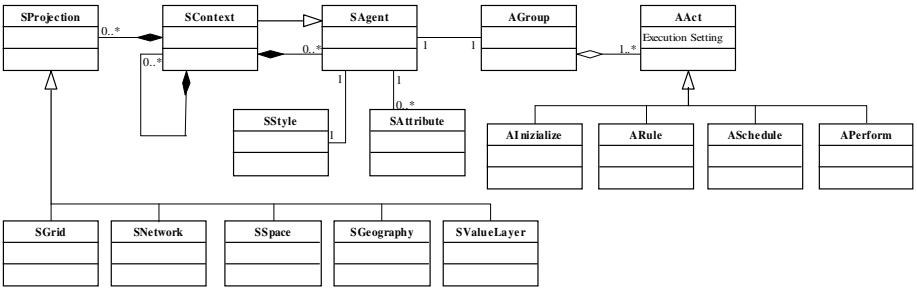


Fig. 3. The PIM metamodel

3.3 From CIM to PIM

With reference to an MDA-based process, a target model can be obtained by transforming a source model (M1 Layer in Figure 1) on the basis of the mapping between the source and target metamodels (M2 Layer in Figure 1). To this end, to enable the definition of instances of concepts of the target metamodel from instances of concepts of the source metamodel, mapping rules among the corresponding concepts along with additional guidelines should be provided [24, 32].

This section, which deals with the mapping between the CIM and PIM metamodels (see Section 3.1 and 3.2) of the MDA4ABMS process, provides the mapping rules (Section 3.3.2) and some guidelines (Section 3.3.3) enabling to transform the CIM entities into PIM entities by taking into account specific aspects (see Section 3.3.1) of the AMF-based PIM metamodel. The subsequent generation of several PSMs (and code for the related ABMS platforms) from the obtained PIM can be then easily carried out by the visual and Eclipse-based modelling environment provided by the AMF framework [3].

3.3.1 Main Aspects of an AMF-Based PIM

Some main aspects have to be considered in the definition of an AMF-based PIM; in this section, the focus is on those which are relevant since they affect the simulation execution of the derived PSMs and which, in particular, concern the proper definition of the *Execution Setting* of an AAct, and the exploitation of *SAttributes* to enable communication among SAgents (see Figure 3).

An AMF-based PIM is defined according to a time-stepped driven simulation approach (the simulation time is incremented in fixed steps) [30], in which, at each simulation step *t*, a set of AAct instances which can be executed and their execution order are defined. Specifically, in a step *t:* (i) for each AAct, belonging to the *AGroup* of an *SAgent* SA, the number of its instances depends on the number of SA instances; (ii) the AAct *Execution Settings* determine the AAct instances to be executed and their execution order.

The Execution Setting of an AAct is characterized by the tuple <*startingTime, period, priority*> where:

- *startingTime* is the first simulation step at which the instances of the AAct are to be executed;
- for each instance of the AAct, *period* is the number of simulation steps which must elapse between two subsequent executions;
- in a simulation step the *priority* value affects the execution order of the *enabled* AActs instances (an AAct is *enabled* at the *simulation step t* if *t* is equal to the AAct *startingTime* which is incremented by a multiple of its *period*).

In a simulation step *t* all enabled AAct instances (regardless of whether they belong to a specific SAgent instance) belong to the same set, *Enabled(t)*, from which the AActs are scheduled for execution on the basis of their *priority* (see Figure 4). As a consequence, the AAct Execution Settings have to be properly defined to guarantee right execution order between AAct instances of both the same SAgent instance (intra-agent AAct interleaving) and different SAgent instances (inter-agent AAct interleaving).

Moreover, in defining the AAct Execution Settings, the different AAct types should be also considered (see Figure 3). In particular, *AActs* of type *AInitialize* are executed once and before any other *AAct* of the *SAgent* (*starting Time* and *period* are both fixed to 0), *AActs* of type *ARule* are executed once at each iteration (*starting Time* and *period* are both fixed to 1), no fixed settings are associated to *AActs* of the *ASchedule* and *APerform* types as *ASchedule* supports periodicity greater than that of

```
ActScheduling (t) {
  AAI = Enabled(t); /* Enabled(t) returns the set of enabled AAct instances at t */
  while (not empty AAI) {
      MPE = maxPriorityEnabled(AAI) ; /* maxPriorityEnabled(AAI) returns a set
                          consisting of the AAct instances with maximum priority in AAI */
      AAI = AAI - MPE;
      while (not empty MPE) {
          aa = randomGet(MPE); /* randomGet(MPE) returns an AAct instance randomly
                                  chosen in (and removed from) MPE */
          execute (aa);
      }
  }
}
```

Fig. 4. Execution of an AMF-based simulation step

a single iteration, whereas *AActs* of type *APerform*, in each iteration in which they are scheduled, are over and over again executed until their escape conditions are met.

With respect to the communication among SAgents, since the *SAttributes* of an SAgent can be freely accessed by all the instances of the SAgent, and the SAttributes of an SContext by all the instances of all the SAgents in the SContext, communication among instances of the same SAgent (intra-agent communication) can exploit SAgent *SAttributes* whereas communication among instances of different SAgents (inter-agent communication) can be enabled by SContext *SAttributes*.

Finally, the design of SAgent communications should take into account how random choices among the enabled AAct (see Figure 4) affect the values of the *SAttributes* on which the communication is based.

3.3.2 Mapping from CIM to PIM Metamodels: Mapping Rules

The automatic generation of a PIM starting from a given CIM is enabled by the QVT/R-based representation of mapping rules [32, 33]. Specifically, due to the different abstraction level between the concepts of the reference CIM and PIM metamodels (see Section 3.1 and 3.2), the mapping rules introduced in this Section along with the QVT/R-based representation allow to obtain a preliminary PIM which needs to be refined by applying additional guidelines (see Section 3.3.3).

The preliminary transformation of a CIM into a PIM, which involves the definition of instances of concepts of the PIM metamodel from instances of concepts of the CIM metamodel by exploiting the mapping rules among the corresponding concepts, consists in the following steps which are listed in the order they should be performed:

R1. each *Society* is transformed into a Simulation Context (*SContext*) and any enclosed Society into a (sub)*SContext* of the corresponding enclosing *Society*; *SAttributes* of each *SContext* are, then, originated by the *Properties* of the corresponding *Society*;

R2. each *Agent* belonging to a *Society* is transformed into an *SAgent* of the corresponding SContext, generating the *SAgent SAttributes* on the basis of the *Agent Properties*, and introducing the *SAgent AGroup* which groups the *AActs* constituting its behavior;

R3. on the basis of the set of *Resources*, which compose the *Environment* in which *Agents* are situated, a set of *SProjections*, whose types (SNetwork, SGrid, SSpace, SGeography, SValueLayer) depend on the characteristics of the mapped *Resources*, are then introduced in the corresponding *SContext*;

R4. *AActs* associated to each *SAgent* are to be defined on the basis of the behavior of the corresponding *Agent* which is composed by a set of *Tasks* organized according to *Composition Task Rules*; this transformation is not direct as requires to take into account the specific aspects of both an AMF-based PIM (see Section 3.3.1) and the simulation scenarios to be represented;

R5. *Actions* which constituted the *Tasks* mapped into an *AAct* have to be properly realized by exploiting the wide set of predefined functions provided by AMF [3].

With reference to the above introduced rules, in the following, some guidelines are provided which address some relevant issues related to: (i) the different communication mechanisms adopted by CIM and PIM metamodels, the former based on

incoming and outgoing signals (see Section 3.1), and the latter on shared *SAttributes* (see Section 3.3.1); (ii) the setting of both *AAct Execution Settings* and related *AAct types* which have to ensure compliance with the *Composition Rules* of the corresponding *Tasks*. Moreover, *AAct Execution Settings* and related *AAct types* should also be set to guarantee intra and inter-agent *AAct* interleavings (see Section 3.3.1) which adhere to the simulation scenarios under consideration.

The QVT/R-based representation of the above introduced mapping rules is exemplified in Figure 5 where the rule R2 for transforming an Agent into an *SAgent* is reported by using the QVT/R graphical notation [33].

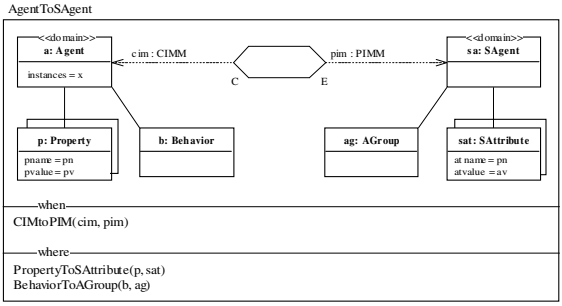


Fig. 5. The QVT/R graphical notation: rule R2

3.3.3 Mapping from CIM to PIM Metamodels: Guidelines

Beside the above introduced *mapping rules* among concepts of the source and target metamodels, further support for CIM to PIM transformation can be provided through *guidelines* which take into account not only the different abstraction level of the concepts in the metamodels but also the main aspects related to the simulation execution model of an AMF-based PIM (see Section 3.3.1). In particular, these *guidelines* propose viable solutions for guiding the choice among the mapping alternatives which often characterize the transformation process from a conceptual level (CIM) to a less abstract level (PIM) typically relying on a simulation execution model. In the following some of these *guidelines* are proposed and exploited in Section 4:

G1. A set of *Tasks* of an *Agent* which, according to the *Composition Task Rules*, can be grouped in a sequence of *Tasks* and in which *Tasks* are related by *Actions* of the *Interaction* type (i.e. the involved *Tasks* send/receive messages to/from the other *Tasks* in the sequence) can be mapped in a single *AAct* of an *SAgent*. The interactions among the involved *Tasks* are then modeled by accessing and modifying the properly introduced *SAttributes* of the *SAgent*.

G2. In case of *Tasks* which should be executed at the same simulation steps, the *Execution Setting* of the resulting *AActs* must have the same *startingTime* and *period* whereas *priorities* must be properly set according to the task organization specified by the *Composition Task Rules*.

G3. The *SAttributes* of an *SContext* should be properly defined not only for mapping the *Properties* of the corresponding *Society* but also for supporting interactions among different *SAgents* belonging to the *SContext*.

G4. *Tasks* (or group of *Tasks*) that must be executed at every simulation step are mapped into *ARules*, except for *Tasks* containing a *Do-While loop* which should be mapped into *APerforms*. Tasks that must be executed with a periodicity different from a single simulation step should be mapped into *ASchedule*; finally, *Tasks* that must be executed once before any other *Tasks* should be mapped into *AInizialize* (an *AAct* of type *AInitialize* should nevertheless be provided for setting the *SAttributes*).

4 Exploiting the Proposed MDA-Based Process

In this section, the MDA4ABMS process is exemplified with reference to the well-known *Demographic Prisoner's Dilemma* which was introduced by Epstein in 1998 [15] and is able to represent several social and economic complex scenarios in which interesting issues regard the identification of starting configurations and conditions that allow initial populations to reach stable configurations (in terms of both density and geographic distribution). Specifically, in these scenarios k players are spatially distributed over an n-dimensional toroidal grid. Each player is able to move to empty cells in its von Neumann neighborhood of range 1 (*feasible* cells), is characterized by a fixed pure *strategy* (c for cooperate or d for defect) and is endowed with a level of wealth w which will be decremented or incremented depending of the payoff earned by the player in each round of the *Prisoner's Dilemma* game played during its life against its neighbors [15]. The player dies when its wealth level w becomes negative, whereas, when w exceeds a threshold level w_b, an *offspring* can be produced with wealth level w_0 deducted from the parent and plays using the same strategy as the parent unless a mutation (with a given rate m) occurs. A player also dies if its *age* exceeds a value age_{max} randomly fixed at the player creation.

4.1 The CIM Model

For the *Demographic Prisoner's Dilemma*, the CIM model envisages a *DPDGame* Society of *k Player* Agents which are situated in an Environment which includes a *Grid* Resource constituted by an n-dimensional toroidal grid. Main Properties of the *DPDGame* Society are Prisoner's Dilemma *payoffs*, initial and threshold wealth levels (w_0, w_b), and mutation rate (m), and those of the *Player* Agent are its wealth level w, *age*, and *strategy*. The Behavior of the *Player* Agent is obtained by composing the set of Tasks reported in Table 1 according to the Composition Task Rules shown in Table 2; corresponding UML Activity diagrams are reported in Figure 6.

4.2 The PIM Model

In this section, the transformation from the defined CIM to a PIM is detailed with reference to a simulation scenario where all players are required to play exactly one round in a simulation step.

The transformation from the CIM to a PIM is enabled by the mapping between the CIM and PIM metamodels (see Sections 3.3.2, 3.3.3) which originates: the *DPDGame* SContex from the DPDGame Society (*rule R1*), the *Player* SAgent from the Player Agent (*rule R2*), the *GameSpace* SProjection from the *Grid* Resource

(*rule R3*), the *Acts* (with their related *Execution Settings*) associated to the *Player* SAgent from the Tasks and associated Composition Task Rules composing the Behavior of the Player (*rule R4*).

Table 1. Indentified tasks

Task Id	Task Name	Description
T1	Walk	The player can move to a *feasible* cell of the *Grid*.
T2	Challenge	*If the von Neumann neighborhood (of range 1) of the player is not empty* the player communicates its strategy to its randomly selected opponent player.
T3	Update Age	The player age is incremented by 1.
T4	Fission	*If the player's wealth level w is greater than the threshold* w_b a new child player can be created in a *feasible* cell of its parent and endowed with w_0 and the same strategy of the parent (unless a mutation with rate *m* occurs). The wealth level of the parent player is decremented by w_0.
T5	Die	*If the wealth level of the player is negative or its age is greater than* age_{max} the player is removed from the *Grid*.
T6	Accept Dare	*When the strategy of an opponent player is provided* the player strategy is communicated to the opponent and the earned payoff is added to the player's wealth level.
T7	Update Wealth Level	*If the strategy of an opponent player is provided* the earned payoff is added to the player's wealth level

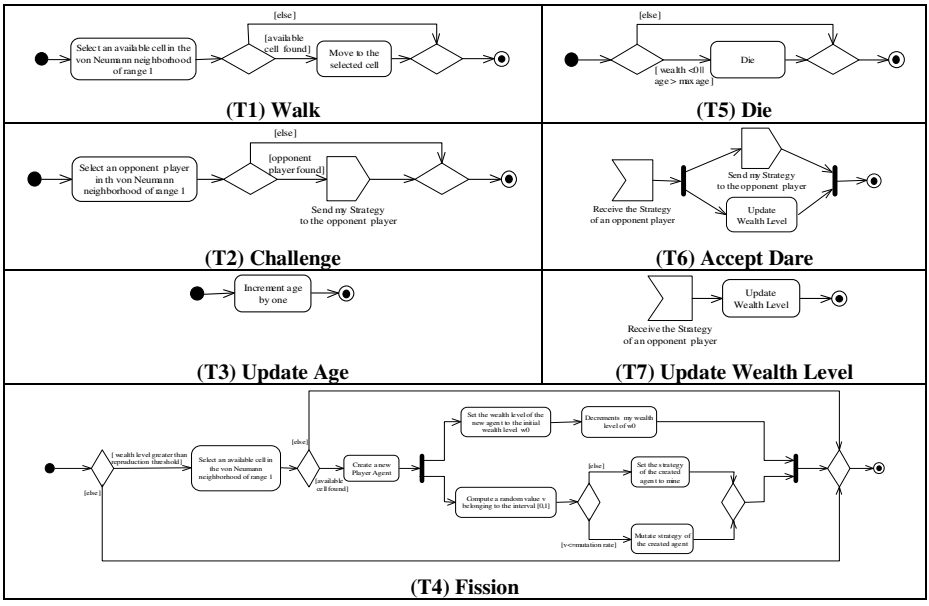


Fig. 6. The UML activity diagrams of the Player Agent tasks

Table 2. Composition task rules

Task Id	Set of Enabling Tasks
T1	-
T2	{T1}
T3	{T1}
T4	{T7}
T5	{T3, T4}
T6	{T2}
T7	{T6}

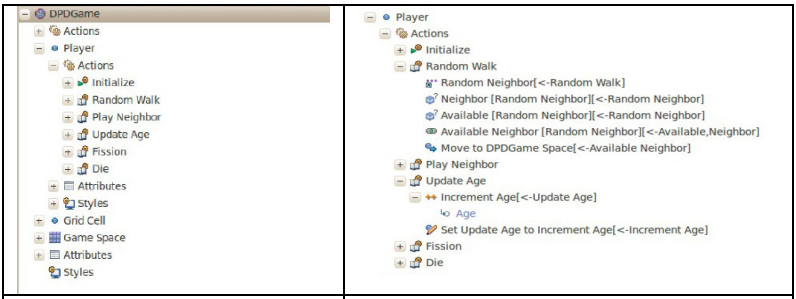
Table 3. Group of acts (AGroup) for the player agent

AAct	**AAct Execution Setting**	**Tasks**
Random Walk	<1,1, a>	T1
Play Neighbor	<1,1, b>, with b<a	T2, T6, T7
Update Age	<1,1, c>, with c<a	T3
Fission	<1,1, d>, with d<c & d<b	T4
Die	<1,1, e>, with e<d	T5

In Table 3 the Acts derived for the *Player* SAgent along with the associated Tasks (see Table 1 and 2) and Execution Settings are reported. As the AMF communication mechanism among instances of an SAgent is based on access to the SAttributes of the SAgent (see Section 3.3.1), a single AAct (*Play Neighbor*) is derived from tasks T2, T6 and T7 which carried out this kind of communication (*guideline G1*). Execution Settings of the AActs in Table 3 are characterized by both *startingTime* and *period* equal to one to guarantee that all the Player SAgents perform all their AActs in each simulation step, and *priorities* are set (*guideline G2*) on the basis of the Compositions Task Rules (see Table 2). On the basis of the AAct *Execution Setting* (see Section 3.3.1) in Table 3 the type of AActs is obtained (*guideline G4*).

In Figure 7.a an example of a PIM model representation, obtained by exploiting the visual and Eclipse-based modelling environment provide by AMF, is reported. Moreover, an AAct of the AInizialize type (*Inizialize*) has been introduced for setting up the *SAttributes* of the *DPDGame* SContext and the *Player* SAgent *(guideline G4)*. In Figure 7.b. the definition of the *Random Walk* and *Update Age* AActs is reported where the actions associated to each AAct are defined by exploiting the wide set of functions provided by AMF (*rule R5*).

Starting from this definition of the PIM model, AMF is able to automatically generate the PSM models and the related code for the ABMS platforms which are currently supported: Repast Simphony [29], Ascape [35] and Escape [3]. The simulation of the system can then be executed in a target simulation environment and simulation results can be thoroughly analyzed by exploiting several analysis tools (as Matlab, R, VisAd, iReport, Jung) which can be directly invoked from the environment.



(a) *DPDGame* model in AMF (b) *Random Walk* and *Update Age* AActs

Fig. 7. The AMF-based PIM model of the DPDGame

5 Conclusions

A wider adoption of the ABMS is still hindered by the lack of approaches able to fully support the experts of typical ABMS domains (e.g. financial, economic, social, logistics, chemical, engineering) in the definition and implementation of agent-based simulation models. In this context, the paper has proposed a solution, centered on the joint use of the Model-Driven Architecture and AMF-based Platform-Independent Metamodel, which aims to overcome the main drawbacks of available ABMS languages, methodologies and tools. In particular, the proposed process (MDA4ABMS) allows to (automatically) produce Platform-Specific simulation Models (PSMs) starting from a Platform-Independent simulation Model (PIM) obtained on the basis of a Computation Independent Model (CIM), thus allowing domain experts to exploit more high-level design abstractions in the definition of simulation models and to exchange/update/refine the so obtained simulation models regardless to the target platform chosen for the simulation and result analysis. Moreover, the semi-automatic model transformations, enabled by the defined metamodels and related mappings, ease the exploitation of the proposed modeling notation and process, while the adoption of the standard UML notation and the visual modeling tool provided by AMP reduce the learning curve of the process.

The MDA4ABMS process has been exemplified with reference to the well-known *Demographic Prisoner's Dilemma* which is able to represent several social and economic complex scenarios thus demonstrating the efficacy of the process and the related tools in supporting domain experts from the definition of conceptual simulation models to their concrete implementation on different target ABMS platforms.

Ongoing research efforts are devoted to: (i) define and extensive experiment a full-fledged ABMS methodology based on the MDA4ABMS process and able to seamlessly guide domain experts from the analysis of a complex system to its agent-based modeling and simulation; (ii) look for frameworks different from AMF (e.g. HLA) suitable to define PIM metamodels able to support the modeling of simulation scenarios with specific requirements such as distribution and/or human participation.

References

1. Agt, H., Bauhoff, G., Cartsburg, M., Kumpe, D., Kutsche, R., Milanovic, N.: Metamodeling Foundation for Software and Data Integration. In: Yang, J., Ginige, A., Mayr, H.C., Kutsche, R.-D. (eds.) UNISCON 2009. LNBIP, vol. 20, pp. 328–339. Springer, Heidelberg (2009)
2. Alonso, F., Frutos, S., Martínez, L., Montes, C.: SONIA: A Methodology for Natural Agent Development. In: Gleizes, M.-P., Omicini, A., Zambonelli, F. (eds.) ESAW 2004. LNCS (LNAI), vol. 3451, pp. 245–260. Springer, Heidelberg (2005)
3. The AMP project, `http://www.eclipse.org/amp/`
4. Atkinson, C., Kühne, T.: Model-driven development: A metamodeling foundation. IEEE Software 20(5), 36–41 (2003)
5. Bauer, B., Müller, J.P., Odell, J.: Agent UML: A Formalism for Specifying Multiagent Software Systems. In: Ciancarini, P., Wooldridge, M.J. (eds.) AOSE 2000. LNCS, vol. 1957, pp. 91–103. Springer, Heidelberg (2001)

6. Bernon, C., Cossentino, M., Gleizes, M.-P., Turci, P., Zambonelli, F.: A Study of Some Multi-agent Meta-models. In: Odell, J.J., Giorgini, P., Müller, J.P. (eds.) AOSE 2004. LNCS, vol. 3382, pp. 62–77. Springer, Heidelberg (2005)
7. Bernon., C., Gleizes, M.P., Picard, G., Glize, P.: The Adelfe Methodology for an Intranet System Design. In: Proc. of the Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS), Toronto, Canada (2002)
8. Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., Perini, A.: TROPOS: an agent-oriented software development methodology. Journal of Autonomous Agents and Multi-agent Systems 8(3), 203–236 (2004)
9. Caire, G., Coulier, W., Garijo, F.J., Gomez, J., Pavón, J., Leal, F., Chainho, P., Kearney, P.E., Stark, J., Evans, R., Massonet, P.: Agent Oriented Analysis Using Message/UML. In: Wooldridge, M.J., Weiß, G., Ciancarini, P. (eds.) AOSE 2001. LNCS, vol. 2222, pp. 119–135. Springer, Heidelberg (2002)
10. Cervenka, R., Trencansky, I.: The Agent Modeling Language - AML. Whitestein Series in Software Agent Technology. Birkhäuser (2007)
11. Collier, N., North, M.: Repast for Python Scripting. In: Proc. of the Agent 2004 Conference on Social Dynamics: Interaction, Reflexivity and Emergence, Chicago, IL (2004)
12. D'Ambrogio, A., Iazeolla, G., Pieroni, A., Gianni, D.: A Model Transformation approach for the development of HLA-based distributed simulation systems. In: Proc. of the International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Noordwikerhout, The Netherlands, July 29-31 (2011)
13. Cossentino, M.: From requirements to code with the PASSI methodology. In: Henderson-Sellers, B., Giorgini, P. (eds.) Agent-Oriented Methodologies, pp. 79–106. Idea Group Inc., Hershey (2005)
14. Cossentino, M., Fortino, G., Garro, A., Mascillaro, S., Russo, W.: PASSIM: a simulation-based process for the development of Multi-Agent Systems. J. of Agent-Oriented Software Engineering 2(2), 132–170 (2008)
15. Dorofeenko, V., Shorish, J.: Dynamical Modeling of the Demographic Prisoner's Dilemma. In: Computing in Economics and Finance. Society for Computational Economics (2002)
16. Garcia-Ojeda, J.C., DeLoach, S.A., Robby, R., Oyenan, W. H., Valenzuela, J.: O-MaSE: A Customizable Approach to Developing Multiagent Development Processes. In: Proc. of the 8th International Workshop on Agent Oriented Software Engineering, Honolulu HI (May 2007)
17. Garro, A., Russo, W.: Exploiting the easyABMS methodology in the logistics domain. In: Proceedings of the Int'l Workshop on Multi-Agent Systems and Simulation (MAS&S 2009) as Part of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW 2009), Turin, Italy, September 7-11 (2009)
18. Garro, A., Russo, W.: easyABMS: a domain-expert oriented methodology for Agent Based Modeling and Simulation. Simulation Modeling Practise and Theory 18, 1453–1467 (2010)
19. Gulyás, L., Bartha, S., Kozsik, T., Szalai, R., Korompai, A., Tatai, G.: The Multi-Agent Simulation Suite (MASS) and the Functional Agent-Based Language of Simulation (FABLES). In: SwarmFest 2005, Torino, Italy, June 5-7 (2005)
20. Gulyas, L., Kozsik, T., Corliss, J.B.: The multi-agent modelling language and the model design interface. J. of Artificial Societies and Social Simulation 2(3) (1999)
21. Hahn, C., Madrigal-Mora, C., Fischer, K.: Interoperability through a Platform-Independent Model for Agents. In: Enterprise Interoperability II, New Challenges and Approaches. Springer (2007)

22. Iba, T., Matsuzawa, Y., Aoyama, N.: From Conceptual Models to Simulation Models: Model Driven Development of Agent-Based Simulations. In: Proc. of the 9th Workshop on Economics and Heterogeneous Interacting Agents, Kyoto, Japan (2004)
23. Iglesias, C.A., Garijo, M., Gonzalez, J.C., Velasco, J.R.: Analysis and Design of Multiagent Systems Using MAS-CommonKADS. In: Singh, M.P., Rao, A., Wooldridge, M.J. (eds.) ATAL 1997. LNCS (LNAI), vol. 1365, Springer, Heidelberg (1998)
24. Karow, M., Gehlert, A.: On the Transition from Computation Independent to Platform Independent Models. In: Proc. of the 12th Americas Conference on Information Systems, Acapulco, Mexico (August 2006)
25. Klügl, F., Herrler, R., Fehler, M.: SeSAm: implementation of agent-based simulation using visual programming. In: Proc. of AAMAS 2006, pp. 1439–1440 (2006)
26. Lees, M., Logan, B., Theodoropoulos, G.: Distributed Simulation of Agent-Based Systems with HLA. ACM Transactions on Modeling and Computer Simulation (TOMACS) 17(3), 11–35 (2007)
27. Molesini, A., Omicini, A., Ricci, A., Denti, E.: Zooming Multi-Agent Systems. In: Müller, J.P., Zambonelli, F. (eds.) AOSE 2005. LNCS, vol. 3950, pp. 81–93. Springer, Heidelberg (2006)
28. Nebrijo Duarte, J., de Lara, J.: ODiM: A Model-Driven Approach to Agent-Based Simulation. In: Proc. of the 23rd European Conference on Modelling and Simulation, Madrid, Spain, June 9-12 (2009)
29. North, M.J., Howe, T.R., Collier, N.T., Vos, J.R.: Repast Simphony Runtime System. In: Proc. of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms, Chicago, IL (2005b)
30. North, M.J., Macal, C.M.: Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation. Oxford University Press (2007)
31. Object Management Group (OMG). Meta Object Facility (MOF) Specifications (version 2.0), http://www.omg.org/spec/MOF/2.0/
32. Object Management Group (OMG). Model Driven Architecture (MDA) Guide Version 1.0.1, http://www.omg.org/cgi-bin/doc?omg/03-06-01
33. Object Management Group (OMG). MOF Query/Views/Transformations (QVT) Specifications (version 1.0), http://www.omg.org/spec/QVT/1.0/
34. Padgham, L., Winikoff, M.: Prometheus: a methodology for developing intelligent agents. In: AAMAS 2002: Proc. of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 37–38. ACM Press (2002)
35. Parker, M.T.: What is Ascape and Why Should You Care? J. Artificial Societies and Social Simulation 4(1) (2001)
36. Pavón, J., Gómez-Sanz, J.J., Fuentes, R.: The INGENIAS Methodology and Tools. In: Agent-Oriented Methodologies. pp. 236–276. Idea Group Publishing (2005)
37. Pavon, J., Sansores, C., Gómez-Sanz, J.J.: Modelling and simulation of social systems with INGENIAS. Int. J. of Agent-Oriented Software Engineering 2(2), 196–221 (2008)
38. Schauerhuber, A., Wimmer, M., Kapsammer, E.: Bridging existing Web modeling languages to model-driven engineering: a metamodel for WebML. In: Proc. of the 6th Int. Conference on Web Engineering (ICWE 2006), Palo Alto, CA. ACM Press (2006)
39. Sierra, C., Sabater, J., Agusti, J., Garcia, P.: Evolutionary Programming in SADDE. In: Procedings of the First International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2002, Bologna, Italy, July 15-19, vol. 3, pp. 1270–1271. ACM Press (2002)

40. Streltchenko, O., Finin, T., Yesha, Y.: Multi-agent simulation of financial markets. In: Kimbrough, S.O., Wu, D.J. (eds.) Formal Modeling in Electronic Commerce. Springer (2003)
41. Topçu, O., Adak, M., Oğuztüzün, H.: A metamodel for federation architectures. ACM Transactions on Modeling and Computer Simulation (TOMACS) 18(3), 10–29 (2008)
42. Wagner, G.: AOR Modelling and Simulation: Towards a General Architecture for Agent-Based Discrete Event Simulation. In: Giorgini, P., Henderson-Sellers, B., Winikoff, M. (eds.) AOIS 2003. LNCS (LNAI), vol. 3030, pp. 174–188. Springer, Heidelberg (2004)
43. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia methodology for agent-oriented analysis and design. Journal of Autonomous Agents and Multi-Agent Systems 3(3), 285–312 (2000)
44. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing Multiagent Systems: the Gaia Methodology. ACM Trans. on Software Engineering and Methodology 12(3), 317–370 (2003)

Dynamic Response of a Wind Farm Consisting of Doubly-Fed Induction Generators to Network Disturbance

Temitope Raphael Ayodele, Abdul-Ganiyu Adisa Jimoh, Josiah Munda, and John Agee

Department of Electrical Engineering, Tshwane University of Technology, Pretoria, South Africa
{AyodeleTR,JimohAA,MundaJL,AgeeJT}@tut.ac.za

Abstract. In this paper, the response of a wind farm consisting of doubly fed induction generators when a disturbance occurs on the network is studied. The disturbances include occurrence of fault on the network, the sudden change in load, loss of transmission line and loss of generation. The influence of generator inertial and fault location on the dynamics of the generator is also considered. First, the mathematical model comprising the variable speed wind conversion system is established. Based on this model, the simulation results describing the behaviour of a wind farm consisting of doubly fed induction generators to different network disturbances are presented.

Keywords: Wind Farm, Doubly Fed Induction Generator, Power System, Disturbance.

1 Introduction

Wind power has proven to be a renewable energy source that is sustainable for electricity generation with lower impact on the environment. The rapid development in wind energy technology and the reduction in wind power production costs have increased its rate of integration into the grid around the world in recent years. At present, the wind power growth rate stands at over 20% annually. At the end of 2010, global cumulative wind power capacity reached 194.4 GW [1] and it is predicted that 12% of the world electricity may come from wind power by the year 2020 [2]. The global exponential growth of wind power cumulative capacity in the last 15 years is depicted in Fig.1.

There are various types of wind turbines in use around the world each having its own advantages and disadvantages [3]. The most used one is the variable speed wind turbine with doubly fed induction generator (DFIG) due to the numerous advantages it offers over others [4]. The stator of DFIG is directly connected to the grid while the rotor is coupled to the grid through a Pulse Width Modulation (PWM) frequency converter. One of the attractive characteristics of this generator is that the converter carries only the rotor slip power typically in the range of 10-15% of the generated power [5]. The reduced rating of the converter reduces the cost and the power losses,

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 131–150.
DOI: 10.1007/978-3-642-34336-0_9

the annual energy capture is in the range of 20–30% higher than the fixed speed wind generator [6] . The use of capacitor banks is eliminated because it has both active and reactive power control capability which also enhances its contribution to voltage and load flow distribution control in the power system. The lower mechanical stress imposed by DFIG on the gearbox extends the life span of this expensive device. The controllability of the speed makes it possible to use aerodynamic pitch control, which effectively limits the generated power during high wind speed periods. Flickers caused by the aerodynamic torque oscillation and the wind gust are greatly reduced thereby improving the power quality of the network.

Until quite recently, the power system mainly consisted of the synchronous generators. The behaviour and the characteristic of these conventional generators to network disturbance are generally well understood by the utility operators. With the advent of wind power, induction generator technologies are introduced into the power system. This poses a lot of concern to most utility operators as the response of these generators to network disturbance is not well understood.

Most existing literature is focused on the analysis of the behavior of power system networks as a result of wind farm integration [4, 7-9]. In this paper, however, the behaviour of a wind farm as a result of disturbance in the power system network is the subject of study. The study is limited to Wind farm (WF) consisting of variable speed DFIGs.

The rest of the paper is organized as follows; section two presents the model of the wind conversion system made of variable speed DFIG. In section three, the system under study is described. Simulation results obtained are discussed in section four while section five presents the conclusions.

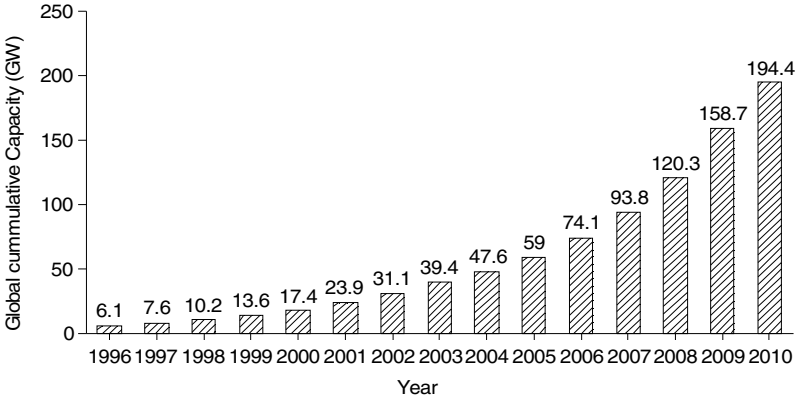


Fig. 1. Source: Adapted from [1]

2 Modelling of DFIG Wind Conversion System

Wind conversion system (WCS) comprises of the aerodynamic system, the mechanical shaft system, electrical system of the induction generator, the pitch control system, the speed control system, the rotor side converter controller and the grid side converter controller. All these systems are combined together to form a unit system of a wind farm as depicted in Fig 2.

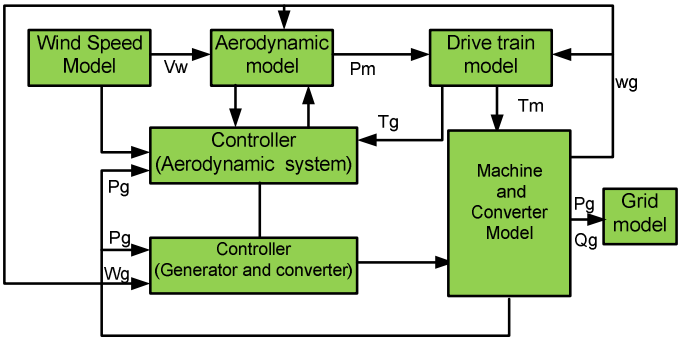


Fig. 2. Generic model of a DFIG

2.1 Aerodynamic Torque Model

Aerodynamic model involves the extraction of useful mechanical power from the available wind power. Available wind power is given by

$$P_{wind} = \frac{1}{2}\rho\pi R^2 V^3 \tag{1}$$

where P_{wind}, ρ, R and V are the available power in the wind, air density (kg/m^2), radius of the turbine blade (m) and the wind speed (m/s) that reaches the rotor swept area (m^2). The fraction of wind power that is converted to the turbine mechanical power P_m is given by

$$P_m = \frac{1}{2}\rho\pi R^2 C_p(\lambda,\beta) V^3 \tag{2}$$

where C_p gives the fraction of available wind power that is converted to turbine mechanical power, λ and β are the tip speed ratio and the pitch angle respectively. The C_p, λ and β are related by equation (3) and (4) [10]

$$C_p(\lambda,\beta) = c_1\left(\frac{c_2}{\lambda_i} - c_3\beta - c_4\right)e^{\frac{c_5}{\lambda_i}} + c_6\lambda \tag{3}$$

$$\frac{1}{\lambda_i} = \frac{1}{1+0.08\beta} - \frac{0.035}{\beta^2+1} \tag{4}$$

Given $c_1 = 0.5176$, $c_2 = 116, c_3 = 0.4, c_4 = 5, c_5 = 21$ and $c_6 = 0.0068$,the relationship between C_p against λ at various β is given in figure 3.

The tip speed ratio is given by (5)

$$\lambda = \frac{R\omega t}{V} \tag{5}$$

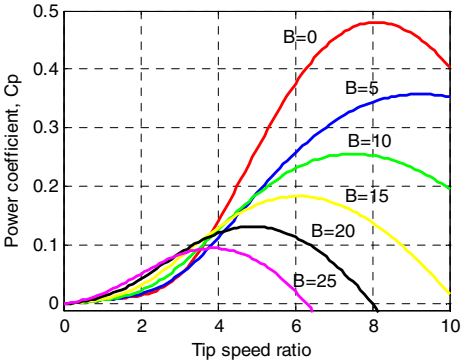


Fig. 3. Relationship between Power coefficient and tip speed ratio at different pitch angle

The mechanical torque (Nm) developed by the wind power is given by (6)

$$T_{ma} = \frac{P_m}{\omega_t} = \frac{\frac{1}{2}\rho\pi R^2 C_p(\lambda,\beta)V^3}{\omega_t} \tag{6}$$

where ω_t is the turbine speed.

2.2 Maximum Power Tracking of Variable Speed Wind Turbine

In the time when the wind speed is in the range of cut-in and rated value, the maximum aerodynamic power available in the wind can be capture. The maximum power in a mass of wind can be extracted by varying the turbine speed with the varying wind speed so that at all times it is on the track of the maximum power curve [6].

For efficient wind power captured by the variable wind turbine [11], $\lambda = \lambda_{opt}$, therefore (5) can be re-written as (7)

$$V = \frac{R\omega t}{\lambda_{opt}} \tag{7}$$

substituting (7) in (2), optimum power can be obtained as (8) which can be re-written as (9)

$$P_{opt} = \frac{1}{2}\frac{\rho\pi R^5 C_p(\lambda_{opt},\beta=0)\omega t^3}{\lambda_{opt}^3} \tag{8}$$

$$P_{opt} = k_{opt}\omega t^3 \tag{9}$$

where,

$$k_{opt} = \frac{1}{2}\frac{\rho\pi R^5 C_p(\lambda_{opt},\beta=0)}{\lambda_{opt}^3}$$

Fig 4 depicts the maximum power tracking of a variable speed wind turbine at different wind speed.

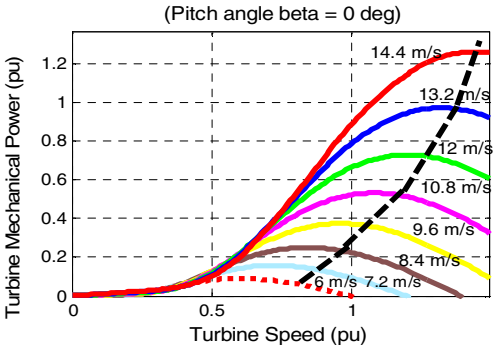


Fig. 4. Maximum torque tracking of a variable speed wind turbine

2.3 The Mechanical Shaft System Model

Adequate model of the mechanical drive train is required when the study involves the response of a system to heavy disturbances. It is better to represent the shaft by at least two- mass model [12] as show in Fig 5 where the turbine is coupled to the generator through a gearbox.

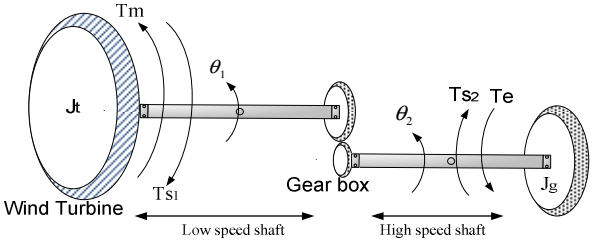


Fig. 5. Two mass model of the mechanical shaft system

From the figure, the following equations can be derived (10)-(17)

$$2H_t \frac{d\omega_t}{dt} = T_m - Ts_1 \tag{10}$$

$$2Hg \frac{d\omega_r}{dt} = Ts_2 - T_e \tag{11}$$

Where,

$$H_g = \frac{J_g \omega^2}{2n_p{}^2 P_g} \text{ and } H_T = \frac{J_T \omega^2}{2n^2 P_g} \tag{12}$$

$$Ts_1 = K_1\theta_1 + F_1\frac{d\theta_1}{dt} \text{ and } Ts_2 = K_2\theta_2 - F_2\frac{d\theta_2}{dt} \tag{13}$$

$$\theta_{eq} = \theta_1 - \theta_2, K_{eq} = \frac{K_1 * K_2}{K_1 + K_2} \text{ and } T_{eq} = \frac{Ts_1 * Ts_2}{Ts_1 + Ts_2} \tag{14}$$

$$T_{eq} = K_{eq}\theta_{eq} + F_{eq}\frac{d\theta_{eq}}{dt} \tag{15}$$

$$\frac{d\theta_{eq}}{dt} = \omega_t - \omega_r \tag{16}$$

$$\frac{\mathrm{d}\theta_{\mathrm{eq}}}{\mathrm{dt}} = \omega_{\mathrm{t}} - \omega_{\mathrm{r}} \tag{17}$$

where H_t, Hg are the pu turbine and generator inertia respectively. J_g and J_T are the inertia in kgm^2. T_e is the electromechanical torque developed by the induction generator, T_m is the pu mechanical torque applied to the turbine by the wind derived from (6). Ts_1, Ts_2, T_{eq} are the torques developed by the shaft at the low speed side, torque developed by the shaft at the high speed side and the equivalent torque developed by the shafts respectively. ω_t and ω_r are the pu turbine and generator rotor speed. K_1, K_2 and K_{eq} are shaft stiffness at low speed side, shaft stiffness at high speed side and the total shaft stiffness. F_1, F_2 and F_{eq} are the damping coefficient of the shaft at the low speed side, high speed side and the equivalent damping coefficient of the shaft respectively. θ_1, θ_2 and θ_{eq} are the angle of twist of the shaft at low speed, high speed and the equivalent angle of twist of the shaft respectively. n_p is the number of pole pairs, n is the gear ratio, P_g is the generator active power, ω is $2\pi f$ where f is the frequency (Hz).

2.4 Pitch Angle Controller Model

Pitch angle controller mainly serves the purpose of limiting the generated power to the rated power at time of high wind speed. It also limits the speed of the generator during heavy disturbances. The pitch controller based on PI is given by (18) [13]

$$\begin{aligned} \frac{d\beta}{dt} &= \frac{1}{\tau_s}\left(\beta_{ref} - \beta\right) \\ \beta_{ref} &= \left(k_P + \frac{k_i}{s}\right)\left(P_{ref} - P_m\right) \end{aligned} \tag{18}$$

where β_{ref} is the reference pitch control, k_P and k_i are the proportional and integral parameters of the PI controller, P_{ref} is the reference turbine power.

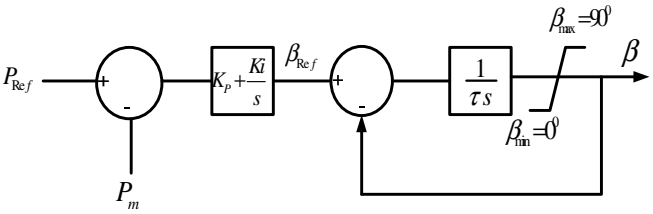


Fig. 6. Pitch angle controller

2.5 Wind Generator Model

Most wind farms are made of induction generators because they are cheap and robust. The dq stator and rotor voltage equations model in generating mode are as follows [14, 15].

$$v_{qs} = -r_s i_{qs} - \omega\lambda_{qs} - p\lambda_{qs} \tag{19}$$

$$v_{ds} = -r_s i_{ds} + \omega\lambda_{qs} - p\lambda_{ds} \tag{20}$$

$$v_{qr} = -r_r i_{qr} - (\omega - \omega_r)\lambda_{qr} - p\lambda_{qr} \tag{21}$$

$$v_{dr} = -r_r i_{dr} + (\omega - \omega_r)\lambda_{qr} - p\lambda_{dr} \tag{22}$$

where r_s , r_r are the stator and rotor speed resistance, p is $\frac{d(.)}{dt}$ term.

The equation presented in (19)–(22) is a fifth order model. Third order model is obtained by neglecting the transient term in the stator voltage equation. The stator and rotor flux equations are

$$\lambda_{qs} = L_s i_{qs} + L_m i_{qr} \tag{23}$$

$$\lambda_{ds} = L_s i_{ds} + L_m i_{dr} \tag{24}$$

$$\lambda_{qr} = L_r i_{qr} + L_m i_{qs} \tag{25}$$

$$\lambda_{dr} = L_r i_{dr} + L_m i_{ds} \tag{26}$$

where L_s , L_r , L_m are the stator, rotor and magnetizing inductance respectively. i_{ds} , i_{qs} i_{dr} and i_{qr} are the stator and rotor d-axis and q-axis current.

The electromechanical torque, T_e developed by the induction generator in pu can be derived as (27)

$$T_e = \frac{1}{\sigma}\left(\lambda_{qs}\lambda_{dr} - \lambda_{qr}\lambda_{ds}\right) \tag{27}$$

where

$$\sigma = 1 - \frac{L_m^2}{L_r L_s}$$

The equation is completed by the mechanical coupling equation in pu between the turbine and the generator using two mass model as derived in (10)–(15)

$$\frac{d\omega_r}{dt} = \frac{1}{2H_g}\left(Ts_2 - T_e\right) \tag{28}$$

the active and reactive power generated by the induction generator is given as

$$P_s = \frac{3}{2}\left(v_{qs}i_{qs} + v_{ds}i_{ds}\right) \tag{29}$$

$$Q_s = \frac{3}{2}\left(v_{qs}i_{ds} - v_{ds}i_{qs}\right) \tag{30}$$

2.6 Grid Connection of DFIG

DFIG technology makes use of wound rotor. The stator is directly connected to the grid while the rotor is coupled to the grid through a PWM) frequency converter as shown in Fig. 7.

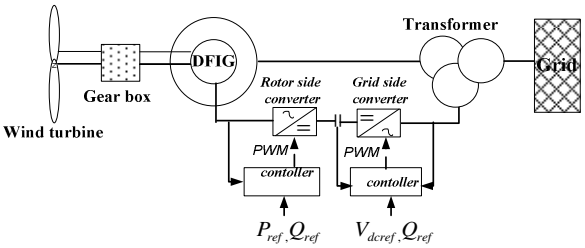


Fig. 7. DFIG with PWM converter control system

For dynamic study of DFIG, the converter controller model is important. Stator flux oriented control is commonly used in the decoupled control of DFIG.

2.7 DFIG Rotor Side Converter Controller

The control of the DFIG rotor is done in a synchronous rotating reference frame i.e. $\omega = \omega_e$ in equation (18)-(21). The rotor side converter controls the stator active and reactive power of the DFIG. By aligning the d-q reference frame in the stator flux reference frame as in figure 8 [16], then $v_{ds} = 0$ $v_{qs} = v_s$, $\lambda_{ds} = \lambda_s$ and $\lambda_{qs} = 0$.

From (25) and (26)

$$i_{qs} = -\frac{L_m}{L_s} i_{qr} \tag{31}$$

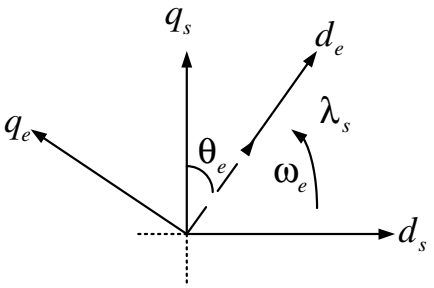


Fig. 8. Stator flux vector orientation along the rotating reference frame

Substituting (31) in (29) and (30) with vector control and re-arranging, we obtain (32) and (33)

$$P_s = -\frac{3}{2}\frac{L_m}{L_s} v_s i_{qr}^* \tag{32}$$

$$Q_s = \frac{3}{2L_s}\left(\frac{v_s^2}{\omega_s} - v_s L_m i_{dr}^*\right) \tag{33}$$

The rotor voltage equation governing the active and reactive power control can be obtained by rearranging equation (19)–(26) and is given by (34) and (35)

$$v_{dr} = r_r i_{dr} + \sigma L_r \frac{d}{dt} i_{dr} - (\omega_e - \omega_r)(\sigma L_r i_{qr}) \tag{34}$$

$$v_{qr} = r_r i_{qr} + \sigma L_r \frac{d}{dt} i_{qr} + (\omega_e - \omega_r)\sigma L_r i_{dr} + (\omega_e - \omega_r)\frac{L_m}{L_s}\lambda_{ds} \tag{35}$$

Where

$$\sigma = 1 - \frac{L_m^2}{L_r L_s}$$

Equations (34) and (35) can be rewritten as (36) and (37) to form the decoupled control of active and reactive power.

$$v_{dr}^* = \left(k_{dp} + \frac{k_{di}}{s}\right)\left(i_{dr}^* - i_{dr}\right) - \left(\omega_e - \omega_r\right)\sigma L_r i_{qr} \tag{36}$$

$$v_{qr}^* = \left(k_{qp} + \frac{k_{qi}}{s}\right)\left(i_{qr}^* - i_{qr}\right) - \left(\omega_e - \omega_r\right)(\sigma L_r i_{dr} - \frac{L_m}{L_s}\lambda_s) \tag{37}$$

where, k_{dp}, k_{di} are the PI proportional and integral constant for the d-axis for the control of reactive power while gain k_{qp}, k_{qi} are the PI constant for controlling the active power. i_{qr}^* and i_{dr}^* are the reference current for the active and reactive power

respectively. v_{dr}^{*} and v_{qr}^{*} are the d-q reference voltage which will be converted to a-b-c frame to generate command for the rotor end PWM converter. The block diagram is shown in figure 9.

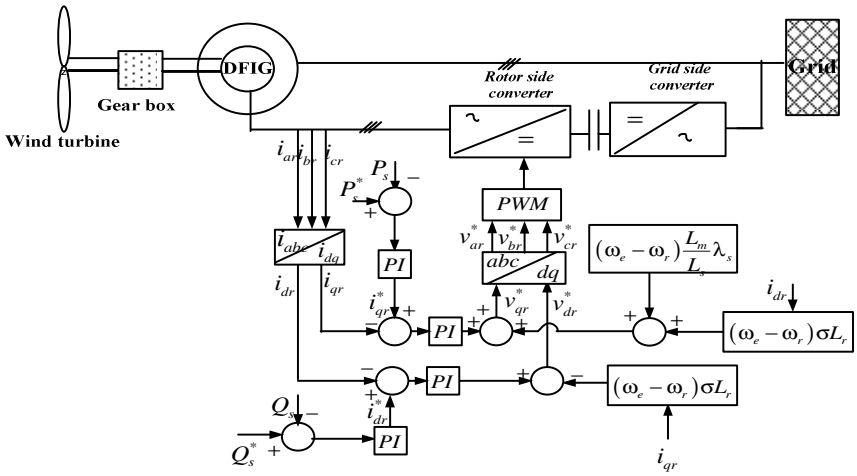


Fig. 9. Rotor side Controller system

2.8 DFIG Grid Side Converter Controller

The main objective of grid side controller is to maintain the dc link between the back to back PWM converters at constant voltage irrespective of the direction of power flow [14]. The voltage for the grid side converter is represented by (38) [17]

$$\begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} = r \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + L\frac{d}{dt}\begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + \begin{bmatrix} v_{a1} \\ v_{b1} \\ v_{c1} \end{bmatrix} \tag{38}$$

The d-q transformation of equation (38) yields (39)

$$\begin{aligned} v_q &= Ri_q + L\frac{di_q}{dt} + \omega_e L_{id} + v_{q1} \\ v_d &= Ri_d + L\frac{di_d}{dt} - \omega_e L_{iq} + v_{d1} \end{aligned} \tag{39}$$

Re-arranging (39) with $v_{qs} = 0$, the governing voltage equation for the grid side converter can be obtained as (40)

$$\begin{aligned} v_{q1}^{*} &= -\left(k_{1p} + \frac{k_{1i}}{s}\right)\left(i_q^{*} - i_q\right) - \omega_e L_{id} \\ v_{d1}^{*} &= -\left(k_{2p} + \frac{k_{2i}}{s}\right)\left(i_d^{*} - i_d\right) + \omega_e L_{iq} + v_d \end{aligned} \tag{40}$$

where k_{1p} , k_{1i} are the q axis PI proportional and the integral constant k_{2p} , k_{2i} are the d axis PI proportionality and integral constant respectively. v_{q1}^{*} and v_{d1}^{*} are the reference voltages that generate the command for the grid side PWM converter after conversion to abc frame. i_q^{*} is derived from the grid reactive power error while i_d^{*} is derived from the dc link voltage error as shown in Fig.10.

Applying the grid voltage oriented control i.e. aligning the d axis of the reference frame along the grid voltage vector, then $v_q = 0$.

Hence, active and reactive power can be written as (41) and (42)

$$P_g = \frac{3}{2} v_d i_d \tag{41}$$

$$Q_g = -\frac{3}{2} v_d i_q \tag{42}$$

The reactive power can be controlled by the i_q of the grid side converter. The energy stored in the dc link can be written as (43).

$$\frac{1}{2} c v_{dc}^2 \tag{43}$$

Where c and v_{dc} are the dc link capacitor and voltage respectively

$$c v_{dc} \frac{d}{dt} v_{dc} = P_g - P_r \tag{44}$$

$$c \frac{d}{dt} v_{dc} = \frac{P_g}{v_{dc}} - \frac{P_r}{v_{dc}} = i_{og} - i_{or} \tag{45}$$

where

$$i_{og} = \frac{3}{2} \frac{v_d}{v_{dc}} i_d$$

$$\frac{v_d}{v_{dc}} = \frac{m}{\sqrt[2]{2}} \tag{46}$$

where m is the modulation factor which gives the ratio of grid voltage to the dc bus voltage [18].

$$c \frac{d}{dt} v_{dc} = \frac{3m}{\sqrt[4]{2}} i_d - i_{or} \tag{47}$$

Hence, i_d can be used for the control of dc bus voltage.

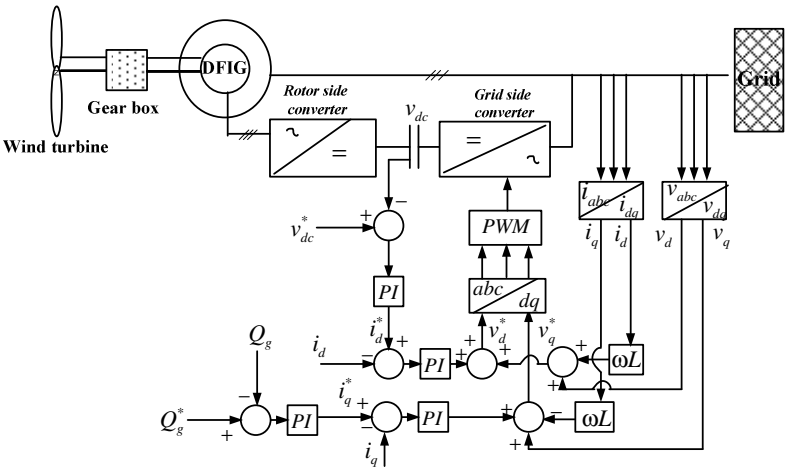


Fig. 10. Grid side controller system

2.9 Protection of Converter (Crowbar Protection)

High currents due to a fault close to the generator can damage the rotor side converter. To avoid any damages, the rotor side converter is bypassed when the rotor current exceeds a predetermined limit. To achieve this, an additional resistance otherwise known as "crowbar" is connected to the rotor circuit. The thyristors are turned on when the rotor current exceeds its preset value. The rotor circuits are then short-circuited by the crowbar and it shunts away the rotor overcurrent. Fig 11 shows a crowbar protection system where r_{cr} is the additional resistance added for crowbar protection. The rotor remains connected to the crowbar until the fault is cleared.

When crowbar protection is initiated as a result of rotor current exceeding the set limit, then $v_{dr} = v_{qr} = 0$, hence the DFIG operates as normal singly fed induction generator.

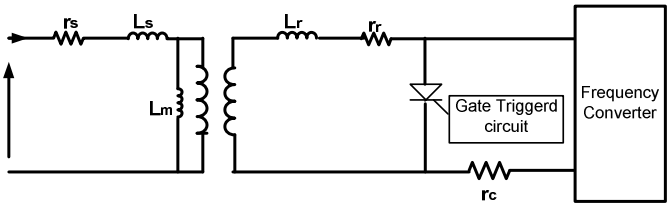


Fig. 11. Crowbar protection circuit

From the diagram the rotor voltage equation (21) and (22) will be modified to (48) and (49)

$$0 = (r_r + r_c) i_{dr} + \frac{d}{dt}\lambda_{dr} - (\omega_e - \omega_r)\lambda_{qr} \tag{48}$$

$$0 = (r_r + r_c) i_{qr} + \frac{d}{dt}\lambda q_r - (\omega_e - \omega_r)\lambda_{dr} \tag{49}$$

2.10 Wind Farm Model

Combinations of several WCS constitute a WF and simulation of complete wind farm with large numbers of WCS will be computationally intensive without much difference in the assessment. Aggregate model reduces simulation time required by detailed multi turbines system [19-21].The objective is to represent a large wind farm with many WCS by a single turbine system[22]. The following criteria must be fulfilled when reducing a large wind farm into an aggregate model assuming a regular wind distribution [19, 22]

1. The MVA rating of the equivalent WF S_{wf} is the sum of individual MVA rating of the WCS

$$S_{wf} = \sum_{i=1}^{n} S_i \tag{50}$$

where S_i is the MVA rating of WCS, i and n are the numbers of WCS in the WF

2.

$$P_{wf} = \sum_{i=1}^{n} P_i \tag{51}$$

Where P_{wf} is the electric power supplied by the equivalent WF, P_i is the electric power supply by i^{th} WCS.

3. The dynamics of the wind generators are given by the slope of the $P-Q$ characteristics of the induction generator.

$$\frac{dQ}{dP_{wf}} = \sum_{i=1}^{n} \frac{dQ}{dP_i} \tag{52}$$

Where dQ/dP_{wf} is the $P-Q$ characteristic of the equivalent WF and dQ/dP_i is the i^{th} $P-Q$ characteristic of the WCS.

3 System under Study

The system considered for the study is shown in Fig. 12. It consists of 110MW, 50MVAR synchronous generator (SG) connected to bus 4 through a 20/400kV transformer.

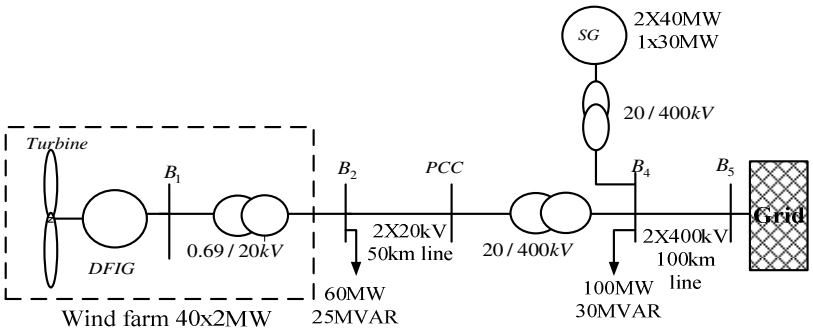


Fig. 12. The system under study

The wind farm (WF) is made up of 40 wind turbines of 2MW, 0.69kV each modelled as an aggregate wind turbine. It is assumed that the wind farms are located far from the point of common connection (PCC) where the wind resources are abundantly located as the case for most real wind farms. The WF is connected to the PCC through two 20km line (to allow disconnection of a line) and 69/20KV transformer. The WF is feeding a 60MW, 25MVAR local load connected to bus2 (B2). Another 100MVA, 30MVAR load is connected to the high voltage bus (B4). The whole system is connected to a strong grid through a two 400kV, 100km transmission lines.

4 The Simulation Results and Discussions

Different scenarios were created to get an insight into the response of WF to disturbances from the grid. First, the response of the wind farm was studied when there is a step change of 20% in the local load connected to B_2 at 1s. The results with the rotor controller in place and out of place are depicted in Fig 13. From the figure it can be observed that with the controller in place, the active power (the negative values indicate a power injected into the grid) and the electrical torque are immediately returned to the pre-disturbance level. The step increase in the local load resulted in a dip in the terminal voltage and an increase in the speed of the generator; however, it stabilizes to a new value almost immediately. This is as a result of a change in the system configuration. With the rotor controller out, the system is stable but it takes about 3s for the wind farm to stabilize.

Fig. 14 shows the response of i_{dr} , i_{qr} , crowbar protection and the pitch angle (β) to a 20% step increase in local load. With this disturbance, the maximum i_{dr} current reaches 0.71pu from the prefault value of 0.65pu. The crowbar protection was set to operate at 1.5pu and therefore could not be inserted as the rotor current is lower than the crowbar predetermined value. The pitch angle controller acts to limit the speed of the generator as a result of the disturbance.

The response of the wind farm to a 3 phase fault of 200ms duration was investigated. The fault was created at 1s at the middle of 100km, 400kV line. The results of the response of the DFIG speed, the electrical torque, voltage at the point of common connection (PCC) and the pitch controller are presented in Fig 15. The speed of the generator is limited by the pitch angle. The first swing of the DFIG speed reached a value of 1.26pu from the prefault value of 1.21pu. The fault causes a dip in voltage at the PCC, the pitch controller acts to stabilize the speed of the generator.

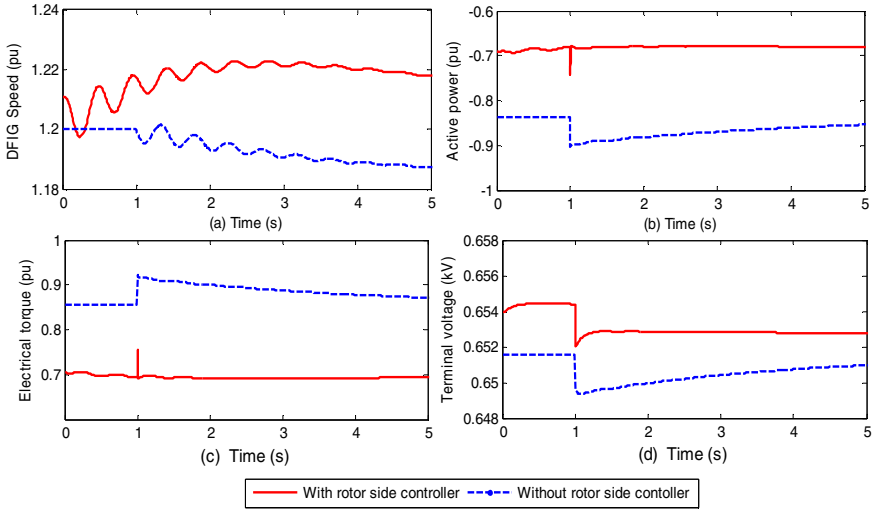


Fig. 13. The response of (a) DFIG speed (b) Active power, (c) Electrical torque (d) Terminal voltage to 20% step change in local load at 1s

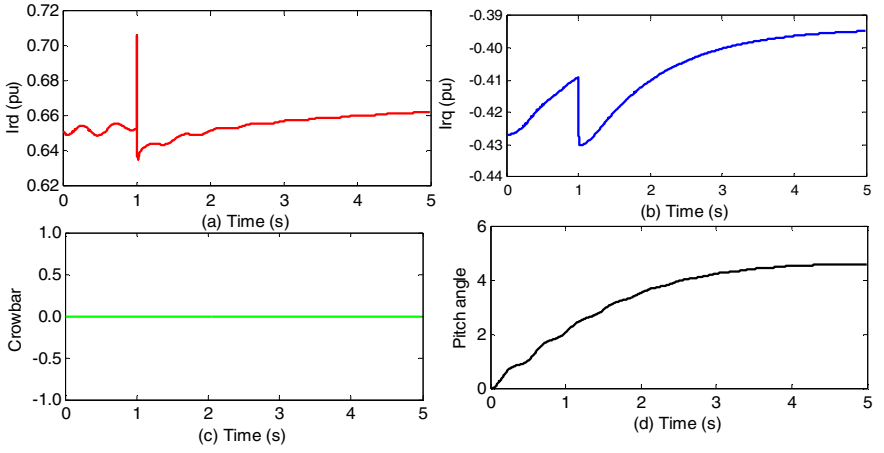


Fig. 14. The response of (a) Rotor d-axis current (b) Rotor q-axis current (c) Crowbar protection (d) Pitch controller when a step change of 20% is initiated in the local load at 1s

Figure 16 depicts the rotor d-axis current, rotor q-axis current, the rotor current and the crowbar protection. The rotor current reached 2pu during the fault which initiated the operation of the crowbar protection so as to prevent damage to the converter.

The response of the wind farm to different fault locations was examined. To get an insight into this scenario, a three phase fault of 200ms duration was created at different locations on the 50km, 20kV line. The result is shown in Fig. 17. From the result, the impact of fault at different locations has almost the same impact on the response of the wind farm. However, the impact is visibly different at the PCC. The closer the fault location to the PCC, the more the dip in voltage and the more the deviation from the nominal grid frequency.

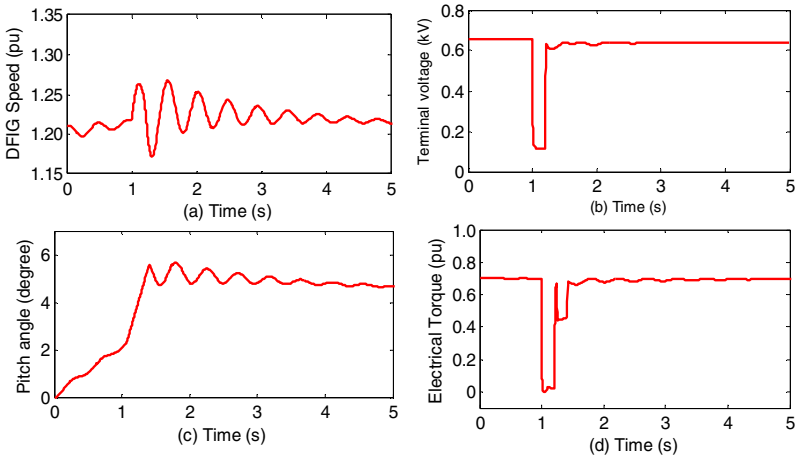


Fig. 15. Response of the Wind farm (a) speed (b) PCC voltage (c) pitch controller (d) Electrical torque when a three phase fault of 200ms duration is created at 1s at the middle of 100 km, 400kV line

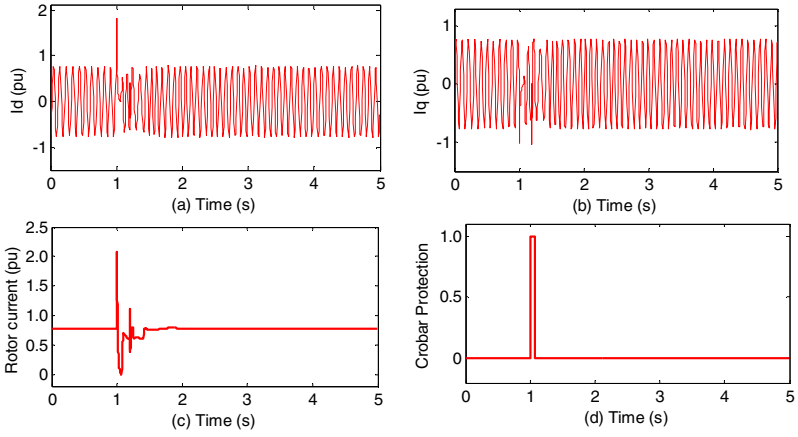


Fig. 16. (a) The rotor d-axis current (b) rotor q-axis current (c) the rotor current (d) the crowbar protection when a three phase fault of 200ms duration is created at 1s at the middle of 100 km, 400kV line

Fig. 18 shows the response of the wind generator with different rotor inertia to a three phase fault created at the middle of 400kV line. The effect of inertia can be noticed in the speed of the generator. The generators with larger inertia are more stable in case of fault compared to the generator with smaller inertia. The first swing in rotor speed for 50kgm^2 is 1.28pu, 1.26pu for 100kgm^2, 1.24pu for 150kgm^2 and 1.22pu for 200kgm^2. No distinct differences in the response of the active power, electrical torque and the terminal voltage are seen.

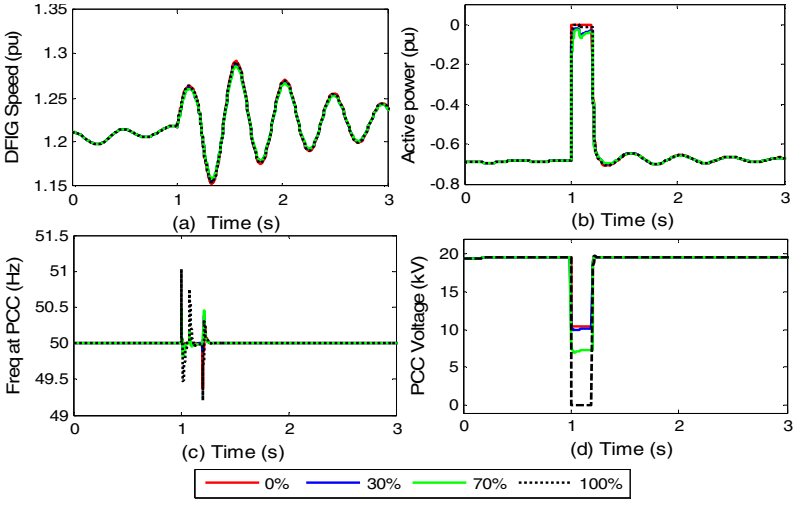


Fig. 17. Response of (a) DFIG speed (b) DFIG active power (c) PCC frequency and (d) PCC voltage to 3 phase fault (200ms duration) at different locations on the 20kV line

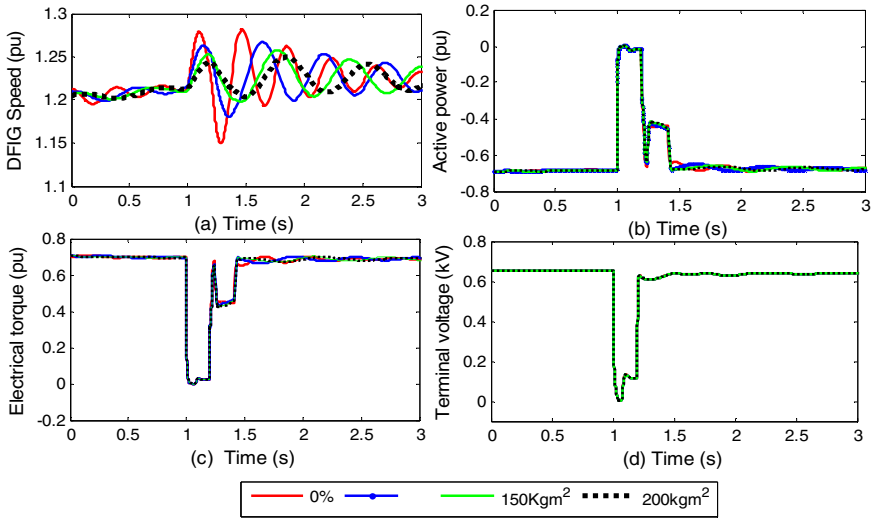


Fig. 18. Wind farm with DFIG of different inertia

The effect of a loss of transmission line (TL) and generation on the behaviour of the WF was studied. For the TL, the circuit breakers at both ends of the lines were opened at 1s for the 400kV, 100km line and then for 20kV, 50km line in turn. The circuit breaker at bus 4 connecting the synchronous generator (SG) to the grid was opened at 1s to disconnect the SG from the power system. The results are shown in Fig 19. A loss of line causes a surge in the system frequency at the PCC; this caused a reduction of active power to the network by the WF to restore the frequency to the prefault value. The 20kV, 50km line has a severe impact compared to the 400kV, 100km line due to close proximity to the WF. At the instant the SG (generation) was lost; a sudden dip in the system frequency was experienced, this in turn resulted into an instant injection of active power from the WF to the grid to restore the system frequency.

The terminal voltage reduces from the prefault value of 0.655kV to a new value of 0.638kV, 0.641kV and 0.650kV for the loss of 50km line, 100km line and SG respectively as a result of change in the system configuration.

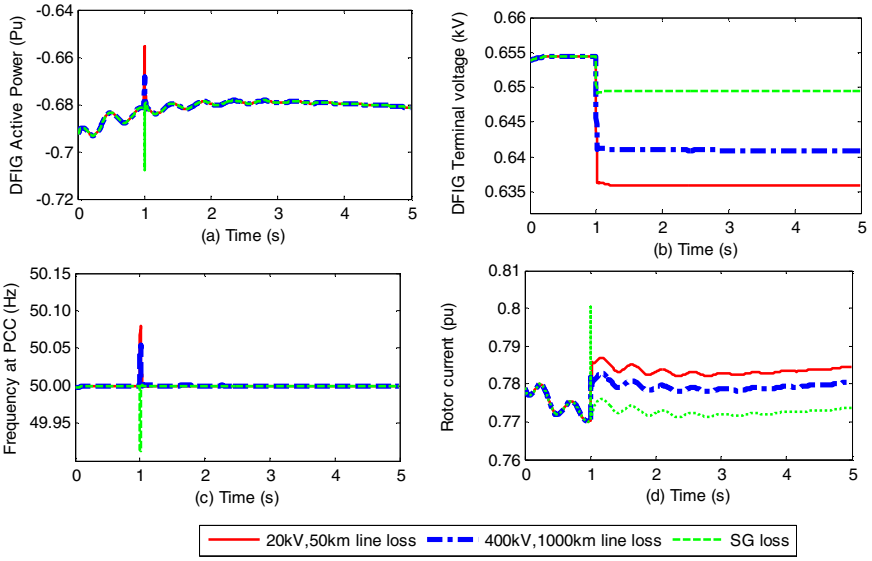


Fig. 19. Response to loss of transmission line

5 Conclusions

The behaviour of a wind farm consisting of DFIG in response to different disturbances emanating from the power system has been studied. From the study, the effect of the rotor controller on the stability of a wind farm has been shown to be significant to the stability of the wind farm following a disturbance. Without controller, prefault condition was achieved after about 3s. With a controller, the prefault condition was achieved almost immediately.

The location of the fault occurrence is seen to have little effect on the wind farm. However the location of fault occurrence has significant effect on the frequency and the voltage at the PCC.

The inertia of wind generators has influence on the response of the WF to a disturbance. The larger the inertia the lower the magnitude of oscillation of the generator speed. A larger inertia enhances good stability. The WF responds to the sudden loss of transmission line and generation in such a way as to restore the system frequency. The rotor current and the terminal voltage assume a new value due to the change in the network configuration.

This paper is useful to the utility operators in understanding the probable response of a wind farm during disturbance in the power system. However, a qualitative study mainly is carried out on a small test system. Further investigation is necessary for a large power system.

References

1. GWEC, Global Wind Statistic (2010), http://www.gwec.net
2. El-Sayed, M.A.: Integrating Wind Energy into Weak Power Grid Using Fuzzy Controlled TSC Compensator. In: International Conference on Renewable Energies and Power Quality (ICREPQ 2010), Granada, Spain (2010)
3. Slootweg, J.G., De Haan, S.W.H., Polinder, H., Kling, W.L.: Modeling wind turbines in power system dynamics simulations. In: Power Engineering Society Summer Meeting, vol. 1, pp. 22–26. IEEE (2001)
4. Xing, Z., Zheng, Q., Yao, X., Jing, Y.: Integration of Large Double-Fed Wind Power Generator System into Grid. In: 8th International Conference on Electrical Machines and System, pp. 1000–1004 (2005)
5. Veganzones, C., Martinez, S., Blazquez, F.: Large Scale Integration of Wind Energy into Power Systems. Electrical Power Quality and Utilisation, Magazine 1, 15–22 (2005)
6. Seul-Ki, K., Eung-Sang, K., Jae-Young, Y., Ho-Yong, K.: PSCAD/EMTDC based dynamic modeling and analysis of a variable speed wind turbine. In: IEEE Power Engineering Society General Meeting, vol. 2, pp. 1735–1741 (2004)
7. Eping, C., Stenzel, J., Poller, M., Muller, H.: Impact of Large Scale Wind Power on Power System Stability. In: 5th International Workshop on Large-Scale Integration of Wind Power and Transmission Networks for Offshore Wind Farms, Glasgow, Scoland (2005)
8. Folly, K.A., Sheetekela, S.: Impact of fixed and variable speed wind generators on the transient stability of a power system network. In: Power Systems Conference and Exposition, PSCE 2009, pp. 1–7. IEEE/PES (2009)
9. Naimi, D., Bouktir, T.: Impact of Wind Power on the Angular Stability of a Power System. Leonardo Electronic Journal of Practices and Technologies 12, 83–94 (2008)
10. El-Sayed, M.A., Adel, M.S.: Wind Energy-Grid Stabilization using a Dynamic Filter Compensator. In: International Conference on Renewable and Power Quality (ICRPQ 2010), Spain (2010)
11. Arifujjaman, M.D., Iqbal, M.T., Quaicoe, J.E.: Vector Control of a DFIG Based Wind Turbine. Journal of Electrical & Electronics Engineering 9, 1057–1066 (2009)
12. Poller, M.A.: Doubly_FEd Induction Machine Models for Stability Assessment of Wind Farms, http://www.digsilent.de/consulting/publication/DFIGmodelling (date accessed September 28, 2009)

13. El-Sattar, A.A., Saad, N.H., El-Dein, M.Z.S.: Dynamic Response of Doubly Fed Induction Generator Variable Speed Wind Turbine Under Fault. Electric Power Systems Research 78, 1240–1246 (2008)
14. Krause, P.C., Wasynczuk, O., Sudhoff, S.D.: Analysis of Electric Machinery and Drive Systems Second Edition. A John Wiley and Sons, Inc. Publication (2002)
15. Lipo, T.A.: Electric Machine Analysis and Simulation, Wisconsin Power Electronic Research Center. University of Wisconsin-Madison, Madison (2000)
16. Simoe, M.G., Farret, F.A.: Renewable Energy Systems: Design and Analysis with Induction Generators. CRC Press, Boca Raton (2004)
17. Soares, O.M., Gocalves, H.N., Martins, A.P., Carvalho, A.: Analysis and NN-Based Control of Doubly Fed Induction Generator in Wind Power Generation. In: International Conference on Renewable Energies and Power Quality (ICREPQ 2009), Valencia, Spain (2009)
18. Salman, S.K., Badrzadeh, B.: New Approach for Modelling Doubly-Fed Induction Genertaor (DFIG) for Grid-Connected Studies. In: European Wind Energy Conference and Exhibition, pp. 1–13 (2004)
19. Akhmatov, V.: An Aggregate Model of a Grid-connected, Large-scale, Offshore Wind Farm for Power Stability Investigations- Importance of Windmill Mechanical system. Electric Power and Energy Systems 24, 709–717 (2002)
20. Conroy, J., Watson, R.: Aggregate Modelling of Wind Farms Containing Full-Converter Wind Turbine Generators with Permanent Magnet Synchronous Machines. Transient Stability Studies. IET Renewable Power Generation 3, 39–52 (2009)
21. Poller, M., Achilles, S.: Aggregated Wind Park Models for Analyzing Power System Dynamics. In: 4th International Workshop on Large- Scale Integration of Wind Power and Transmission Networks for Off-shore Wind Farms, Billund, Denmark (2003)
22. Zhao, S., Nair, N.C.: Behavior of Doubly - Fed Induction Generator Unit during System Disturbance. In: Australasian Universities Power Engineering Conference, AUPEC 2008, pp. 1–6 (2008)

Educational Simulators for Industrial Process Control

L.F. Acebes[1], A. Merino[1], L. Gómez[1], R. Alves[2], R. Mazaeda[1], and J. Acedo[3]

[1] Department of Systems Engineering and Automatic Control, University of Valladolid, Higher Tech. College of Industrial Engineering, c/Real de Burgos s/n 47011, Valladolid, Spain
[2] Department of Informatics and Automatic Control, Faculty of Sciences, University of Salamanca, Spain
[3] Centro Superior de Formación de Repsol (CSFR), Madrid, Spain
felipe@autom.uva.es

Abstract. The paper shows a Windows© NT/XP/7 application oriented to learn control skills to process engineers. It is a dynamic simulation based tool with a friendly user interface that contains two sets of diverse process control problems (more than twenty study cases are available). It is possible to study typical control problems as cascade, ratio, selective, override and feedforward control techniques and the tuning, configuration and operation of PID controllers. Additionally, it allows analyzing complex control systems installed in boilers, furnaces, distillation columns or reactors and special industrial control techniques to ensure the process safety. In order to outline the functional features of the tool, one of the simplest modules is shown. To conclude, an overview of the methodology and software used to develop this tool is also outlined. In particular, an object oriented modeling and simulation tool is used to develop the simulation models, a self-developed SCADA is used as graphical user interface and the simulation-SCADA communications are supported by the OPC standard. Finally, it must be remarked that this tool is used successfully in an industrial master of instrumentation and process control.

Keywords: Dynamic Simulators, Continuous Process Control, Learning, OPC, Object oriented Modelling Languages.

1 Introduction

In nuclear, power, thermal, oil, gas, petrochemical, pulp and paper plants, as well as in other sectors, the use of process simulators is widespread, both for operators training and for production process improvement. Some examples of training simulators are [1-6]. These simulators are oriented to the operators training in particular industries and they are so much complex and high cost ones.

There are simulators oriented to the study of certain control subjects such as Looppro [7] or Topas [8]. They are good tools to learn process control, but many advanced aspects of the industrial implementations are not considered. However, one advantage is that are not so expensive.

Other simulation packages, the so called design simulators, are oriented to build the process and control structure model and experiment with it. One example in the field of engineering process is Hysys [9]. Other examples of general purpose

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 151–163.
DOI: 10.1007/978-3-642-34336-0_10

modeling and simulation tools are Dymola [10] or EcosimPro [11]. These modeling and simulation packages require that the user has a deep knowledge about them. Modeling and simulation skills are necessaries, especially in some cases in which he should develop their own model libraries. Besides, for training purposes, the experimental frame is not the more suitable one and also its price is high.

These tools pursue different objectives ranging from PID controllers tuning, process identification, design of process and control structures, study of advanced control strategies, operation of process unit and, even, complete industrial processes. Some of them are reduced to a single industrial field and other ones cover a reduced number of processes. Some aspects of interest in the training of process control engineers cannot be covered by any of them. For instance: some special control aspects related to process safety, as anti surge mechanism in centrifugal compressors; special processes, as blending processes; or parameterization procedures, as the linearization of the static operation curves of valves. In addition the graphical user interfaces (GUIs) are different ones, both in appearance and functionality.

So, to give a complete training to a control engineer requires the use of different tools that use dynamic simulation. This implies a high economic cost to the institution that provides training, both for the licenses purchase as for maintaining and updating them. For the students, it means an effort to adapt and learn different tools, some of which have many features that are not used by the students and they are being paid by the institution offering the training.

For these reasons, a simulation tool oriented to study typical problems of operation and control in production units of the process industry has been developed. The modules have been carried out by the Department of Systems Engineering and Automatic Control of the University of Valladolid and they are based on the expertise of control and instrumentation engineers of Repsol (a Spanish company in which one of its main activities is the production of petroleum derivates). This tool is being used in the "Master in instrumentation and process control ISA-REPSOL") given by the CSFR ("Superior Training Center of Repsol").

The paper describes the mentioned tool. In particular, a simulation module will be shown as an example. Afterwards, the software structure of the simulation tool is detailed, as well as, the software used for its development.

2 Tool Description

The mentioned tool is a Windows® NT/XP/7 application that allows selecting a set of simulation modules organized in two graphical main menus: "Control techniques" and "Process control units". The tool provides a complete help that explain each module in Spanish language. However, if more information is required, the great majority of the study cases are well explained in [12-13]. In order to give a general idea of the training capacity of the developed modules, these are listed and briefly described.

The "Control techniques" modules are:

- Ratio Control. Two options comparative for products mixture.
- Cascade control. Level tank control using cascade controllers. Mainly the tuning and the switching of the manual and automatic mode of the nested controllers are outlined.

- Selective control. Two cases, case 1 is a compression station control and case 2 is a pumping station control. Particular interest in the anti reset windup mechanism is shown.
- Feed forward control. Temperature control comparative in a heat exchanger without feedforward compensator and with a static or dynamic one.
- Split range control. This technique is applied to three systems. A pressure control in a distillation column head, a pressure control in a blanketing and a simultaneous flow and temperature control.

Fig. 1. "Process control units" menu

The “Process control units” modules are (Fig. 1):

- Steam production boiler. Level and pressure control (ten coupled control loops).
- Boiler Burner Management System (BMS). A security system to monitor the boiler and execute, in a safe way, all scripts to turn on and off the burners
- Exothermic chemical reactor. Hydrodesulphurization process control.
- Endothermic chemical reactor. Catalytic re-forming process control.
- Furnace. Temperature control (eight coupled control loops) driving two combustibles (fuel and gas).
- Distillation column. Two control structures and study of the economic and control aspects integration.
- Blending. The Blending is a batch process whose aim is to mix different components in appropriate proportions to meet a required specification. The simulated blending manages five components according to a mixture prescription that is defined by the user.

- Automatic valves. To study the importance of the control valves, the simulated system allows selecting the inherent characteristics of the valves and characterizing its digital smart positioners.
- Heat exchangers. The module aims to study the beneficial effect of using feedforward compensators and cascade control in simple systems such as heat exchangers.
- Centrifugal compressors. This module shows a control structure that prevents centrifugal compressors can enter in an unstable operation region:"anti-surge" mechanism.
- Alternative compressors. The simulated process is composed by an alternative compressor and a recycling system which aim is to compress all the gas that reaches the system. To avoid problems in the working of the compressor, the suction pressure is controlled using this recycled gas flow and the compressor load. Two control techniques can be compared: load steps or split-range control.
- Centrifugal pump. Minimum recycling control of centrifugal pump.
- ON-OFF level controller. The simulated system allows to study the well-known on-off control and pays attention in the hysteresis effect of the dead band and the logic of the controller.

When a module is selected, from one of the main menus, the corresponding dynamic simulator and its graphical user interface (GUI) are started. Later, some details about the simulation are given. Now, the GUI of each module will be the focus of attention.

Each GUI of the selected module corresponds to a P&ID [14], Piping and Instrumentation Diagram. These schematics have passive components that show information as standing or running equipment indicators, trend and historical charts, other types of charts (characteristic valve curves), value displays,... and active components that allow acting over the system: starting or stopping pumps, valves or process units; selecting automatic/manual/cascade mode in controllers; modifying the boundary condition and the process and control parameters,...

By default the simulation runs in real time, but the user can change the simulation run speed using a time scale factor that can be greater than 1 to accelerate it, if the PC allows it, or lower than 1 to decelerate it (Fig. 2). In this kind of process simulators is unusual to reduce the time scale factor because the system dynamic is slow or not so fast.

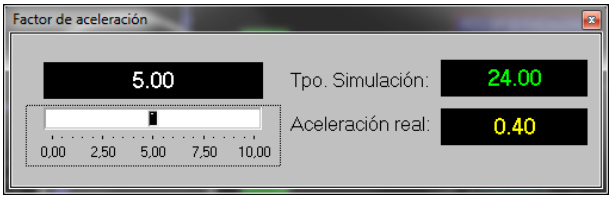


Fig. 2. Time factor

3 Example

In order to show how works the simulation tool, one of the simplest modules has been selected (minimum recycling control of centrifugal pump). First, a physical system description and the GUI will be outlined. Second, an experiment is run.

The system (Fig. 3) is composed by a tank that receives a flow of water, a centrifugal pump connected to the tank outlet, a recirculation valve (V1) and an outlet valve (V2). The level controller (LC1) output is connected, in cascade, with the flow controller (FC2). The FC2 output drives V2. At the pump outlet, there are two pipes; one is connected to V2 and the other one to V1. The water can be sent back to the tank and this flow (FI4) is governed by the flow controller (FC1) that drives V1.

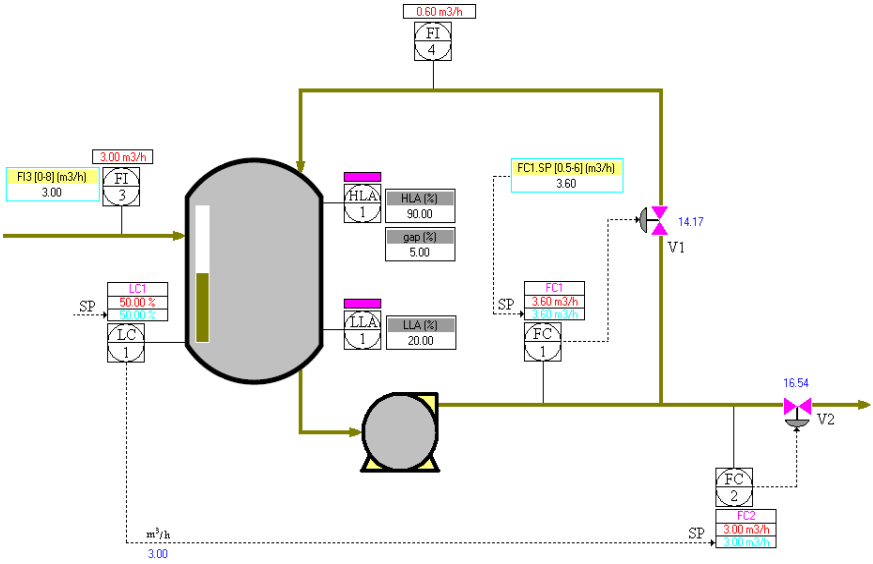


Fig. 3. Minimum recycling control of centrifugal pump

The aim of the control structure is ensure that, regardless of the LC1 actions, the pumped flow (FC1) must be always greater that a minimum value in order to avoid both thermal, mechanical or electrical problems and the pump cavitation. So, the Set Point (SP) of FC1 is the minimum pumped flow, which is a manufacturer specification that can be changed by the user. All controllers are implemented by PIDs (Proportional, Integral and Derivative). The process disturbance is the external flow to the tank (FI3) and it is a boundary condition that can be modified by the user.

Besides, there are two alarm indicators at the P&ID scheme, one for high level (HLA1) and the other for low level (LLA1). The HLA1 indicator will be active and change its color when HLA1 is inactive and the level of the tank will be upper than 90% (HLA value), and HLA1 will be inactive and change it color again when HLA1 is active and the level will be minor than 85% (HLA-gap value). The values of HLA and gap can be modified by the user. The LLA1 indicator works in a similar way than the HLA1 indicator.

The module allows modifying the PIDs parameters and observing the control structure performance when the feed flow changes, in particular when it is less than the minimum pumped flow.

As it was previously mentioned, FI3 and FC1 SP can be modified using a step, ramp, oscillatory or random signal. The user must click on the corresponding

Fig. 4. Change boundary condition

indicator of the P&ID. So, an experiment is made in which a step from 5 to 3 m^3/h in FI3 signal is activated (Fig. 4).

Initially, the outlet and pumped flow are equals and greater than the minimum pumped flow (3.6 m3/h) and therefore V1 is closed and FI4 is zero. As FI3 step result, FC2 and FC1 will be under the minimum pumped flow and the controller of minimum flow must act. First, FI3 decreases from 5 to 3 m^3/h. The tank level decreased and LC1 acts decreasing FC2 SP. Consequently, the outlet and pumped flows (in Fig. 5, FI1 and FI2) decrease simultaneously. When the pumped flow (FI1) is under the minimum pumped flow, FC1 acts opening V1 and, as result, FI4, the tank level and FI1 are increased and the pumped flow raises the minimum pumped flow value.

Clicking on the control signals or variables displays, trend charts showing the performance of the control structure are shown. These charts can be configured by the user. Fig. 5 shows the flows performance and Fig. 6 the dynamic of the control signals.

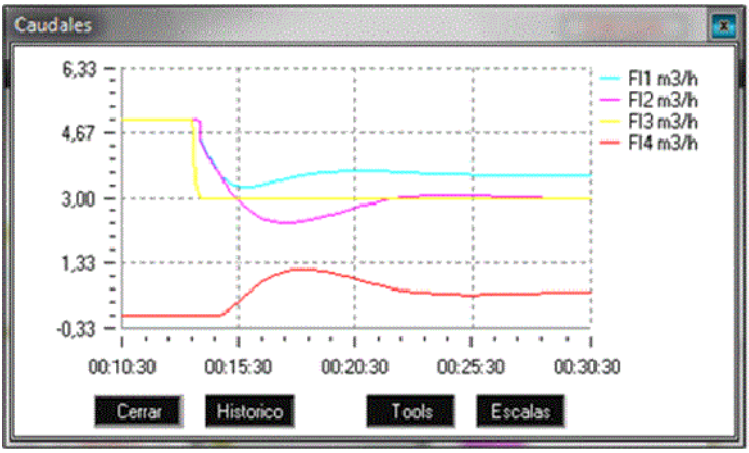


Fig. 5. Process and control structure response

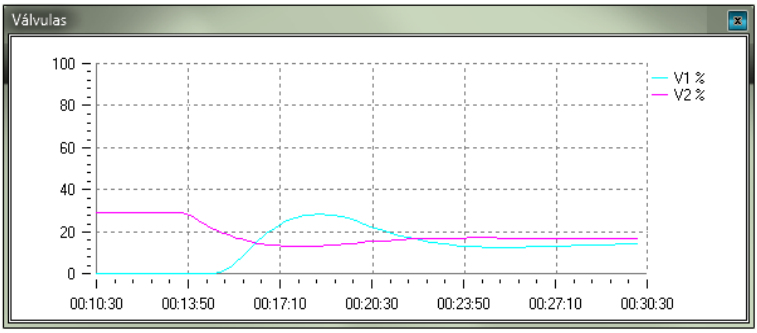


Fig. 6. Control Signals

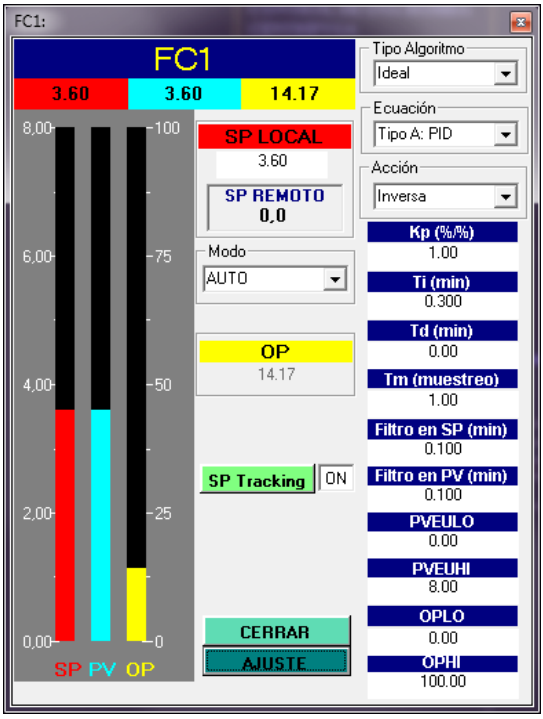


Fig. 7. PID interface

PID controllers implement the main aspects of the industrial controllers. Pressing the left mouse button placed on each PID controllers, the GUI for managing the corresponding PID is shown (Fig. 7). The bar graph lets you see (graphically and numerically) the value of the process variable (PV in blue), set point (SP in red) and output to process or control signal (OP in yellow). The user can select the controller mode (Automatic, Manual or Cascade). In AUTO mode, he can specify the SP value and, in MAN mode, the OP value. In manual mode, the user can activate the SP tracking

mechanism to avoid the well-known "bumpless" of the auto/man controller commutations. With respect to the algorithm to calculate the control signal, it is possible to use diverse algorithm and PID structures and the PID calculus use normalized SP, PV and OP values.

Pressing the setting ("AJUSTE") button of the GUI of each PID, the user accesses to the tuning parameters: proportional gain (Kp), reset time (Ti), derivative time (Td), sampling period (Tm), SP and PV time constant filters, PV and OP span values in Engineering Units (PVEULO: Process Variable Engineering Units Low, PVEUHI: Process Variable Engineering Units High, OPLO: Output to Process Low, OPHI: Output to Process high). Additionally, there are three menus to select the type of algorithm (Ideal or Interactive), the PID equation (PID, PI-D, I-PD, I) and the action controller (direct or reverse) that affects to the sign of the controller gain.

The previous experiment can be repeated with other FC1 controller parameters, for instance: Kp=5 and Ti=0.1. Then the control structure doesn't work well (Fig. 8).

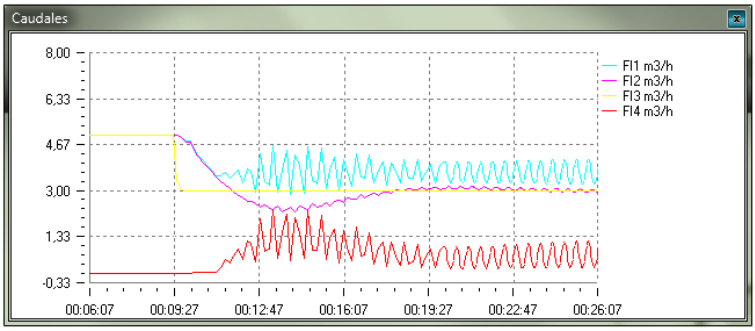


Fig. 8. Bad FC1 tuning: process and control structure response

4 Software Structure and Development Tools

When selecting a particular module a SCADA system is started. This SCADA is called EDUSCA [15] and it is the simulation module GUI. EDUSCA starts the simulation program linked to the selected module. The development of each module GUI involves the EDUSCA setting, which is done by a drag & drop strategy through a setting tool (Fig. 9).

The communication between EDUSCA and the simulation program is performed by the OPC (OLE for Process Control) communications standard for process control applications for Windows environments [16]. EDUSCA acts as an OPC client and the simulation program as an OPC server.

The simulation models have been performed using EcosimPro. EcosimPro belongs to the so called object oriented modeling languages (OOML). Many of the EcosimPro characteristics are similar to the modeling tools that implement Modelica [17]. In the sense that it supports non-causal models able to be modified automatically according to the context in which they are used. Its simulation language, called EL

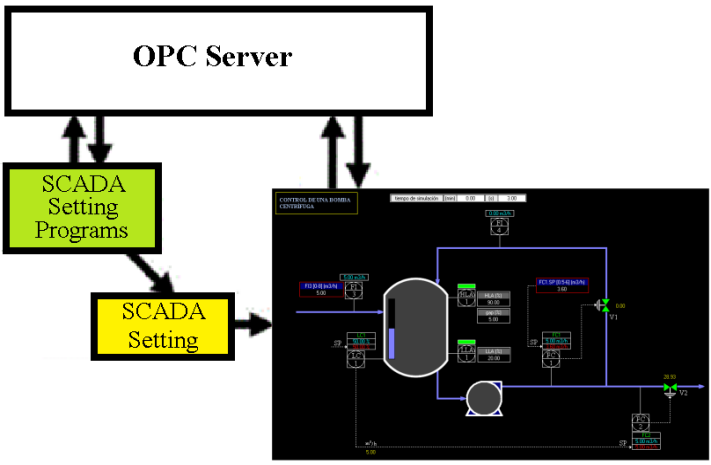


Fig. 9. GUI setting

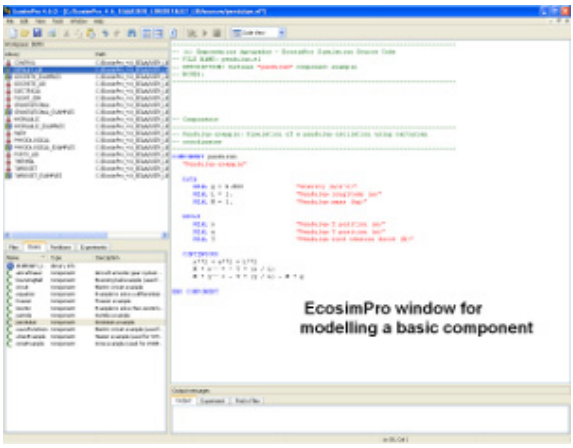


Fig. 10. Ecosimpro textual modelling view

(Ecosimpro Language), allows the description of process models, named components, in a natural way by means of continuous differential algebraic equations and discrete events variables. Each component can have a ports based interface to connect to other components. These components are grouped in libraries and an icon can be attached to each one. The user can built the system model interconnecting components by ports, using directly the modeling language (Fig. 10) or the GUI that allows the graphical modeling (Fig. 11).

Then, the resulting mathematical model is compiled and, after establishing a partition, that is describing which variables constitute the known boundary conditions and solve the problems related to the symbolic manipulation of the mathematical model (high index problems and tearing of algebraic loops), EcosimPro generates the simulation model. This simulation model is converted to C++ simulation code linked to the

numerical solvers. Finally, the user runs simulation experiments from another EcosimPro GUI view: the experimental view (Fig. 12).

The experimental view of EcosimPro allows changing the values of the boundary conditions and parameters of the model and shows the numerical value of the model variables. So, it is possible to represent graphically the value of the model variables. The main problem is that the user must know the name of the variables in order to change or show their values and it is quite difficult if the user hasn't developed the model and if the model contains hundreds or thousands of variables. So, it is the reason why it looks like convenient to dispose of a friendly interface to use the simulation model by a user different to the model builder.

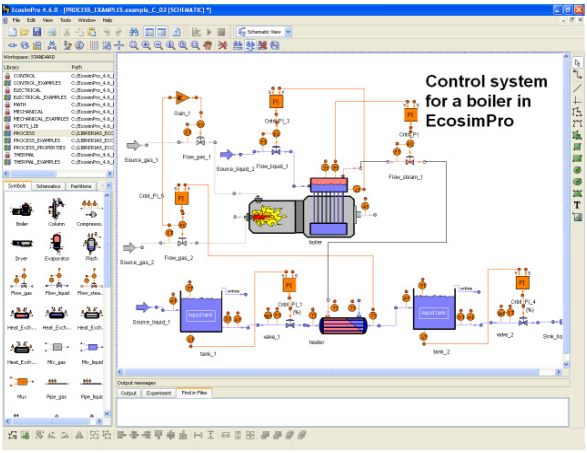


Fig. 11. Ecosimpro graphical modelling view

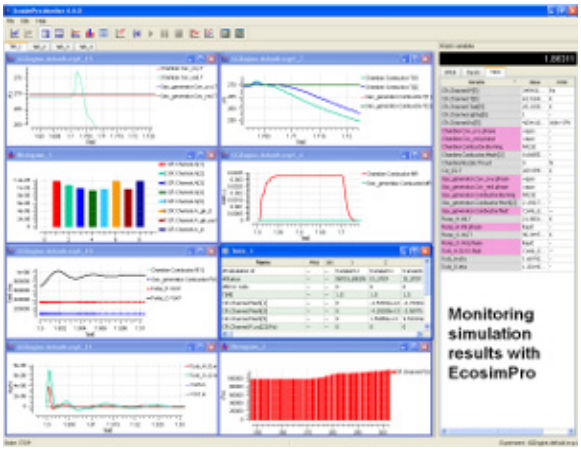


Fig. 12. Ecosimpro experimental view

In this project, two basic model libraries have been developed, one for process units and another one to design control structures. The library of mathematical models of process units is based on first principles and the degree of detail of the models is imposed for the purpose each the simulator. So, distributed or globalized parameter models can be found; fast dynamics can be explicitly modeled or simplified using static equations; empiric equations can be used to reduce the model complexity, …

In order to use the EcosimPro simulation models from a GUI different from the experimental view, EcosimPro disposes of an add-in to execute models from Excel and another module to execute models from MATLAB. But, it isn't enough to communicate the EcosimPro simulation models with our GUI (EDUSCA), because the EcosimPro simulation models don't hold OPC communications.

However an OPC server can be created by adding to the C++ simulation code the communication routines provided by the OPC standard. Then, the simulation program is converted to an OPC server can be accessed from any OPC client. This process can be automated. In our case, an application, CreaOPC [18] has been developed to set up OPC servers from the C++ sources files generated by EcosimPro (Fig. 13).

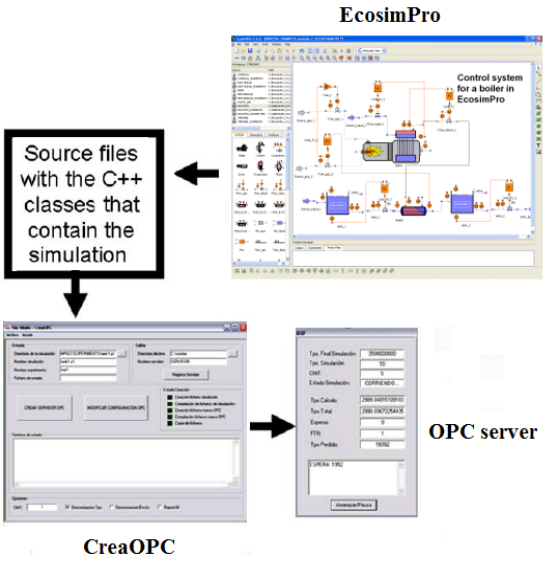


Fig. 13. OPC server simulation generation

5 Conclusions

A program with a library of simulation modules of typical control problems has been briefly exposed. This library deals with normal control problems (PID tuning; cascade, feedforward and ratio control) but, additionally, it includes other type of control problems (selective, override, split-range control) and special control strategies to guarantee security and quality process requirements. A variety of processes are considered, from the simplest ones, as tanks or heat exchangers, to the more complex ones, as boilers or distillation columns.

We consider that the program is user-friendly, few hardware and software resources are required and the functionality, the level of detail and the GUI are adapted to the industrial environment and the learning requirements for control process engineers. Moreover, it has been validated by experts with industrial skills. So, it's used successfully in the "Master in instrumentation and process control ISA-REPSOL".

Finally, to make the simulation based learning tool for control engineers a diverse set of programs and EcosimPro libraries have been developed:

- Two complete EcosimPro model libraries of process unit and control elements. They can be used to design and test different control structures to production process.
- EDUSCA: a SCADA and its setting tool. EDUSCA can be used for different purposes to the outlined in this paper. It can access to any OPC server and, consequently, it can be used to supervise laboratory plants or any OPC server simulator.
- CreaOPC, to generate OPC server simulation programs from the EcosimPro simulation models. So, the OPC server simulators can be connected to any OPC client, for instance any industrial SCADA.

Additionally, as future work, new modules can be added to the library, for instance a multivariable predictive control module. Other possible development and research line is to follow up EDUSCA to support the web based learning.

Acknowledgements. The authors want to express their gratitude to the ISE (Instituto Superior de la Energía, Fundación Respsol-YPF) and to the ISA (International Society of Automation) Spanish Section for the financial and technical support.

References

1. Industrial System 800xA simulator by ABB, http://www.abb.com
2. UniSim by Honeywell, http://hpsweb.honeywell.com
3. SimSCI-Esscor by Invensys, http://iom.invensys.com
4. TEAM_AIDES by Tecnatom, http://www.tecnatom.es
5. APROS Process Simulation Software by VTT Technical Research Centre of Finland, http://www.apros.fi
6. Acebes, L.F., Merino, A., Mazaeda, R., Alves, R., de Prada, C.: Advanced dynamic simulators to train control room operators of sugar factories. International Sugar Journal 113, 18–25 (2011)
7. LoopPro by ControlStation, http://www.controlstation.com
8. Topas by ACT, http://www.act-control.com
9. Hysys by Aspentech, http://www.aspentech.com
10. Dymola by Dynasim, http://www.dynasim.se
11. EcosimPro Dynamic modeling and simulation tool by EA International, http://www.ecosimpro.com
12. Acedo, J.: Control avanzado de procesos. Teoría y Práctica. Ediciones Díaz de Santos (2003)
13. Acedo, J.: Instrumentación y control avanzado de procesos. Teoría y Práctica. Ediciones Díaz de Santos (2006)

14. ISA-The Instrumentation, Systems, and Automation Society: Instrumentation Symbols and Identification. ISA-5.1-1984 (R1992). Formerly ANSI/ISA-5.1-1984 (R1992)
15. Alves, R., Normey-Rico, J.E., Merino, A., de Prada, C.: EDUSCA (EDUcational SCAda): Features and applications. In: Advances in Control Education, vol. 7, Part 1. Elsevier (2006)
16. Modelica Foundation, `http://www.modelica.org`
17. OPC Foundation, `http://www.opcfoundation.org`
18. Alves, R., Normey-Rico, J.E., Merino, A., Acebes, L.F., de Prada, C.: OPC based distributed real time simulation of complex continuous processes. Simulation Modelling Practice and Theory 13(7), 525–549 (2005)

Simulation-Based Development of Safety Related Interlocks

Timo Vepsäläinen and Seppo Kuikka

Tampere University of Technology, Department of Automation Science and Engineering
P.O. Box 692, FIN-33101 Tampere, Finland
{timo.vepsalainen,seppo.kuikka}@tut.fi

Abstract. Dynamic simulations could support in several ways the industrial automation and control systems development, including their interlocking functions, which constitute an important and tedious part of the development. In this paper, we present a tool-supported, automated approach for creating simulation models of controlled systems and their interlocking functions based on UML AP models of control systems and ModelicaML models of the systems to be controlled. The purpose of the approach is to facilitate manual development work related to model-based development of control systems and to enable early testing and comparison of control and interlocking strategies. The tools and the techniques are demonstrated with an example modelling project and the paper also discusses extending the approach to verifiable safety systems including their security aspects.

Keywords: Model-based development, UML AP, Simulation, Industrial control, Interlocks, Safety.

1 Introduction

Model-based development of software applications and systems has recently been the topic of numerous publications in different application domains, including software engineering and industrial control. Due to these interests, there already exist guidelines, languages and tool sets for implementing such approaches. For example, Object Management Group (OMG) has pioneered in standardization of model-based development approaches (Model-Driven Architecture, MDA) and languages for modelling (UML and profiles e.g. SysML), metamodeling (Meta Object Facility, MOF) and transforming (Query/View/Transformation, QVT) purposes. The modelling and transformation languages are already mature and supported by different tool vendors on several platforms, such as the open source Eclipse platform.

The idea of Model-Driven Architecture (MDA) and related approaches, e.g. Model-Driven Development (MDD) and Model-Driven Software Development (MDSD) is to use models (instead of documents) as primary engineering artefacts during the development. In the systems engineering domain, Model-Based Systems Engineering (MBSE) refers to applying models as part of the systems engineering process with the aim to support analysis, specification, design and verification of the systems being developed [5].

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 165–182.
DOI: 10.1007/978-3-642-34336-0_11

In model-based development processes, models are refined towards executable applications by use of model transformations but also manual development work with the models. Such processes often enable automated processing of bulk design information and are aimed at automatic code generation but can also aid analysis, understanding and documentation of the system.

In addition to analysis of models and automating error-prone development phases, another approach to improve the quality of systems and applications could be to integrate the use of simulations to model-based development. Especially, simulations could be used to facilitate the manual development work of developers by enabling, for example, comparisons of alternative design decisions. In their previous work, the authors of this paper have created and prototyped a preliminary approach to transform functional models conforming to the UML Automation Profile (UML AP, see [12],[6]) to simulation models conforming to ModelicaML [13]. The concept was presented in detail by Vepsäläinen et al. [19] and its purpose is to facilitate control system development by enabling automated creation of simulation models of controlled manufacturing systems.

In the process, the simulation models of controlled systems are composed by creating and integrating a ModelicaML simulation model of the control system to an existing ModelicaML model of the process to be controlled. The focus of the paper was in basic control functionality, e.g. feedback and cascade control structures, and the ability to support simulation of both platform independent and platform specific functions. However, according to, for example, our discussions with professionals of industrial control domain in Finland, an important and tedious part of development of control applications is related to interlocking or constraint control functions.

Interlocks could be characterized as non-safety-critical safety functions. They are often aimed to prevent deviation situations from occurring or the instrumentation from being misused, such as, to prevent pumps from running dry or to be started against closed pipelines. Interlocks do not need to be developed according to safety standards because safety is usually ensured with separate safety systems. However, because actual safety systems are often designed to ensure the safety in a simple manner, e.g. to shut down the whole processes, they should not be activated unless absolutely necessary. Another goal of interlocks can thus be seen in keeping the system in its designed operating state in order to improve the availability and productivity of it. To achieve this goal, interlocks can be more complex than actual safety functions because they do not need to meet the strict requirements of safety standards.

The development of interlocks is, however, difficult. This is because of both the complexity of the functions and because they are specific to applications and thus cannot be re-used similarly as, for example, control functions (e.g. parameterizable function blocks implementing control algorithms) can be. The actual logic, how to keep the system in its designed operating state and protect the devices, is dependent on both the controlled process and the control approach for controlling the plant or process. Another reason for the difficulty is that interlocks may originate from several sources. For example in industrial processes, part of the interlocking needs may originate from process design whereas others originate from hydraulics and electrics design. Because of the separate sources, they may have unpredictable cross-effects to the controlled system. This is another good reason for using simulations.

In this paper, we aim to extend our approach to automatically generate simulations to cover and facilitate the development of interlocking functions. We present a modelling framework supporting the modelling of the functionality of interlocks and how a simulation model of a controlled system can be created using model-based techniques. The paper also discusses the relationship between safety functions and interlocks with the purpose of assessing whether also the development of critical safety functions could be based on modelling and simulations. For defining interlocks, we do not suggest any new modelling notation. Instead, we integrate a commonly used notation to our model-based approach. The novelty of the approach is, thus, not in the way of specifying the interlocks but in the way in which simulations are integrated to model-based interlock development and how the simulation models can be created based on early design models.

This paper is organized as follows. Section 2 reviews work related to use of simulations and model-based development in industrial control and automation domain. Sections 3 and 4 present a more detailed introduction to interlocking functions, our approach to simulation-assisted development of interlocks and the developed tool support, respectively. Section 5 presents an example modelling project in which the approach and tools are utilized. Finally, before concluding the paper, section 6 discusses whether model-based, simulation assisted development techniques could be used in development of actual safety functions and to reveal security-related problems.

2 Related Work

Simulations can facilitate the development of manufacturing processes, machines and plants as well as automation and control systems in several ways. For example, Karhela in [9] mentions the use of simulations to control system testing, operator training, plant operation optimisation, process reliability and safety studies, improving processes, verifying control schemes and strategies, and start-up and shutdown analyses.

In [3] the author compares the I/O simulation approach to the traditional approach of performing system testing only on-site with the actual processes. According to the paper, the use of simulations may result in shorter start-up times as well as less waste of end products during the start-ups. In addition, simulations enable better operator training, ability to test control programs in smaller modules, and the ability to thorough testing of emergency and dangerous situations. [3]

A more recent survey on use of simulations in industrial control domain was made by Carrasco and Dormido in 2006 [2]. According to the paper, the benefits of using control systems in simulators before installation include improvements to 1) design, development and validation of the control programs and strategies, 2) design, development and validation of the HMI (human-machine Interface) and 3) adjustments of control loops and programs. [2] It is thus evident that simulations may facilitate both the development and commissioning of control systems. Simulation solutions are nowadays also provided by major control system vendors as listed in [2].

The goal of our approach is to enable automated utilization of design-time models of control systems and applications so that, for example, early simulated testing of a control or interlocking approach would not need the actual control system hardware

or tools and fully setting the system parameters. Later in development, the same techniques could enable testing and validating larger entities. Development of simulation models could be less tedious and they could be utilized also by companies performing out-sourced development phases. In our approach, we assume that a simulation model of the process to be controlled is already available. In creation of a simulation model of the controlled system including both the parts of the control system and the controlled process, we utilize model transformations that are commonly used in model-based development approaches, such as MDA of OMG.

Model-Driven Architecture (MDA) is an initiative of OMG that encourages the use of models in software development as well as re-use of solutions and best practices. MDA identifies three types of models which are Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM). [10]

In MDA, the development starts from CIM models and proceeds to PIM models and finally to PSM models which are the most detailed ones and often source models for code generation. In our approach, the focus is in PIM and PSM models with the goal of being capable of utilizing both PIM and PSM models in creation of simulation models. Thus, for example, a preliminary (early) simulation model could be created based on PIM and used for evaluating control strategies. Later, after selection of the control system vendor, the model could be refined to PSM level and simulated in conjunction with vendor specific functions in order to obtain more precise results.

In addition to our approach (see [20] and [6]), the use of model-based techniques in the automation domain has been recently proposed by several projects and papers. However, not all of these approaches identify and highlight simulation as an essential and beneficial part of development. The approach of the MEDEIA project, as discussed by Strasser et al. [15] and Ferrarini et al. [4], is based on Automation Components - composable combinations of embedded hardware and software including integrated simulation, verification and diagnostics services. In their approach, the simulation of models will be based on their interfaces, behaviour and timing specifications using IEC 61499 as a basic simulation model language [14].

Another application of model based techniques to development of industrial control applications has been presented by Tranoris and Thramboulidis [16]. In their approach, the design and deployment of applications is addressed by means of the function block (FB) construct of IEC 61499. Model transformations are used to create function block models. In the paper, they dont address simulations but similarly to the MEDEIA approach, FB models could possibly be used with simulations of the process to be controlled.

In both the approach of MEDEIA and that of Tranoris and Thramboulidis, simulations could be supported with the implementation technology (IEC 61499) of produced applications. The essential difference to our approach is that we aim to support simulation with a simulation language so that, for example, basic simulation functions of simulation tools could be fully exploited. These functions are listed in [2] and include saving and loading current and initial states, freeze, run and replay simulation, working in slow and fast mode and support for malfunction situations.

Furthermore, we identify simulation as a beneficial and important activity also in case of model-based development. We claim that also model-based development

requires manual work and genuine design decisions made by developers because it may not be possible to express all the relevant aspects in models and all the relevant knowledge about decision making in model transformations. To facilitate the manual design work, we foresee that simulation techniques could provide a feasible solution and that model-based techniques could facilitate the creation of the required simulation models.

Similarities between interlocks of basic control system and safety functions of safety systems are remarkable. The main difference is that actual safety functions need to be developed according to safety standards, such as IEC 61508 [7], which may require a sophisticated development process, use of techniques recommended by the standards and a detailed documentation about the system and the development activities used. In their recommendations, standards are always conservative which may be one reason why the use of model-based techniques in safety system development has been unusual in the past. However, according to the present (second) edition of IEC 61508, automatic software generation could aid the completeness and correctness of architecture design as well as freedom from intrinsic design faults. Hence, the use of model-based techniques in development of also safety-critical applications may be increasing in near future. The question of how to develop safety-critical systems with model-based techniques is thus both important and current but not addressed by many researchers, so far.

However, Biehl et al. [1] have attempted to integrate safety analysis to model-based software development in automotive industry in order to automate performing of safety-analysis on refined models with minimal effort. In [21] the authors have extracted the key safety-related concepts of RTCA DO-178B standard into a UML profile in order to use them to facilitate the communication between different stakeholders in software development.

3 Simulation of Interlocking Designs

The focus of this paper is in interlocking (or constraint control) functions of basic control systems, which are an important and challenging part of control system development. Interlocks are control functions, the purpose of which is to either guarantee the safety of the process or to keep the system in its designed operating state and protect the devices and actuators from being misused by the control system. Quite often, safety is achieved with a separate safety system so that the purpose of the interlocks is the latter one.

Interlockings are typically designed during the basic design phase of the control systems [20]. The amount of program code, related to interlockings is often smaller than that of code related to basic control functionality. However, their development is still time-consuming and prone to errors because interlocks cannot be reused similarly as, for example, controllers can be. This is due to the fact that the actual interlocking needs, logics and delays are always specific to the application. Solutions to re-occurring needs in controlled processes can be librarized but even they need careful examinations and potential modifications before re-use.

For industrial systems, interlocks are often specified with vendor neutral logic diagrams - or vendor specific logic and function block diagrams if the control system

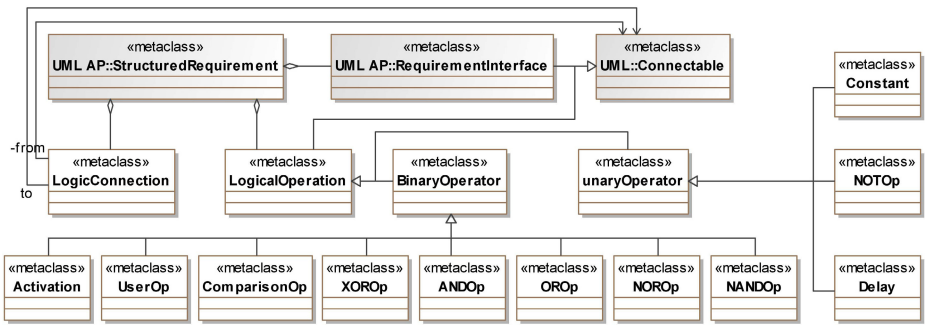


Fig. 1. The essential additions to UML AP metamodel to support the definition of interlocks

vendor has been selected. In the process, the diagrams are used for depicting the activating and disabling conditions of the functions, and possibly overriding control values for locked actuators or devices. Logic diagrams suit well to this purpose because they are familiar to developers and unambiguous. Logic diagrams, as a semi-formal method, are also highly recommended by IEC 61508 to detailed design of safety-critical software [7]. Logic diagram based approach for defining the interlocks is thus both sound and already familiar to developers of the domain.

The purpose of UML AP is to cover both the specification of requirements and functionality of automation and control applications. Logic diagrams may aid in supporting both of these features but used from separate points of view. Especially, in the development of safety-related applications, requirements must be defined clearly and in an unambiguous manner. On the other hand, formal or semi-formal specification of functionality is a necessity in enabling simulation of design or in automating generation of code. In our approach, the logic diagram concepts were added to be used with both the Requirements Modelling sub-profile and functional Automation Concepts sub-profile of UML AP and the UML AP tool (see [17]). The concepts and some related existing modelling concepts of the profile are presented in figure 1. Existing UML AP and UML metamodel elements are highlighted with grey colour.

In UML AP, requirements are structured concepts that can be connected to other requirements with port-like requirement interfaces in order to model dependencies between required functions. The purpose of the logical operations and logic connections, on the other hand, is to enable the modelling of required activations of interlocks and algorithms to compute control values inside requirements. Required interchange of computed signals and values can then be modelled with the requirement interfaces that extend the same UML::Connectable concept than logical operations and can be thus connected together with logic connections. The operations include familiar operations, such as AND and OR, but also delay, constant, Activation gate (that lets its input flow to output when control input is activated), comparison operator and a UserOperation with which the developer can specify the logic to output from inputs with a textual equation. Examples of use of part of the concepts will be provided in section 5.

The functional modelling concepts of UML AP, Automation Functions, constitute a hierarchy of function-block-like concepts. The hierarchy is based on their purpose,

such as to execute control algorithms, compute interlocking signals or to interface with sensors or actuators of the system. The hierarchy including its justification is presented in detail in [6]. Automation Functions (AFs) exchange signals between them with ports that extend the UML::Connectable concept (see figure 1). The logic operators and connections, on the other hand, can be used inside the AFs to define the functionality of them. In the profile and tool implementation this was enabled by adding an aggregation association from the Automation Function base metaclass to logical operation and logic connection metaclasses similarly to the structure presented in figure 1. Consequently, the technical challenges of our approach to simulate the models are in transforming the specifications conforming to UML AP to simulation models. The solution to transform the models to ModelicaML models and finally to simulateable Modelica models will be discussed in next section.

4 Technical Background and Implementation of the Approach

It is first necessary to present some basic information about Modelica and ModelicaML that are used in our approach as target simulation languages. Modelica is an object oriented simulation language for large, complex and heterogeneous physical systems and a registered trademark of Modelica Association. Modelica models are mathematically described by differential, algebraic and discrete equations. Modelica includes also a graphical notation and user models are usually described by schematics that are also called object diagrams. A schematic consists of components, such as pumps, motors and resistors, which are connected together using connectors (ports) and connections. A component, on the other hand, can be defined by another schematic or, on the lowest level, as a textual equation based definition.

Modelica Modeling Language (ModelicaML) has been created to enable an efficient way to create, read, understand and maintain Modelica models with UML tools [13]. ModelicaML is a UML profile and defines stereotypes and tagged values of stereotypes that correspond to the keywords and concepts of the textual Modelica language. For example, a Modelica block with a set of equations can be modelled by creating a UML class, applying a <<block>> stereotype to it and defining the equations to the equations tagged value of to the stereotype.

ModelicaML models are not simulateable as they are (at least with current tool support) but can be transformed to simulateable Modelica models. Tool support for generating textual Modelica models, as well as the profile, is made publicly available by the OpenModelica project. [11] The profile is based on UML2 implementation of the UML metamodel on the Eclipse platform. UML2 is itself based on Eclipse Modeling Framework (EMF) which is an implementation of OMG Meta Object Facility (MOF) specification.

EMF is also utilized in our UML AP metamodel implementation that is the basis of the UML AP Tool [17]. UML AP and ModelicaML models are thus instances of metamodels defined with EMF implementation of MOF and because of this similar background, the shifting between UML AP and ModelicaML can be realized with use of standardized QVT languages. QVT languages are intended for defining model transformations between models conforming to MOF based metamodels and they are also

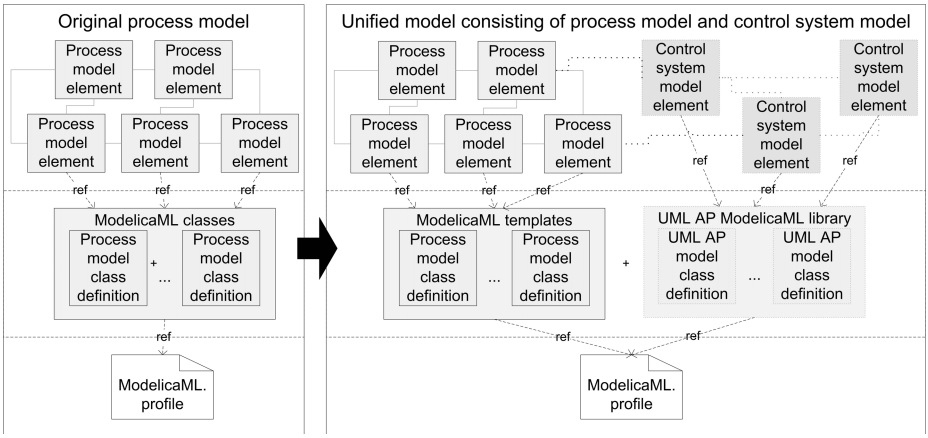


Fig. 2. The purpose of the transformation is to add the control system specific parts to an existing model of the physical process

specified by OMG. The possibility to use standardized transformation languages with existing open source tool support and the open source background of Modelica and ModelicaML are good reasons for selecting Modelica as the target simulation language in our approach.

In Modelica (and in ModelicaML) simulation classes are defined separately from their use context, similarly to classes in object oriented programming languages. In ModelicaML models, the model elements also need to reference the ModelicaML profile in order to use the stereotypes and tagged values of it. This results in a structure sketched in the left side of figure 2. ModelicaML models consist of Modelica class definitions and instances of the classes. Classes may contain ports with which they can be connected and both the definitions and instances of the classes need to use the stereotypes of the ModelicaML profile in order to map the concepts to Modelica keywords. In our approach, we assume that ModelicaML models of processes to be controlled are available and conform to this structure.

The purpose of the transformation is to create and add the control system specific parts to the existing model of the process to be controlled and to connect the created parts to the existing model so that the controlled system can be simulated. In this process, Modelica class definitions corresponding to platform independent (PIM) and platform specific (PSM) UML AP elements are copied to the model of the process to be controlled (process model) so that they can be referenced from the process model. Instances of the templates are instantiated to the process model and connected together according to the control system model in the UML AP tool. Instances corresponding to measurement and actuation AutomationFunctions are connected to the elements of the process model that are used to model sensors and actuators.In more detail, this process and the characteristics of different kind of AFs are discussed in detail in [19].

However, because interlocks are specific to applications they cannot be librarized, as explained earlier. Instead, the definitions of interlock classes need to be created by the transformation based on the logic diagrams. This process is rather simple and illustrated

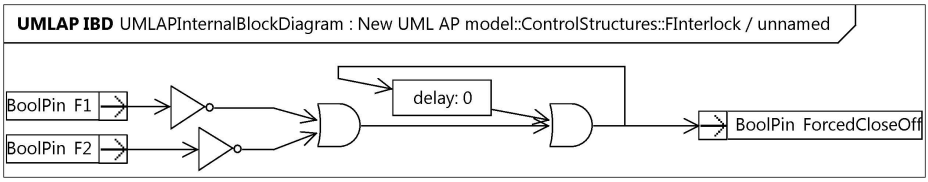


Fig. 3. Simple example of an interlocking function

with an example shown in figure 3. Ports contained by classes, such as interlocks, are special kind of classes in Modelica and finally typed by type definitions in ModelicaML profile. When creating ModelicaML classes based on UML AP classes, instances of such special port classes can be created based on ports used in the UML AP model so that their naming will be maintained and the suitable type chosen based on the type of the port. This applies to both input and output ports.

Figure 3 contains only three kinds of logical operations of the 11 presented in figure 1: two NOT and two OR operations and one delay. The transformation processes logical operations by creating a property (variable) for each operation instance. In case of Boolean operations (NOT, AND, NAND, OR, NOR, XOR), the type of the property is always Boolean. In case of other operations, the type needs to be defined in the UML AP model so that the corresponding ModelicaML type can be chosen by the transformation. The equations determining the values of the properties are created based on the kind of the operation (for example NOT or AND) and the connections coming into the operation which can be followed to another operation or port, for which there will also be a property (variable) with the same name. In case of the example interlock presented in figure 3, the value of the first OR operation (from left in the figure) can be defined to be equal to the logical OR of the values of the NOT operations and the second OR operation to equal to the logical OR of the first OR operation and the delay operation.

The transformation, thus, tries to define the values of properties with equations. However, if a model contains loops, this may not be possible. For example, figure 3 contains a loop the purpose of which is to keep the interlock activated if it once activates so that the output of the second OR operation (from left) is true. Certain kinds of loops may produce errors due to discontinuation of the variables, at least with the OpenModelica tool that we use for simulating. The problem was solved by using algorithms in which operations are applied in an order (instead of equations that apply all the time). This is also one of the interactive features of our transformation. If the transformation detects a loop within an interlock or other kind of AF, it creates algorithmic statements instead of equations based on the model, shows the statements to the user of the tool and lets the user arrange the order in which they will be executed in the model.

Another interactive feature of the transformation is related to connecting parts of the simulation model created based on the UML AP control system model to the existing parts of the process to be controlled. These connections are necessary for, for example, connecting measurement and actuating functions of control systems to sensors and actuators of the process models, respectively. By default, the transformation uses

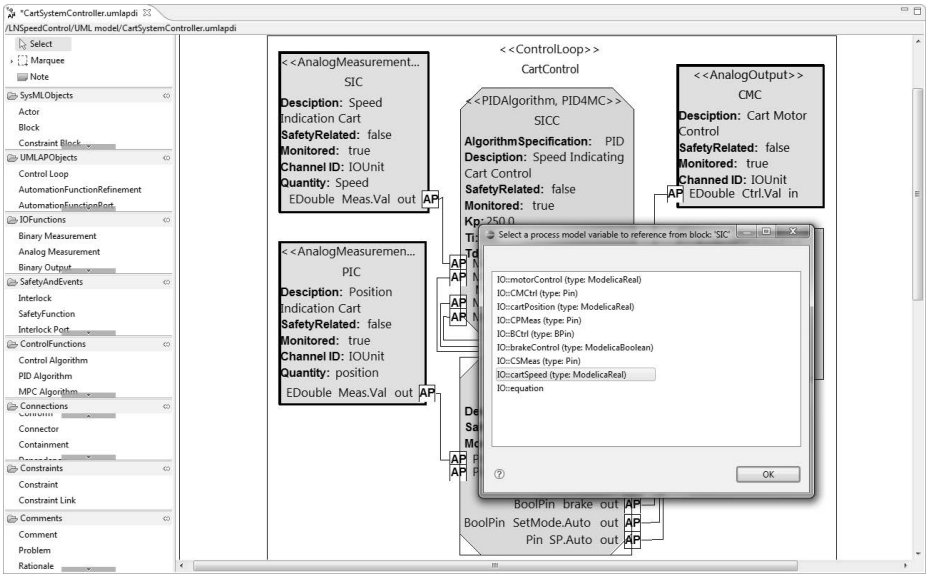


Fig. 4. The graphical user interface of the UML AP Tool for selecting the correct property to reference from the model of the process to be controlled

properties of the process model with specific names or the names of properties that have been specified with a specific <<VariableMapping>> stereotype. However, if suitable properties cannot be found by the transformation, it provides the user of the tool with a list of properties available in the model class in question and lets the user choose the correct property as in figure 4.

The third interactive feature is related to non-connected input ports. When an unconnected input port is detected by the transformation, the user of the tool is asked for a constant value for the port. In this case, the user of the tool may define a constant value for it in order to be able to simulate the model or, if the port has been left unconnected unintentionally, leave the port unconnected and fix the problem before executing the transformation again.

The transformation definition was written with QVT operational mappings language and it specifies how to process a target ModelicaML model based on a source UML AP model. Executable Java-transformation code implementing the definition and to be used in the Eclipse environment was generated with SmartQVT tooling. In order to be able to launch and control the transformation from the UML AP Tool, the Java class was packaged to a plugin defining an extension to one of the extension points of the tool. The structure of the plugin was similar to the plugin structure presented in [18]; the referred paper also presents in detail the extension points of the tool.

In the plugin structure, the generated transformation class is extended with an own (hand-written) transformation class that can be used to implement also required black box operations. In QVT vocabulary, black boxes are operations written with common programming languages (in this case Java) in order to implement operations that are

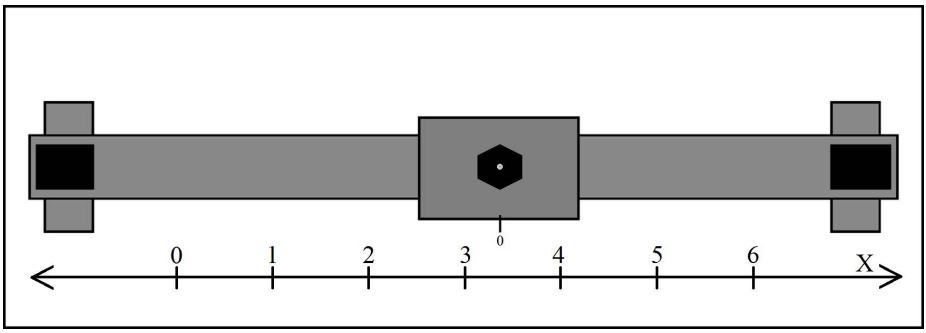


Fig. 5. Simple example system to be controlled includes a cart that can be moved along a rail

hard to express with QVT concepts. In this case, for example handling of stereotypes and tagged values related to them as well as interactive features of the transformation were implemented as black boxes.

5 Example System and Utilization of the Tools

The purpose of this section is to provide a simple example in which the modelling concepts and tools are used in creation of a simulation model of a controlled process to evaluate two alternative interlocking approaches.

An illustration of the (partial) system to be controlled is shown in figure 5. The system consists of a cart and a rail along which the cart can be moved with an electric motor. The cart can be stopped with a brake, if necessary. The purpose of the cart is left unspecified and not illustrated in the figure. It could be assumed, for example, to operate a boom or a gripping device nearby the rail. The control needs to be addressed in the example are related to controlling and interlocking the velocity and location of the cart. The operator of the system controls the system by giving speed requests (setpoints) with a joystick that is connected to the control system. The control system, on the other hand, is required to control the velocity of the cart with feedback control and to protect the cart from colliding to stoppers at the end of the rail. The location of the cart must be kept between 0.0 and 6.0.

In industrial installations the need for a similar stopping interlocks could be caused by, for example, forbidden areas in factories and sites or needs to restrict operation ranges of booms and devices of mobile platforms to ensure their stability in varying terrains. Consequently, despite the simplicity of the example, it can be considered as a generalization of a common functionality in industrial systems.

There are several ways in which the functionality could be implemented; however, in this paper we will sketch and simulate only two simple versions of them. Firstly, the control system could observe the location and direction of the cart and stop it with the brake, if the cart violates the limits. For example, when reaching coordinate 0.0, the control system could activate the brake and keep it activated until the speed request would be towards back to the allowed area. Secondly, the control system could

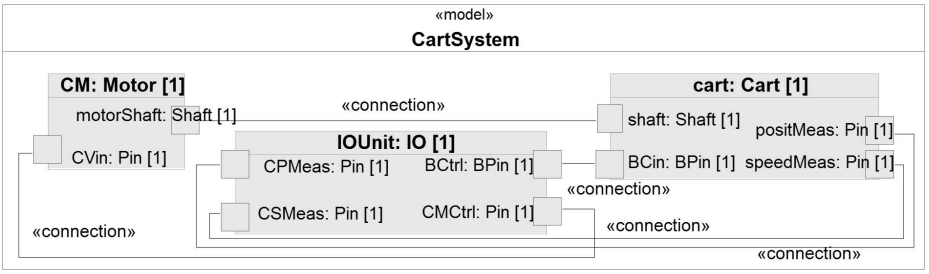


Fig. 6. Model of the system to be controlled as a ModelicaML model, composite structure diagram

be designed to constrain the speed setpoint near the limits so that the setpoint would be zero at the limit coordinates and it would be reduced already before reaching the limits. These approaches will be simulated next based on a ModelicaML model of the process to be controlled and UML AP models of the two control approaches.

To be able to utilize the tools and techniques presented in this paper, the system to be controlled need to be available as a ModelicaML model. The UML composite diagram presenting the simplified model of the system is presented in figure 6. The model was specified with open source Papyrus UML tool, with OpenModelica extensions, and it consists of 3 ModelicaML components that are instances of ModelicaML classes. The cart is operated with a motor (CM) that takes its control signal from the IOUnit that collects all measurement and control signals between the process and control system. The total weight of the cart and motor is assumed to be 20kg (m_{total}) and the radius of the drive wheel 0.1m (r_{dw}). The torque (T) and acceleration (a) equations of the motor and cart based on drive voltage (V_d) are presented in equations 1, 2 and 3. The numerical values of the constants of the motor are: $R_m = 0.5$, $L_m = 0.0015$, $K_{emf} = 0.05$ and $K_t = 0.01$. The brake is assumed to be able to decelerate the cart with force of 200N (Fb). The equations are, thus, simple but sufficient for demonstration purposes.

$$V_d - \omega * K_{emf} = L_m * dI/dt + R_m * I \tag{1}$$

$$T = K_t * I \tag{2}$$

$$T/r_{dw} + F_b = m_{total} * a \tag{3}$$

The UML AP control structure diagram presenting the similar parts of the two control solutions for the system is depicted in figure 7. The control solution consists of analogue measurements of cart position and speed, an interlock, a PID controller and an analogue and a binary output for controlling the motor and the brake, respectively. The two interlocking approaches discussed earlier influence mainly the contents of the interlock. In the first approach, the speed request (setpoint) is not constrained. However, in order to enable that to be implemented in the second approach, the speed request is relayed through the interlock AF.

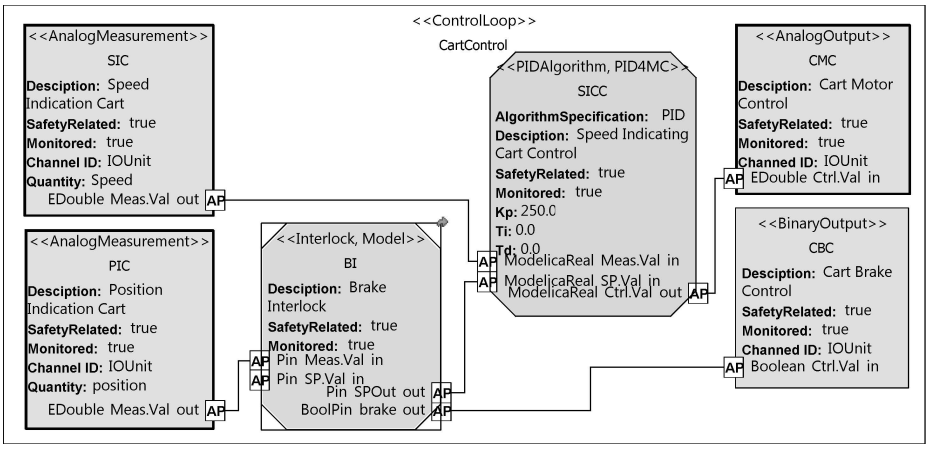


Fig. 7. UML AP control structure diagram of a control solution for controlling the process

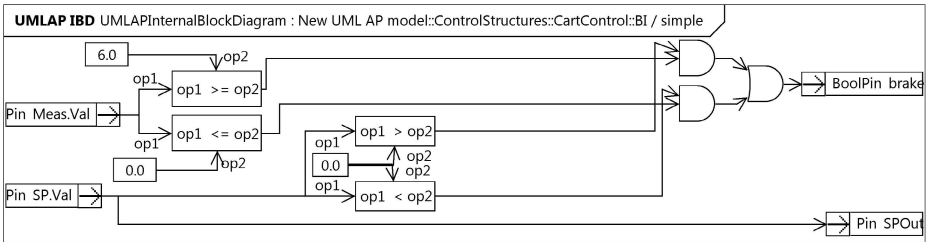


Fig. 8. An illustration of the first interlocking solution

The detailed logic of the first interlocking solution is presented in figure 8. The solution is designed to activate the brake outside the intended working area if the speed request is driving the cart away from the working area. In order to be able to revert back to the working area, the brake is not activated if the speed request is towards the allowed working area. In order to work well, the system should probably also wait for the cart to stop before deactivating the brake; however, because of simplicity this feature was not modelled.

After specification of the detailed control solution, the transformation, discussed in section 4, was used to transform the UML AP control solution to ModelicaML and to append it to the existing model of the physical process (see figure 6). In order to simulate the model, the ModelicaML model was further transformed to Modelica code with OpenModelica tooling. The shifting from UML AP model of the control solution to simulateable model of the system including both the process and the control system was, thus, automated with two model transformations.

The simulation result related to the first interlocking approach is presented in figure 10a. At the beginning, both the position and velocity of the cart are 0. The speed request (from the operator) is ramped from 0 to 1 and kept at 1 for 7 seconds in order to drive

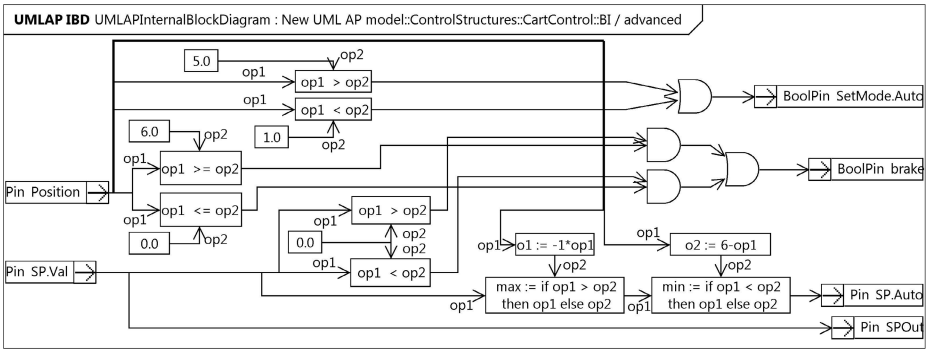


Fig. 9. The second interlocking solution

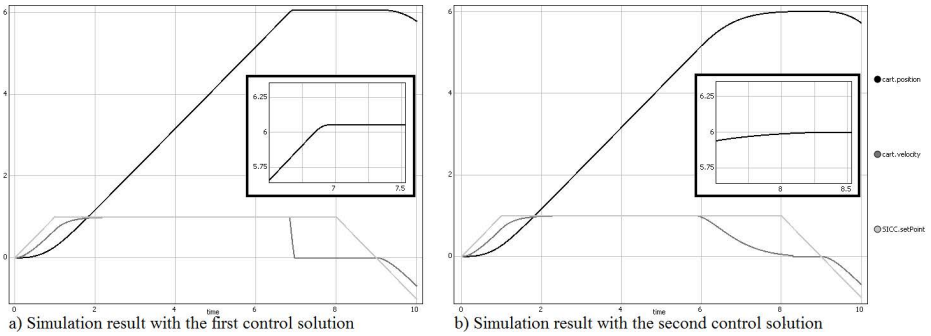


Fig. 10. Simulation results of the first (a) and second (b) control solutions plotting cart position, velocity and speed request from the operator (SICC.setPoint)

the cart to the forbidden area in the positive end of the rail. After this, the speed request is ramped to -1 in order to revert the cart away from the border of the forbidden area. According to the simulation, the control solution works as it was intended; however, because it takes time to stop the cart, the location of the cart reaches approximately 6.05 before stopping. Clearly, the control solution could be improved by decelerating the cart already before reaching the limits. In case of strict limits, violating the forbidden area (overshoot) could also be avoided by activating the brake before the strict limits; however, this would reduce the size of the actual, allowed working area.

The second control solution is illustrated in figure 9. In this solution, the braking is implemented similarly to the first solution and the speed request from the user is relayed similarly to the controller. However, when the measured location of the cart is between 0 and 1 or between 5 and 6, an automatic mode is activated and another speed setpoint is calculated by the interlock function. The setpoint is constrained so that between 0 and 1 and between 5 and 6, the maximum allowable speed setpoint towards the forbidden area is equal to the distance left to the forbidden area. For example, if the location of the cart is 5.5, the maximum allowed speed setpoint to the positive direction is 0.5. In order to relay the second, calculated setpoint signal and the mode activation signal, the interlock

AF has been added two new ports. Similar ports were added also to the controller block (see figure 7) and its equations.

The simulation result related to the second, improved interlocking AF is presented in figure 10b. The speed request obtained from the operator is similar to that of the first simulation. In this case, the cart is smoothly decelerated already before reaching the limit and the overshoot is much smaller than that in the first simulation. Clearly, this alternative provides a better control performance as even in case of strict safety limits, the cart would not need to be stopped with the brake before the actual limit.

6 Towards Development of Certifiable Safety Functions

The use of model-based techniques in development of safety-critical applications has not been recommended by safety standards, such as IEC 61508, until recently. However, due to the new version of the standard, they could be used, for example, during architecture design to aid the completeness and correctness of design as well as freedom from intrinsic design faults.

Perhaps the most essential difference between the development of safety systems and basic control systems is that safety systems require extensive documentation about all the development activities and their results for verification, validation and certification purposes. The development of safety systems should also be risk-driven so that the requirements and design artefacts could be always traced to the perceived safety needs which originate from risk and hazard analysis. When assessing the quality of design, design artefacts could be compared to the original safety needs and when in doubt, developers could always refer to the risks and hazards.

We are currently striving to extend the scope of UML AP to cover also the development and design of safety systems. The work is targeted to the requirement concepts of the profile (see [6]) but also to documenting the results of risk and hazard analysis. Including the risk and hazard information in a compact form in the same models with the design would not only enable the use of explicit traces between them but also aid the discovery of the information when the developers need it. Moreover, compared to use of separate documents, a unified (one) model could be easier to maintain and keep up-to-date if and when something needs to be changed in design.

An admitted difficulty in development of both safety critical and basic control systems is related to the specification of requirements. In development of safety-critical applications, the functional requirements (what the system must do) originate from hazard analysis and the non-functional requirements (how well it must be done) from risk analysis. However, unambiguous and complete specification of the functional requirements is still difficult. Perhaps this task could be facilitated with a semi-formal, domain specific modelling approach. Another working direction is the ability to simulate designs and specifications.

In [8] the author has analysed the quality of produced software in about 12500 projects from year 1984 to 2008 and the defects delivered (and removed) during the projects. The results may not be directly generalizable to safety-critical applications as such. However, according to the survey, also in the best-in-class-quality, a significant portion of defects delivered were related to defects in requirements specifications, partly because defects in requirements are difficult to discover.

If the design could be simulated earlier, for example with the techniques presented in this paper, simulations could also be used to assess whether the required functionality is able to detect and handle the hazardous situations. The feedback loop between design and requirements could thus be shortened. This could further facilitate the development of both basic control and safety-systems.

Testing or simulation-aided testing of design and development specifications cannot be used to prove the correctness of them. However, simulations can be used to test the reactions of control or safety systems to events in the system that could not be tested with the actual system without compromising safety. Moreover, simulations may aid in comparing alternative solutions in terms of, for example, availability of the controlled system that may be important from the point of view of the developer organization or the end user but not that of safety standards. Extensive testing is also required by safety standards. The problem with conventional testing is that the system should be already implemented in order to be tested. If both the simulation model and the executable application would be produced by trusted, automatic transformations based on the same model, testing could as well utilize the simulation model of the application. Consequently, with our approach, an improvement would also be the ability to test earlier in the development process based on platform independent models.

Another issue that must be increasingly considered in industrial control systems in future is security, as demonstrated by the recent Stuxnet worm. It can be easily justified that compromising security may lead to compromising safety, for example if a safety-critical, measured value is lost or modified. However, security is hardly mentioned in safety standards such as IEC 61508. We are expecting a change in this in near future. Security issues could also be taken into account in simulations. For example, security-related test simulations could be supported by making it possible to mark vulnerable information channels for the simulation engine. The engine could then add a constant or time-varying gain for the values transferred with use of the channel to test the reactions of the control system to an unusual situation. Losing a connection totally would also be a meaningful simulation case that could reveal serious vulnerabilities in systems.

7 Conclusions

This paper has presented a tool-supported approach to transform functional UML AP models and their interlocking specifications to ModelicaML models and finally to simulateable Modelica models. The aim of the approach and transformation implementation is to enable automated and less tedious creation of simulation models and thus to support model-driven development of control systems, including their interlocking and constraint control functions. Compared to present development practices of control systems, this could enable the testing of the solutions earlier during the development process. The approach also offers the other, listed benefits of simulations.

The example system and the control approaches presented in this paper were simple but still adequate for demonstrating the techniques in creation of two simulation models. Simulations could be used to compare the two designed interlocking approaches within a feedback control system. This is also how simulations are currently typically used if their development is considered worthwhile.

Simulations can facilitate the analysis – not directly the synthesis – of control systems. Nevertheless, simulations can help developers in making judicious design decisions. The purpose of model-based techniques is often to automate simple development tasks. However, also within model-based development, real design decisions need to be made by developers and this work can be eased with simulations. In our approach, we use model-based techniques also for developing the simulation models. We thus aim to enhance model-based development of control systems by widening the scope of model-based techniques.

A future working direction of our approach is to shift towards safety functions which share several similarities with interlocks. It is clear that also development of safety functions could benefit from simulations; possibly also from the security point of view. However, the development of safety related systems requires extensive documentation of design and traceability between design artefacts. This is why we are currently working with the requirement sub-profile of UML AP. With this work, we not only support the detailed definition of requirements but also documentation of information originating from risk and hazard analysis. The rationale is that the requirements of safety functions are based on these analyses but the information is not always visible for, for example, the software developers, which makes it difficult to judge the correctness and completeness of design.

References

1. Biehl, M., DeJiu, C., Törngren, M.: Integrating safety analysis into the model-based development toolchain of automotive embedded systems. In: LCTES 2010, pp. 125–132. ACM, New York (2010)
2. Carrasco, J., Dormido, S.: Analysis of the use of industrial control systems in simulators: State of the art and basic guidelines. ISA Transactions 45(2), 295–312 (2006)
3. Dougall, J.: Applications and benefits of real-time I/0 simulation for PLC and PC control systems. ISA Transactions 36(4), 305–311 (1998)
4. Ferrarini, L., Dede, A., Salaun, P., Dang, T., Fogliazza, G.: Domain specific views in model-driven embedded systems design in industrial automation. In: INDIN 2009 the 7th IEEE International Conference on Industrial Informatics, Cardiff, UK, June 23-26 (2009)
5. Friedenthal, S., Moore, A., Steiner, R.: A practical guide to SysML. Morgan Kaufmann OMG Press, San Francisco (2008)
6. Hästbacka, D., Vepsäläinen, T., Kuikka, S.: Model-driven Development of Industrial Process Control Applications. The Journal of Systems and Software 84(7), 1100–1113 (2011), doi:10.1016/j.jss.2011.01.063
7. IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems. parts 1-7 (2010)
8. Jones, C.: Software quality in 2008: A survey of the state of the art. Software Productivity Research LLC, 59 p. (2008), http://www.jasst.jp/archives/jasst08e/pdf/A1.pdf (achieved February 13, 2011)
9. Karhela, T.: A software architecture for configuration and usage of process simulation models: Software component technology and XML-based approach. PhD Thesis, VTT Technical Research Centre, Finland (2002)
10. Object Management Group. Technical Guide to Model Driven Architecture: The MDA Guide. Version 1.0.1 (2003)

11. OpenModelica project website (2011), http://www.ida.liu.se/pelab/modelica/OpenModelica.html
12. Ritala, T., Kuikka, S.: UML Automation Profile: Enhancing the Efficiency of Software Development in the Automation Industry. In: The Proceedings of the 5th IEEE International Conference on Industrial Informatics (INDIN 2007), Vienna, Austria, July 23-27, pp. 885–890 (2007)
13. Schamai, W.: Modelica Modeling Language (ModelicaML) a UML Profile for Modelica, Technical Report 2009:5, EADS IW, Germany, Linköping University, Institute of Technology
14. Strasser, T., Rooker, M., Ebenhofer, G.: MEDEIA - Model-Driven Embedded Systems Design Environment for the Industrial Automation Sector. 1st Version of the MEDEIA open source modelling prototype, documentation (2009), http://www.medeia.eu/26.0.html
15. Strasser, T., Rooker, M., Hegny, I., Wenger, M., Zoitl, A., Ferrarini, L., Dede, A., Colla, M.: A research roadmap for model-driven design of embedded systems for automation components. In: INDIN 2009 the 7th IEEE International Conference on Industrial Informatics, Cardiff, UK, June 23-26 (2009)
16. Tranoris, C., Thramboulidis, C.: A tool supported engineering process for developing control applications. Computers in Industry 57, 462–472 (2006)
17. Vepsäläinen, T., Hästbacka, D., Kuikka, S.: Tool Support for the UML Automation Profile - for Domain-Specific Software Development in Manufacturing. In: The Proceedings of the 3rd International Conference on Software Engineering Advances, Sliema, Malta, October 26-31, pp. 43–50 (2008)
18. Vepsäläinen, T., Hästbacka, D., Kuikka, S.: A Model-driven Tool Environment for Automation and Control Application Development - Transformation Assisted, Extendable Approach. In: Proceedings of the 7th Nordic Workshop on Model Driven Software Engineering, Tampere, Finland, August 26-28 (2009)
19. Vepsäläinen, T., Hästbacka, D., Kuikka, S.: Simulation Assisted Model-Based Control Development - Unifying UML AP and Modelica ML. In: 11th International Middle Eastern Simulation Multi Conference, Alexandria, Egypt, December 1-3 (2010)
20. Vepsäläinen, T., Sierla, S., Peltola, J., Kuikka, S.: Assessing the Industrial Applicability and Adoption Potential of the AUKOTON Model Driven Control Application Engineering Approach. In: Proceedings of International Conference on Industrial Informatics, Osaka, Japan, July 13-16 (2010)
21. Zoughbi, G., Briand, L., Labiche, Y.: A UML Profile for Developing Airworthiness-Compliant (RTCA DO-178B), Safety-Critical Software. In: Engels, G., Opdyke, B., Schmidt, D.C., Weil, F. (eds.) MODELS 2007. LNCS, vol. 4735, pp. 574–588. Springer, Heidelberg (2007)

Modelling Molecular Processes by Individual-Based Simulations Applied to Actin Polymerisation

Stefan Pauleweit[1], J. Barbara Nebe[2], and Olaf Wolkenhauer[1]

[1] Institute of Computer Science, Dept. of Systems Biology & Bioinformatics, University of Rostock, 18051 Rostock, Germany
[2] Center for Biomedical Research, Dept. of Cell Biology, University of Rostock 18051 Rostock, Germany
sp173@informatik.uni-rostock.de,
barbara.nebe@med.uni-rostock.de
http://www.sbi.uni-rostock.de

Abstract. Used in ecology, economics and social science, agent-based modelling is also increasingly used in the life science. We use this technique to model and simulate the processing of actin filaments. These filaments form a major part of the cell-shape determining cytoskeleton and contribute to a number of cell functions. In our paper, we develop and investigate three models with different levels of detail. Our work demonstrates the potential of individual-based modelling in systems biology.

1 Introduction

Agent-based simulations are a promising application emerging in life sciences [15]. Applications of agent-based technologies in systems biology include studies in which each cell is modelled as an agent [28]. Examples include bacterial chemotaxis [6], the phenomenon where cells direct their movements in response to external signals, models of epidermal tissue [10], the formation of a 3D skin epithelium [26] or a hybrid model, and combination of agent-based simulations and differential equations to analyse the cell response to epidermal growth factors [30]. Moreover, agent-based models for intracellular interactions representing the carbohydrate oxidation cell metabolism [4], the cell cycle [27], the NF-κB signalling pathway [21] and molecular self-organisation, with the focus on packing rigid molecules [29], have been proposed.

Actin polymerisation is a molecular process that generates long filaments with a barbed and a pointed end from single actin molecules that become part of the cytoskeleton. The cytoskeleton provides the physical structure and shape of cells, as well as plays an important role in a number of cell functions, including cell motility [3,19], endocytosis [8], or cell division [20]. Understanding of actin organisation has important implications for practical medical applications, including the development of new topographies for implant surfaces [14,17].

Here we focus on the spatial and time dependent simulation of actin polymerisation. The literature describes a number of models analysing the cell motility driven by actin filaments, using partial differential equations [16]. Another study used Brownian

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 183–191.
DOI: 10.1007/978-3-642-34336-0_12

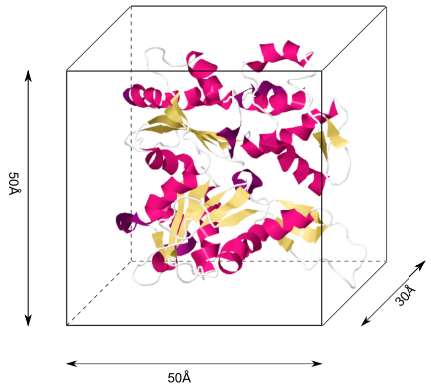


Fig. 1. The physical size of the actin molecule determines the size of an agent in the simulation. For the two dimensional simulation the width and height of an actin agent is set to 50Å×50Å [18].

dynamics to analyse the self-assembly process of actin and the dynamics of long filaments [11]. The distribution of the length of actin filaments inside a cell was analysed with a discrete and continuous model [5]. Different models using stochastic π-calculus as a representative of process algebra, have also been published [2].

In this paper we describe simulations using an agent-based approach with communicating X-machines [9], implemented in a software called Flexible Large-scale Agent-based Modelling Environment (FLAME) [13]. This allows us to analyse the spatial and time dependent behaviour during the composition of the filament structure by free actin with a high degree of physical realism. The outline of the paper is as follows. Section 2 explains the three models in detail. Section 3 discusses the output of the models and Section 4 sums up the conclusions and gives a brief outlook for further studies.

2 Agent-Based Simulation

An agent is formally defined as an finite-state machine. Because the finite-state machine model is too restrictive for general system specification, an extension with a memory, the so called X-machine promise a better implementation [12]. If a system contains more than one agent, the particular X-machines must be able to communicate together and this leads to a communication X-machine system [9]. This concept is implemented in the software named Flexible Large-scale Agent-based Modelling Environment (FLAME) [13].

Using the actin model generated by X-ray analysis [18] we fix the size of one molecule to 50Å×50Å. The dimension of the molecule is shown in Figure 1. Each agent contains an identification number and two binding sides to connect to another agent, namely bottom-bound (BB) and top-bound (TB) and can switch between three different states (free, bottom-bound, fully bound). A free binding side is denoted with the constant -1. As long as both binding sides are marked with -1 (free), the agent is randomly rotating and moving around in a distance of 1–200Å, which is an approximation for the computational expensive calculation of Brownian dynamics. If a

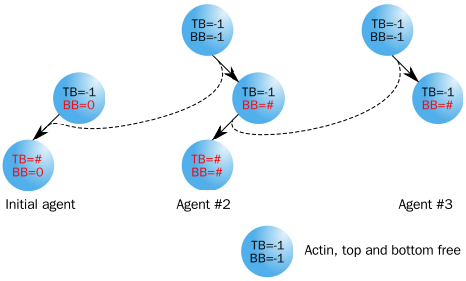


Fig. 2. An actin-agent can be in three different states. A free molecule can bind to an already bound initial agent (BB $= 0$). The already bound actin-agent then become fully bound. Then the third actin-agent can bind to the second actin-agent which become fully bound and so on.

molecule binds to another, then the identification number of the counterpart is stored in the BB (respectively TB) variable; the agent becomes immobilised and its rotation will be adapted. The precondition for binding is, that one of the agents is already bound (bottom-bound). This leads to the condition that at least one agent has to be stuck in the beginning of the simulation. This is done by initialising one agent with BB $= 0$. If a free agent binds to an already bound one, the second becomes then fully bound (BB $\neq -1$, TB $\neq -1$). The whole schema of the actin–actin interactions is also shown in Figure 2.

The polymerisation of actin filaments is characterised by a 70° angle branching on several positions mediated by the Arp2/3 protein [25]. To simulate this branching process, a new agent with a third binding side was implemented. The orientation of the branching side to the left or right was set randomly. This agent is restricted to bind only to actin-agents, so that a Arp2/3-Arp2/3 combination is prohibited. In Figure 3 the scheme for the interactions and state changes is illustrated. Similar to the actin-agent, the size of this agent was determined from published measurements [24]. Figure 4 shows the approximated dimensions of Arp2/3.

An agent-based model has to include the reaction kinetic in a reasonable way. Due to the nature of spatial simulations with individual molecules, this may be done by an interaction volume, which defines a reaction zone around a particular agent (see Figure 5). Andrews and Bray (2004) developed an algorithm to determine this volume, but considered more detailed interactions. Another way is described by Pogson et al. (2006) where the interaction radius r is calculated by:

$$r = \sqrt[3]{\frac{3\mathrm{k}\Delta t}{4\pi \mathrm{N_A} 10^3}}$$

where k is the kinetic rate constant, Δt the discrete time interval and $\mathrm{N_A}$ is Avogadro's constant (6.022×10^{23}). The rate constant for actin-actin assembly was determined with $11.6 \mu M^{-1} s^{-1}$ [7] and leads to a radius of 0.166Å for $\Delta t = 1s$. If two or more agents enter the interaction volume at the same time step, the closest molecule to the reaction molecule assembles to it, if two or more have the same distance, one will be chosen by chance.

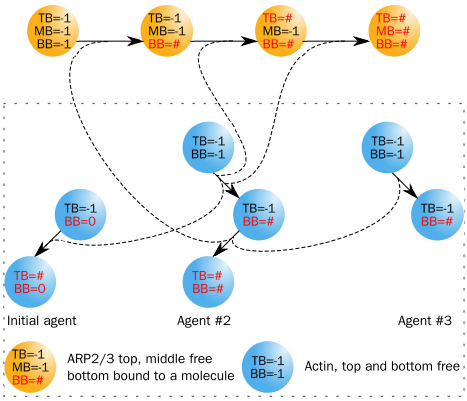


Fig. 3. In addition to the actin-agent (Figure 2), the simulation was extended with a second type of agent for Arp2/3. This agent can bind to an bottom-bound actin-agent. Then another actin-agent can bind to the top-binding side of the Arp2/3-agent, the next actin-agent to the middle binding side and the Arp2/3-agent becomes fully bound.

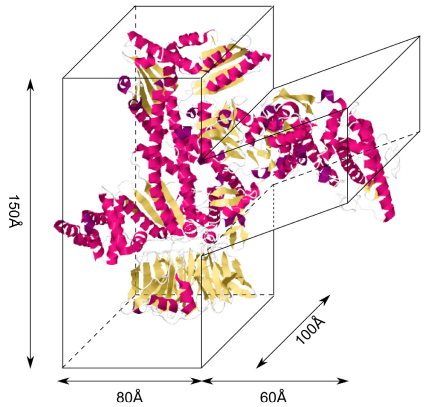


Fig. 4. The physical size of Arp2/3 determines the size of the agents in the simulation [24]

To compare our results with the simulation of Cardelli et al. (2009), we used the same number of 1200 free actin agents and 30 Arp2/3 agents. Cardelli uses this number of agents to simulate a concentration of 1200 μM. For concentration values in a spatial simulations, it is necessary to calculate the volume of the environment:

$$
\begin{aligned}
n_{\mathrm{Actin}} &= \mathrm{N_A} \times V \times c \quad [1/mol \times l \times mol/l] \\
V &= 1200/(\mathrm{N_A} \cdot 1200 \times 10^{-6}) \\
V &= 1.66 \times 10^{-18}\, l = 1.66 \times 10^{-21} m^3 \quad ,
\end{aligned}
$$

where n_{Actin} is the number of molecules, $\mathrm{N_A}$ is again Avogadro's constant, c is the concentration of molecules and V is the volume. Assuming the environment as a cube, the length of a side is approximately 1184.0Å.

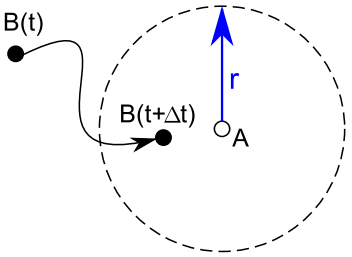


Fig. 5. The interaction boundary (dashed circle) defines the reaction volume around an agent. If a second agent enters this area, the reaction takes place

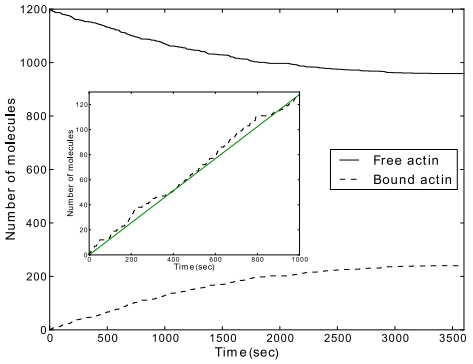


Fig. 6. The time plot shows the result of the simulation for a simple actin polymerisation with 1200 agents and a time step $\Delta t = 1s$. Inset: Linear slope of the binding process in the beginning

For a simulation including the dissolving of actin from a filament, we used the rate constant of 5.4 s^{-1} for ADP-actin at the barbed end from the literature [7]. Only agents with a free top-bound (in case of Arp2/3 also middle bound) can be released from the filament. To avoid an instant re-coupling, the molecule will be moved to outside the interaction boundary.

Our present agent-based simulation takes place in a 2D environment, so that we have to introduce a factoring constant of 100 for the radius, the dissolving rate constant and the size of the environment, following the paper of Cardelli et al. (2009).

3 Results

3.1 Actin-Actin Interactions

Figure 6 shows the time plot of the growth of one filament. The curve shows in the beginning a linear increase (see inset of Figure 6), but later becomes logarithmic. After 390 seconds 50 agents were integrated in the filament, which corresponds to a filament of length $0.25\mu m$. A length of $1\mu m$ is reached after 1882 seconds and at the end of one hour, 240 agents form a filament with a length of $1.2\mu m$. In agreement with published measurements [7], the increase in length of actin is linear in the beginning of the

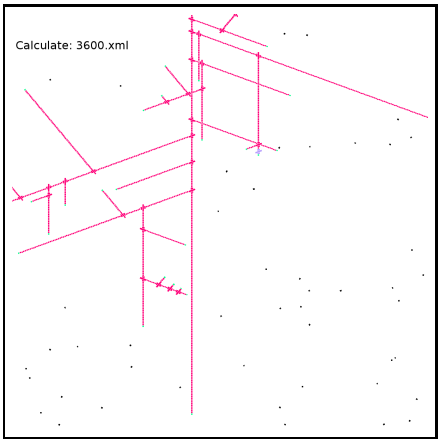


Fig. 7. The figure (cropped for better illustration) shows the end result of the simulation for one hour with 1200 actin-agents and 30 Arp2/3-agents. The black points mimic the free actin, the blue the agents bind to the filamental structure.

simulation. The logarithmic curve on can be explained by the decreased number of free molecules and the spatial phenomena, by which the simulated filament is growing close to the boundary of the environment. The number of reachable free molecules close to this boundary is then much lower. The difference in the speed of elongation is related to two reasons:

1. Actin filaments can growth on both side, whereas the simulation allows only the growth at the barbed end.
2. Actin can build small motile fragments, which then elongate the filament [25]. This increases the speed of polymerisation significantly.

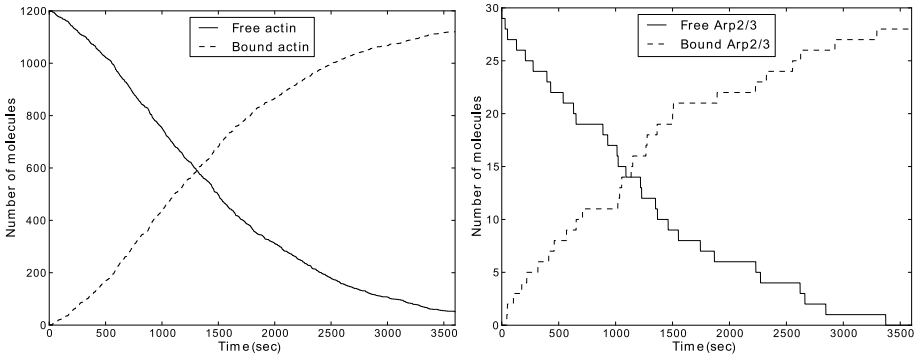
3.2 Branching Process

Figures 8(a) and 8(b) show the time plots for the actin and arp agents respectively. Both time curves are sigmoidal with an inflection point around 1300 seconds.

Adding a new agent for the Arp2/3 protein, we simulated the actin polymerisation with the branching process. To visualise this, Figure 7 shows a snapshot of the spatial distribution at the end of one hour. In this simulation an overall filament length of $1\mu m$ was reached after 578 seconds. At the end of one hour, nearly all agents were involved in the filament structure, 1121 actin agents were fully bound. Additionally 28 Arp2/3 agents are fully bound, two of them had an open binding side. In contrast to the filament formation, solely with actin, the branching process accelerate the elongation significantly. The snapshot in Figure 7 shows the spatial consideration and is in good agreement with previously published simulations [2, Figure 17].

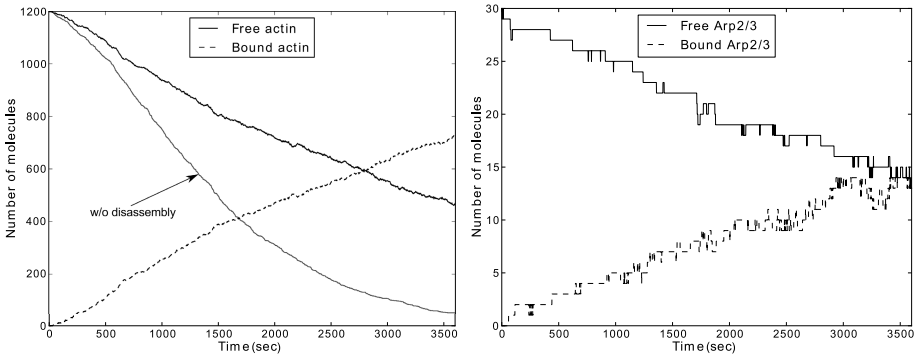
3.3 Disassembly Process

To model the disassembly of actin and Arp2/3 molecules from the filamental structure, we added a new probability for each agent.



(a) Time plot of actin-agents. (b) Time plot of arp-agents.

Fig. 8. The time plots show the result of the branching simulation for 1200 actin-agents, 30 Arp2/3-agents and a time step $\Delta t = 1s$



(a) Time plot of actin-agents. (b) Time plot of arp-agents.

Fig. 9. The time plots show the result of the branching simulation, including the disassembly process, for 1200 actin-agents, 30 Arp2/3-agents and a time step $\Delta t = 1s$

After introducing this new variable, the assembly of the actin filament slowed down. As shown in Figure 9(a), the assembly of 200 molecules and therefore an overall length of $1\mu m$ is reached after 760 seconds. After one hour, the filament contained 717 fully bounded actin-agents and is branched out at 14 different positions (see also Figure 9(b)). This model shows therefore a comparable time progression to the simulation of Cardelli et al. (2009), especially for Arp2/3, although our filamental growth is somewhat slower.

4 Conclusions and Outlook

Instead of the commonly used rate equations to simulate intracellular molecular processes, we introduced an agent-based approach. This allowed us to overcome some restrictions imposed by differential equation models, more precisely any number and

any distribution, as well as spatial behaviour of molecules can be easily modelled. Our model simulates actin polymerisation, an important key player for different cell functions.

The spatial outcome of our model is comparable to alternative models of Cardelli et al. (2009), using the stochastic π-calculus. Because the FLAME-framework produces XML-files for each time step, we are also able to create an animated version for tracking the filament formation (not shown here). Additionally a time dependent analysis of the behaviour of the single molecules and the filaments can be done. The limit in using the agent-based approach is only given by computational purposes.

Our overall aim is the development of a biophysical realistic model for actin polymerisation in human cells. The advantage of our approach is the possibility to to extend the simulation to a massive number of molecules with the aid of the parallelised FLAME software version and, more important, the easy implementation of external influences. This should enable us to analyse observed phenomena of actin clustering on titan pillar surface structures [14] with applications to implant technologies. This interesting issue makes it necessary to include more proteins like capping proteins which stop the elongation of the filament [23].

Acknowledgements. We are grateful for financial support of the research training school "Welisa", which is founded by the German Research Foundation (DFG 1505/1). Furthermore the authors are thankful for the helpful advice of Prof. Mike Holcombe and Mark Burkitt from the University of Sheffield.

References

1. Andrews, S.S., Bray, D.: Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. Phys. Biol. 1(3-4), 137–151 (2004)
2. Cardelli, L., Caron, E., Gardner, P., Kahramanoğullari, O., Phillips, A.: A process model of actin polymerisation. Electronic Notes in Theoretical Computer Science 229(1), 127–144 (2009); Proceedings of the Second Workshop From Biology to Concurrency and Back (FBTC 2008)
3. Cooper, J.A.: The role of actin polymerization in cell motility. Annu. Rev. Physiol. 53, 585–605 (1991)
4. Corradini, F., Merelli, E., Vita, M.: A Multi-agent System for Modelling Carbohydrate Oxidation in Cell. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3481, pp. 1264–1273. Springer, Heidelberg (2005)
5. Edelstein-Keshet, L., Ermentrout, G.B.: Models for the length distributions of actin filaments: I. simple polymerization and fragmentation. Bull. Math. Biol. 60(3), 449–475 (1998)
6. Emonet, T., Macal, C.M., North, M.J., Wickersham, C.E., Cluzel, P.: Agentcell: a digital single-cell assay for bacterial chemotaxis. Bioinformatics 21(11), 2714–2721 (2005)
7. Fujiwara, I., Vavylonis, D., Pollard, T.D.: Polymerization kinetics of adp- and adp-pi-actin determined by fluorescence microscopy. Proc. Natl. Acad. Sci. U. S. A. 104(21), 8827–8832 (2007)
8. Galletta, B.J., Mooren, O.L., Cooper, J.A.: Actin dynamics and endocytosis in yeast and mammals. Curr. Opin. Biotechnol. Curr. Opin. Biotechnol. (July 2010) (in press)
9. Gheorghe, M., Stamatopoulou, I., Holcombe, M., Kefalas, P.: Modelling Dynamically Organised Colonies of Bio-entities. In: Banâtre, J.-P., Fradet, P., Giavitto, J.-L., Michel, O. (eds.) UPP 2004. LNCS, vol. 3566, pp. 207–224. Springer, Heidelberg (2005)

10. Grabe, N., Neuber, K.: A multicellular systems biology model predicts epidermal morphology, kinetics and ca2+ flow. Bioinformatics 21(17), 3541–3547 (2005)
11. Guo, K., Shillcock, J., Lipowsky, R.: Self-assembly of actin monomers into long filaments: Brownian dynamics simulations. J. Chem. Phys. 131(1), 015102 (2009)
12. Holcombe, M.: X-machines as a basis for dynamic system specification. Softw. Eng. J. 3(2), 69–76 (1988), http://portal.acm.org/citation.cfm?id=48328
13. Kiran, M., Coakley, S., Walkinshaw, N., McMinn, P., Holcombe, M.: Validation and discovery from computational biology models. Biosystems 93(1-2), 141–150 (2008)
14. Matschegewski, C., Staehlke, S., Loeffler, R., Lange, R., Chai, F., Kern, D.P., Beck, U., Nebe, B.J.: Cell architecture-cell function dependencies on titanium arrays with regular geometry. Biomaterials 31(22), 5729–5740 (2010)
15. Merelli, E., Armano, G., Cannata, N., Corradini, F., d'Inverno, M., Doms, A., Lord, P., Martin, A., Milanesi, L., Müller, S., Schroeder, M., Luck, M.: Agents in bioinformatics, computational and systems biology. Brief. Bioinform. 8(1), 45–59 (2007)
16. Mogilner, A., Edelstein-Keshet, L.: Regulation of actin dynamics in rapidly moving cells: a quantitative analysis. Biophys. J. 83(3), 1237–1258 (2002)
17. Nebe, J.G.B., Luethen, F., Lange, R., Beck, U.: Interface interactions of osteoblasts with structured titanium and the correlation between physicochemical characteristics and cell biological parameters. Macromol. Biosci. 7(5), 567–578 (2007)
18. Oda, T., Iwasa, M., Aihara, T., Maéda, Y., Narita, A.: The nature of the globular- to fibrous-actin transition. Nature 457(7228), 441–445 (2009)
19. Pantaloni, D., Clainche, C.L., Carlier, M.F.: Mechanism of actin-based motility. Science 292(5521), 1502–1506 (2001)
20. Pelham, R.J., Chang, F.: Actin dynamics in the contractile ring during cytokinesis in fission yeast. Nature 419(6902), 82–86 (2002)
21. Pogson, M., Holcombe, M., Smallwood, R., Qwarnstrom, E.: Introducing spatial information into predictive nf-kappab modelling–an agent-based approach. PLoS One 3(6), e2367 (2008)
22. Pogson, M., Smallwood, R., Qwarnstrom, E., Holcombe, M.: Formal agent-based modelling of intracellular chemical interactions. Biosystems 85(1), 37–45 (2006)
23. Pollard, T.D., Cooper, J.A.: Actin and actin-binding proteins. A critical evaluation of mechanisms and functions. Annu. Rev. Biochem. 55, 987–1035 (1986)
24. Robinson, R.C., Turbedsky, K., Kaiser, D.A., Marchand, J.B., Higgs, H.N., Choe, S., Pollard, T.D.: Crystal structure of arp2/3 complex. Science 294(5547), 1679–1684 (2001)
25. Stossel, T.P., Fenteany, G., Hartwig, J.H.: Cell surface actin remodeling. J. Cell Sci. 119(Pt. 16), 3261–3264 (2006)
26. Sun, T., Adra, S., Smallwood, R., Holcombe, M., MacNeil, S.: Exploring hypotheses of the actions of tgf-beta1 in epidermal wound healing using a 3d computational multiscale model of the human epidermis. PLoS One 4(12), e8515 (2009)
27. Sütterlin, T., Huber, S., Dickhaus, H., Grabe, N.: Modeling multi-cellular behavior in epidermal tissue homeostasis via finite state machines in multi-agent systems. Bioinformatics 25(16), 2057–2063 (2009)
28. Thorne, B.C., Bailey, A.M., Peirce, S.M.: Combining experiments with multi-cell agent-based modeling to study biological tissue patterning. Brief. Bioinform. 8(4), 245–257 (2007)
29. Troisi, A., Wong, V., Ratner, M.A.: An agent-based approach for modeling molecular self-organization. Proc. Natl. Acad. Sci. USA 102(2), 255–260 (2005)
30. Walker, D., Wood, S., Southgate, J., Holcombe, M., Smallwood, R.: An integrated agent-mathematical model of the effect of intercellular signalling via the epidermal growth factor receptor on cell proliferation. J. Theor. Biol. 242(3), 774–789 (2006)

Marine Ecosystem Model Calibration through Enhanced Surrogate-Based Optimization

Malte Prieß[1], Slawomir Koziel[2], and Thomas Slawig[1]

[1] Institute for Computer Science, Cluster The Future Ocean
Christian-Albrechts Universität zu Kiel, 24098 Kiel, Germany
[2] Engineering Optimization & Modeling Center, School of Science and Engineering
Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland
{mpr,ts}@informatik.uni-kiel.de, koziel@ru.is

Abstract. Mathematical optimization of models based on simulations usually requires a substantial number of computationally expensive model evaluations and it is therefore often impractical. An improved surrogate-based optimization methodology, which addresses these issues, is developed for the optimization of a representative of the class of one-dimensional marine ecosystem models. Our technique is based upon a multiplicative response correction technique to create a computationally cheap but yet reasonably accurate surrogate from a temporarily coarser discretized physics-based coarse model. The original version of this methodology was capable of yielding about 84% computational cost savings when compared to the fine ecosystem model optimization. Here, we demonstrate that by employing relatively simple modifications, the surrogate model accuracy and the efficiency of the optimization process can be further improved. More specifically, for the considered test case, the optimization cost is reduced three times, i.e., from about 15% to only 5% of the cost of the direct fine model optimization.

Keywords: Marine Ecosystem Models, Surrogate-based Optimization, Parameter Optimization, Response Correction, Data Assimilation.

1 Introduction

Numerical simulations nowadays play an important role to simulate the earth's climate system and to forecast its future behavior. The processes to be modeled and simulated are ranging from fluid mechanics (in atmosphere and oceans) to bio- and biochemical interactions, e.g., in marine or other type of ecosystems. The underlying models are typically given as time-dependent partial differential or differential algebraic equations [7,10,12].

Among them, marine ecosystem models describe photosynthesis and other biogeochemical processes in the marine ecosystem that are important, e.g., to compute and predict the oceanic uptake of carbon dioxide (CO_2) as part of the global carbon cycle [17]. They are typically coupled to ocean circulation models. Since many important processes are non-linear, the numerical effort to simulate the whole or parts of such a coupled system with a satisfying accuracy and resolution is quite high.

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 193–208.
DOI: 10.1007/978-3-642-34336-0_13

There are processes in the climate system where where even without much simplification (through e.g. "parametrizations" to reduce the system size, see for example [12]) several quantities or parameters are unknown or very difficult to measure. This is for example the case for growth and dying rates in marine ecosystem models [5,17], one of which our work in this paper is based on. Before a transient simulation of a model (e.g., used for predictions) is possible, the latter has to be calibrated, i.e., relevant parameters have to be identified using measurement data (sometimes also known as data assimilation). For this purpose, large-scale optimization methods become crucial for a climate system forecast.

The aim of parameter optimization is to adjust or identify the model parameters such that the model response fits given measurement data. The mathematical task thus can be classified as a least-squares type optimization or inverse problem [2,3,21]. This optimization (or calibration) process requires a substantial number of function and optionally sensitivity or even Hessian matrix evaluations. Evaluation times for the high-fidelity model of several hours, days or even weeks are not uncommon. As a consequence, optimization and control problems are often still beyond the capability of modern numerical algorithms and computer power. For such problems, where the optimization of coupled marine ecosystem models is a representative example, development of faster methods that would reduce the number of expensive simulations necessary to yield a satisfactory solution becomes critical.

Computationally efficient optimization of expensive simulation models (*high-fidelity* or fine models) can be realized using surrogate-based optimization (SBO), see for example [1,6,9,15]. The idea of SBO is to exploit a surrogate, a computationally cheap and yet reasonably accurate representation of the high-fidelity model. The surrogate replaces the original high-fidelity model in the optimization process in the sense of providing predictions of the model optimum. Also, it is updated using the high-fidelity model data accumulated during the process. The prediction-updating scheme is normally iterated in order to refine the search and to locate the high-fidelity model optimum as precisely as possible. One of possible ways of creating the surrogate, our work in this paper is based on, is to utilize a physics-based *low-fidelity* (or coarse) model. The development and use of low-fidelity models obtained by, e.g., coarser discretizations (in time and/or space) or by parametrizations is common in climate research [12], whereas their applications for surrogate-based parameter optimization in this area is new.

In [14], a surrogate-based methodology has been developed for the optimization of climate model parameters. As a case study, a selected representative of the class of one-dimensional marine ecosystem models was considered. Since biochemistry mainly happens locally in space and since the complexity of the biogeochemical processes included in this specific model is high, this model serves as a good test example for the applicability of surrogate-based optimization approaches. The technique described in [14] is based on a multiplicative response correction of a temporally coarser discretized physics-based low-fidelity model. It has been successfully applied and demonstrated to yield substantial computational cost savings of the optimization process when compared to a direct optimization of the high-fidelity model.

In this paper, we demonstrate that by employing simple modifications of the original response correction scheme, one can improve the surrogate's accuracy, as well as further

reduce the computational cost of the optimization process. We verify our approach by using synthetic target data and by comparing the results of SBO with the improved surrogate to those obtained with the original one. The optimization cost is reduced three times when compared to previous results, i.e., from about 15% to only 5% of the cost of the direct high-fidelity ecosystem model optimization (used as a benchmark method). The corresponding time savings are increased to from 84% to 95%.

It should be emphasized that the proposed approach does not rely on high-fidelity model sensitivity data. As a consequence, the first-order consistency condition between the surrogate and the high-fidelity model (i.e., agreement of their derivatives) is not fully satisfied. Nevertheless, the combination of the knowledge about the marine system under consideration embedded in the low-fidelity model and the response correction is sufficient to obtain a quality solution in terms of good model calibration, i.e., its match with the target output.

The paper is organized as follows. The high-fidelity ecosystem model, considered here as a test problem, as well as the low-fidelity counterpart that we use as a basis to construct the surrogate model, are described in Section 2. The optimization problem under consideration is formulated in Section 3. The original and improved response correction schemes and the comparison of the corresponding surrogate model qualities are discussed in Section 4. Numerical results for an illustrative SBO run are provided in Section 5. Section 6 concludes the paper.

2 Model Description

The considered example for the class of one-dimensional marine ecosystem models simulates the interaction of dissolved inorganic nitrogen, phytoplankton, zooplankton and detritus (dead material), thus is of so-called $NPZD$ type [13]. The model uses pre-computed ocean circulation and temperature data from an ocean model (in a sometimes called *off-line mode*), i.e., no feedback by the biogeochemistry on the circulation and temperature is modeled, see again [13]. The original high-fidelity (fine) model and its low-fidelity (coarse) counterpart which we use as a basis to construct a surrogate for further use in the optimization process are briefly described below.

2.1 The High-Fidelity Model

The $NPZD$ model simulates one water column at a given horizontal position. This is motivated by the fact that there have been special time series studies at fixed locations. Clearly, the computational effort in a one-dimensional simulation is significantly smaller than in the three-dimensional case. However, as pointed out in the introduction, the model – from point of view of the complexity of the included processes – serves as a good test example for the applicability of SBO approaches.

In the $NPZD$ model, the concentrations (in mmol N m^{-3}) of dissolved inorganic nitrogen N, phytoplankton P, zooplankton Z, and detritus (i.e., dead material) D are summarized in the vector $y = (y^{(l)})_{l=N,P,Z,D}$ and described by the following coupled PDE system

$$\begin{aligned}\frac{\partial y^{(l)}}{\partial t} &= \frac{\partial}{\partial z}\left(\kappa\frac{\partial y^{(l)}}{\partial z}\right) + Q^{(l)}(y, u_2, \ldots, u_n), \quad l = N, P, Z,\\ \frac{\partial y^{(D)}}{\partial t} &= \frac{\partial}{\partial z}\left(\kappa\frac{\partial y^{(D)}}{\partial z}\right) + Q^{(D)}(y, u_2, \ldots, u_n) - \frac{\partial y^{(D)}}{\partial z}u_1, \quad l = D,\end{aligned} \tag{1}$$

in $(-H, 0) \times (0, T)$, with additional appropriate initial values. Here, z denotes the only remaining, vertical spatial coordinate, and H the depth of the water column. The terms $Q^{(l)}$ are the biogeochemical coupling (or *source-minus-sink*) terms for the four tracers and $\mathbf{u} = (u_1, \ldots, u_n)$ is the vector of unknown physical and biological parameters, with $n = 12$ for this specific model. The sinking term (with the sinking velocity u_1) is only apparent in the equation for detritus. In the one-dimensional model no advection term is used, since a reduction to vertical advection would make no sense. Thus, the circulation data (taken from an ocean model) are the turbulent mixing coefficient $\kappa = \kappa(z, t)$ and the temperature $\Theta = \Theta(z, t)$, which goes into the nonlinear coupling terms $Q^{(l)}$ but is omitted in the notation.

The parameters $\mathbf{u}$ to be optimized are, for example, growth and dying rates of the tracers and thus appear in the nonlinear coupling terms $Q^{(l)}_{l=N,P,Z,D}$ in (1). For the sake of brevity and for the purpose of this paper we omit the explicit formulation of the coupling terms as well as the explicit physical meaning of the involved parameter. For details we refer the reader to [13, 16].

2.2 Numerical Solution

The continuous model (1) is discretized and solved using an operator splitting method [11], an explicit Euler time stepping scheme for the nonlinear coupling terms Q and the sinking term while using an implicit scheme for the diffusion term. For further details we refer the reader to [13, 14].

More explicitly, in every discrete time step, at first the nonlinear coupling operators Q_j (that depend on t_j directly and/or via the temperature field Θ) are computed at every spatial grid point and integrated by four explicit Euler steps with step size $\tau/4$. Then, an explicit Euler step with full step size τ is performed for the sinking term. Finally, an implicit Euler step for the diffusion operator, again with full step size τ, is applied.

In the original model, the time step τ is chosen as one hour. By choosing this time step, all relevant processes are captured and further decrease of the time step does not improve the accuracy of the model. The model with this particular time step will be referred to as the high-fidelity or fine one in the following.

We furthermore denote by $\mathbf{y}_j \approx y(\cdot, t_j)$ the discrete fine model solution of the continuos model (1) in time step j (containing all tracers N, P, Z, D) given as

$$\mathbf{y}_j = (y_{ji})_{i=1,\ldots,I}, \quad j = 1, \ldots, M_f, \quad \mathbf{y} \in \mathbb{R}^{M_f I}, \quad I = n_z n_t, \tag{2}$$

where I denotes the number of spatial discrete points n_z times the number of tracers n_t, which is four for the considered model, and where M_f denotes the total number of discrete time steps, given the discrete time step τ_f. More specifically, the model consists of $n_z = 66$ vertical layers and is integrated over totally $M_f = 8760$ time steps/year $\times$

5 years $= 43800$ discrete time step. We will furthermore use the subscript f to distinguish the relevant fine model variables, which read $\mathbf{y}_f, \tau_f$ and M_f, from those we will introduce for the coarse model, respectively.

2.3 The Low-Fidelity Model

Marine ecosystem model, are typically given as coupled time-dependent partial differential equations, compare [5,17]. One straightforward way to introduce a low-fidelity (or coarse) model for these models is to reduce the spatial and/or temporal resolution, whereas, in this paper, we exploit the latter one.

The coarse model, which is a less accurate but computationally cheap representation of $\mathbf{y}_f$ is obtained by using a coarser time discretization with a discrete time step τ_c given as

$$\tau_c = \beta \tau_f, \tag{3}$$

with a *coarsening factor* $\beta \in \mathbb{N} \setminus \{0, 1\}$, while keeping the spatial discretization fixed. The state variable for this coarser discretized model will be denoted by $\mathbf{y}_c$, the corresponding number of discrete time steps by $M_c = M_f/\beta$, i.e., we have

$$(\mathbf{y}_c)_j = ((y_c)_{ji})_{i=1,\ldots,I}, \quad j = 1, \ldots, M_c, \quad \mathbf{y}_c \in \mathbb{R}^{M_c I}, \quad I = n_z n_t \,. \tag{4}$$

Note that the parameters $\mathbf{u}$ for this model are the same as for the fine one.

Clearly, the choice of the temporal discretization, or equivalently, the coarsening factor β, determines the quality of the coarse model and of a surrogate if based upon the latter one. Moreover, both the computational cost, the performance and quality of the solution obtained by a SBO process might be affected.

Altogether, we seek for a reasonable trade-off between the accuracy and speed of the coarse model. From numerical experiments, a value of $\beta = 40$ turned out be a reasonable choice, as was shown in [14]. Numerical results presented in Section 4 demonstrate that such a coarse model leads to a reliable approximation of the original fine ecosystem model when a response correction technique as described in this paper is utilized. Furthermore, it was observed that, for this specific choice of β, while additionally restricting the parameter u_1, i.e., the sinking velocity, using an appropriate upper bound, the resulting model response does not show any numerical instabilities.

3 Optimization Problem

The task of parameter optimization in climate science typically is to minimize a least-squares type cost function measuring the misfit between the discrete model output $\mathbf{y} = \mathbf{y}(\mathbf{u})$ and given observational data $\mathbf{y}_d$ [2,21]. In most cases, the problem is constrained by parameter bounds. The optimization problem can generally be written as

$$\min_{\mathbf{u} \in U_{ad}} \; J(\,\mathbf{y}(\mathbf{u})\,), \tag{5}$$

where

$$\begin{aligned} &J(\,\mathbf{y}\,) := ||\,\mathbf{y} - \mathbf{y}_d\,||^2, \\ &U_{ad} := \{\mathbf{u} \in \mathbb{R}^n : \mathbf{b}_l \leq \mathbf{u} \leq \mathbf{b}_u\}, \mathbf{b}_l, \mathbf{b}_u \in \mathbb{R}^n,\ \mathbf{b}_l < \mathbf{b}_u \,. \end{aligned} \tag{6}$$

The inequalities in the definition of the set U_{ad} of admissible parameters are meant component-wise. The functional J may additionally include a regularization term for the parameters. However, from numerical experiments, it turned out that such a term is not necessary to ensure a well performing optimization process.

Additional constraints on the state variable $\mathbf{y}$ might be necessary, e.g., to ensure non-negativity of the temperature or of the concentrations of biogeochemical quantities. In our example model, however, by using appropriate parameter bounds $\mathbf{b}_l$ and $\mathbf{b}_u$, non-negativity of the state variables can be ensured. This was already observed and used in [16].

4 Surrogate-Based Optimization

For many nonlinear optimization problems, a high computational cost of evaluating the objective function and its sensitivity, and, in some cases, the lack of sensitivity information, is a major bottleneck. The need for decreasing the computational cost of the optimization process is especially important while handling complex three-dimensional models.

Surrogate-based optimization [1,6,9,15] is a methodology that addresses these issues by replacing the original high-fidelity or fine model $\mathbf{y}$ by a surrogate, in the following denoted by $\mathbf{s}$, a computationally cheap and yet reasonably accurate representation of $\mathbf{y}$.

Surrogates can be created by approximating sampled fine model data (*functional* surrogates). Popular techniques include polynomial regression, kriging, artificial neural networks and support vector regression [15,18,19]. Another possibility, exploited in this work, is to construct the surrogate model through appropriate correction/alignment of a low-fidelity or coarse model (*physics-based* surrogates) [20].

Physics-based surrogates inherit physical characteristics of the original fine model so that only a few fine model data is necessary to ensure their good alignment with the fine model. Moreover, generalization capability of the physics-based models is typically much better than for functional ones. As a results, SBO schemes working with this type of surrogates normally require small number of fine model evaluations to yield a satisfactory solution. On the other hand, their transfer to other applications is less straightforward since the underlying coarse model and chosen correction approach is rather problem specific. The specific correction technique exploited in this work is recalled in Section 4.1 (see also [14]).

The surrogate model is updated at each iteration k of the optimization algorithm, typically using available fine model data from the current and/or also from previous iterates. The next iterate, $\mathbf{u}_{k+1}$, is obtained by optimizing the surrogate $\mathbf{s}_k$, i.e.,

$$\mathbf{u}_{k+1} = \operatorname*{argmin}_{\mathbf{u} \in U_{ad}} J\left(\mathbf{s}_k(\mathbf{u})\right), \tag{7}$$

where, again U_{ad} denotes the set of admissible parameters. The updated surrogate $\mathbf{s}_{k+1}$ is determined by re-aligning the coarse model at $\mathbf{u}_{k+1}$ and optimized again as in (7). The process of aligning the coarse model to obtain the surrogate and subsequent optimization of this surrogate is repeated until a user-defined termination condition is satisfied, which can be based on certain convergence criteria, assumed level of cost function

value or a specific number of iterations (particularly if the computational budget of the optimization process is limited).

If the surrogate $\mathbf{s}_k$ satisfies so-called zero-order and first-order consistency conditions with the fine model at $\mathbf{u}_k$, i.e.,

$$\mathbf{s}_k(\mathbf{u}_k) = \mathbf{y}_f(\mathbf{u}_k), \quad \mathbf{s}'_k(\mathbf{u}_k) = \mathbf{y}'_f(\mathbf{u}_k), \tag{8}$$

with $\mathbf{y}'$ and $\mathbf{s}'_k(\mathbf{u}_k)$ denote the derivatives of the responses, the surrogate-based scheme (7) is provable convergent to at least a local optimum of (5) under mild conditions regarding the coarse and fine model smoothness, and provided that the surrogate optimization scheme (7) is enhanced by the trust-region (TR) safeguard, i.e.,

$$\mathbf{u}_{k+1} = \underset{\substack{\mathbf{u}\in U_{ad},\\ \|\,\mathbf{u}-\mathbf{u}_k\,\| \le \delta_k}}{\operatorname{argmin}} J\left(\,\mathbf{s}_k(\mathbf{u})\,\right), \tag{9}$$

with δ_k being the trust-region radius updated according to the TR rules. We refer the reader to e.g. [4,8] for more details.

4.1 Surrogate Model Using Basic Multiplicative Response Correction

It has been found in [14] that a natural way of constructing the surrogate would be *multiplicative response correction*. This approach is motivated by the fact that the qualitative relation of the fine and coarse model response is rather well preserved (at least locally) while moving from one parameter vector to another. As a result, a multiplicative correction allows constructing a surrogate model with a good generalization capability. The technique is briefly recalled below.

The surrogate response $\mathbf{s}_k(\mathbf{u})$, at iteration k of the optimization process, is generated by multiplicative correction of the *smoothed* coarse model response, denoted by $\tilde{\mathbf{y}}_c$, which we briefly formulate as

$$\left.\begin{aligned} \bar{\mathbf{s}}_k(\mathbf{u}) &:= \mathbf{a}_k\,\tilde{\mathbf{y}}_c(\mathbf{u}), \\ \mathbf{a}_k &:= \frac{\tilde{\mathbf{y}}_f^{\beta}(\mathbf{u}_k)}{\tilde{\mathbf{y}}_c(\mathbf{u}_k)} \end{aligned}\right\} \quad \begin{aligned} k &= 1, 2, \ldots \\ \beta &= M_f/M_c \end{aligned} \tag{10}$$

where the operations in (10) are meant point-wise and where a_k denote the *correction factors* which are included in the vector $\mathbf{a}_k$. They are defined as the point-wise division of the smoothed and *down-sampled* fine model response, denoted by $\tilde{\mathbf{y}}_f^{\beta}$, by the smoothed coarse model response at the point $\mathbf{u}_k$.

It was observed that smoothing allows us to remove the numerical noise from the coarse model response and identify the main characteristics of the traces of interest (see [14] for details). The fine model response is smoothed accordingly in the formulation (10).

Down-sampling was necessary to make the fine model response commensurable with the corresponding response of the coarse model. The down-sampled fine model response $\mathbf{y}_f^{\beta}$ is simply given as

$$y_{ji}^{\beta} := y_{\beta j,i}, \quad j = 1, \ldots, M_c, \quad i = 1, \ldots, I\,. \tag{11}$$

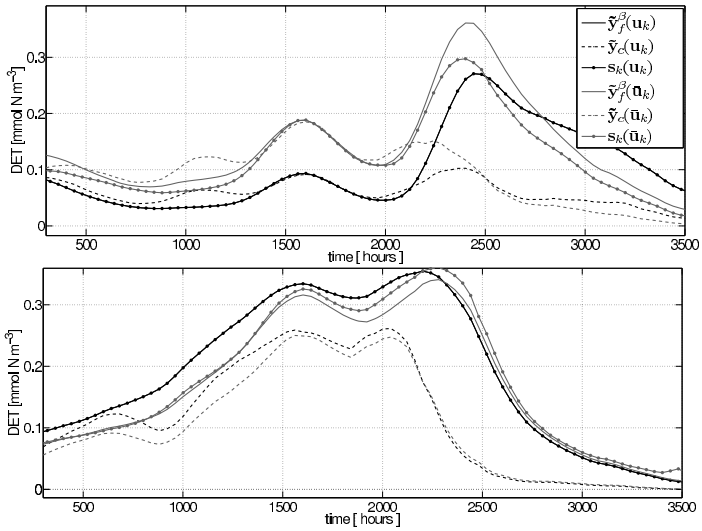


Fig. 1. Surrogate's, fine (down-sampled, smoothed) and coarse (smoothed) model responses $\mathbf{s}_k, \tilde{\mathbf{y}}_f^{\beta}$ and $\tilde{\mathbf{y}}_c$ for the tracer detritus at the uppermost depth layer at two points $\mathbf{u}_k$ and corresponding perturbation $\bar{\mathbf{u}}_k$, illustrating the generalization capability of the surrogate

By definition, the surrogate model is zero-order consistent with the (down-sampled and smoothed) fine model in the point $\mathbf{u}_k$, i.e.,

$$\mathbf{s}_k(\mathbf{u}_k) = \tilde{\mathbf{y}}_f^{\beta}(\mathbf{u}_k)\,. \tag{12}$$

As we do not use sensitivity information from the fine model, the first-order consistency condition cannot be satisfied exactly. Nevertheless, as was shown in [14], this surrogate model exhibits quite good generalization capability, which means that the surrogate provides a reasonable approximation of the fine one in the neighborhood of $\mathbf{u}_k$.

Figure 1 shows the surrogate's, fine (down-sampled) and coarse model responses $\mathbf{s}_k, \tilde{\mathbf{y}}_f^{\beta}$ and $\tilde{\mathbf{y}}_c$ at two different points, $\mathbf{u}_k$ and $\bar{\mathbf{u}}_k$. The surrogate model is established at $\mathbf{u}_k$ and, therefore, its response is perfectly aligned with the one of the fine model at $\mathbf{u}_k$, whereas its prediction is still reasonably accurate at $\bar{\mathbf{u}}_k$.

Note that only the selected tracers for a chosen section in the whole time interval and at one selected depth layer are shown. The total dimension of the model response is too large to present a full response here. We emphasize that shown responses are representative for the overall qualitative behavior the other tracers, time sections and depth layers.

4.2 Difficulties of Basic Surrogate Formulation

Occasionally, when using the surrogate given in (10), there might occur a situation where the coarse model response is close to zero (and maybe even negative due to approximation errors) and a few magnitudes smaller than the fine one, which leads to

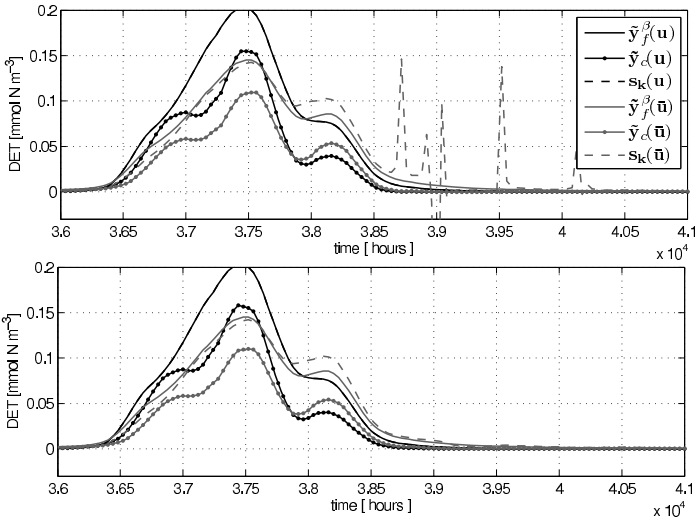


Fig. 2. Responses as in Figure 1 for a different time interval using the basic surrogate formulation (10) (top) and exploiting the modifications (13) of the response correction scheme (bottom)

large (possibly negative) correction factors a_k. While such a correction ensures zero-order consistency at the point where it was established (i.e., $\mathbf{u}_k$), it may lead to (locally) poor approximation in the vicinity of $\mathbf{u}_k$.

Figures 2 and 3 (top) illustrate these issues by showing the smoothed surrogate's, fine (down-sampled) and coarse model responses $\mathbf{s}_k, \tilde{\mathbf{y}}_f^{\beta}$ and $\tilde{\mathbf{y}}_c$ for the state detritus at one illustrative time interval and depth layer. Shown are the model responses at the same points $\mathbf{u}_k$ and its neighborhood $\bar{\mathbf{u}}_k \in B_\delta(\mathbf{u}_k)$ as in Figure 1.

It should be pointed out that the overall shape of the surrogate's response still provides a reasonable approximation of the fine model one (and more accurate than the corresponding coarse model response) despite of the distortion illustrated in Figures 2 and 3. This is supported by the fact that even without addressing these issues, the SBO was able to yield satisfactory results, not only with respect to the quality of the final solution, but, most importantly, in terms of the low computational cost of the optimization process. This was already demonstrated in [14].

4.3 Improved Response Correction Scheme

The response distortion described in the previous section is problematic towards the end of the surrogate-based optimization run when a higher accuracy of the surrogate is required to locate the fine model optimum more accurately. The "'spikes"' appearing in the response due to large values of the correction term can be viewed, in a way, as a numerical noise that slows down the algorithm convergence and makes the optimum more difficult to locate.

A few simple means described below can address these issues and further improve the accuracy of the surrogate's response as well as the performance of the optimization algorithm. We introduce non-negative bounds for the coarse model response (the

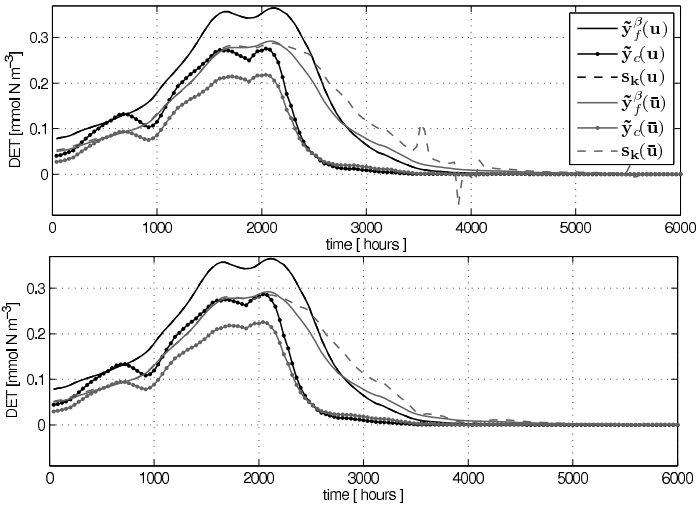


Fig. 3. Responses as in Figure 2, but for yet another section within the whole time interval. Again, after employing the improvements in (13), the positive and negative peaks are removed (bottom).

negative response is non-physical and is a result of numerical errors due to using large time steps in the numerical solution of the coarse model) and an upper bound a_{ub} for the correction factors. We furthermore restrict the correction factors to one in case the fine and coarse model responses are below a certain threshold ϵ which should be of the order of the discretization error below which the responses can be treated as zero.

More specifically, the following modifications of the model outputs and the scaling factors are performed for each iteration k

$$(i)\ \mathbf{y}_c = \begin{cases} 0; & \text{if } \mathbf{y}_c \le 0 \\ \mathbf{y}_c; & \text{else} \end{cases}, \quad (ii)\ \mathbf{a}_k = \begin{cases} a_{\text{ub}}; & \text{if } \mathbf{a}_k \ge a_{\text{ub}} \\ \mathbf{a}_k; & \text{else} \end{cases}, \tag{13}$$

$$(iii)\ \mathbf{a}_k = 1 \text{ if } (\tilde{\mathbf{y}}_f^{\beta} \le \epsilon \text{ and } \tilde{\mathbf{y}}_c \le \epsilon),$$

where the operations are again meant point-wise and where (i) is applied before smoothing. From numerical experiments, $a_{\text{ub}} = 10$ turned out to be a reasonable choice and we furthermore consider $\epsilon = 10^{-4}$.

Figure 2 (bottom) shows the surrogate's, fine (down-sampled) and coarse model response for the same illustrative tracer, time interval and depth layer, however, while employing the improvements given in (13). It can be observed that the positive and negative peaks present in the surrogate responses shown in Figure 2 (top) are removed after applying (13). As additional evidence, Figure 3 (bottom) shows the same model responses but for a different section within the whole time interval.

The numerical results presented in Section 5 demonstrate that this enhanced response correction scheme allows us to further improve the computational efficiency of the SBO.

5 Numerical Results

For all optimization runs, we use the MATLAB[1] function `fmincon`, exploiting the active-set algorithm. The following cost functions

$$J(\mathbf{z}) := \| \mathbf{z} - \mathbf{y}_d \|^2 = \sum_{i=1}^{I} \sum_{j=1}^{M_c} \left(z_{ji} - (y_d)_{ji} \right)^2 , \tag{14}$$

$$\tilde{J}(\mathbf{z}) := \| \mathbf{z} - \tilde{\mathbf{y}}_d \|^2 = \sum_{i=1}^{I} \sum_{j=1}^{M_c} \left(z_{ji} - (\tilde{y}_d)_{ji} \right)^2 , \tag{15}$$

were the target data – as a test case – is given by model generated, attainable data as

$$\mathbf{y}_d := \mathbf{y}_f^{\beta}(\mathbf{u}_d) .$$

For the optimization runs presented in this paper we employ the following cost functions: for the fine model optimization, we use (14) with $\mathbf{z} = \mathbf{y}_f^{\beta}$, for the coarse model optimization, (15) with $\mathbf{z} = \tilde{\mathbf{y}}_c$ and for the SBO, (15) with $\mathbf{z} = \mathbf{s}_k$, whereas (14) was used in the termination condition and to compare the results and where the down-sampled fine model response $\mathbf{y}_f^{\beta}$ is defined by (11). Sampling was necessary to yield a comparable fine model optimization run while in (15) the smoothed target data is considered accordingly, since the coarse model and thus also the surrogate's response are smoothed. Note that the cost functions we employ are not normalized by the total number of discrete model points. The dimension of the responses is of the order of 10^5. Clearly, this has to be taken into account for presented cost function values in the following.

We perform an exemplary direct fine and coarse model optimization as well as a SBO based on the surrogate in (10) exploiting the original and improved response correction scheme (cf. Sections 4.1, 4.3). In the following, the solutions of the four optimization runs are compared through visual inspection of the (down-sampled) fine model response $\mathbf{y}_f^{\beta}$ and the corresponding cost function value $J(\mathbf{y}_f^{\beta})$ (cf. (14)) at the respective optima.

The optimization cost is measured in *equivalent* fine model evaluations which are determined taking into account the coarsening factor β. More specifically, one evaluation of the coarse model with a coarsening factor β is equivalent to $1/\beta$ evaluations of the fine model. On the other hand, the cost of one iteration of the SBO (in terms of equivalent fine model evaluations) equals to the number of coarse model evaluations necessary to optimize the surrogate model divided by this factor β, and increased by the cost for the response correction. Recall that the specific correction (10) we use in this paper requires one fine model evaluation only.

Figure 4 shows the value of the cost function $J(\mathbf{y}_f^{\beta})$ versus the equivalent number of fine model evaluations for the SBO algorithm using the surrogate model exploiting the original and the improved correction scheme, as well as for the fine and coarse model optimization. Points 1 and 3 in Figure 4 indicate those solutions obtained in the SBO

[1] MATLAB is a registered trademark of The MathWorks, Inc., `http://www.mathworks.com`

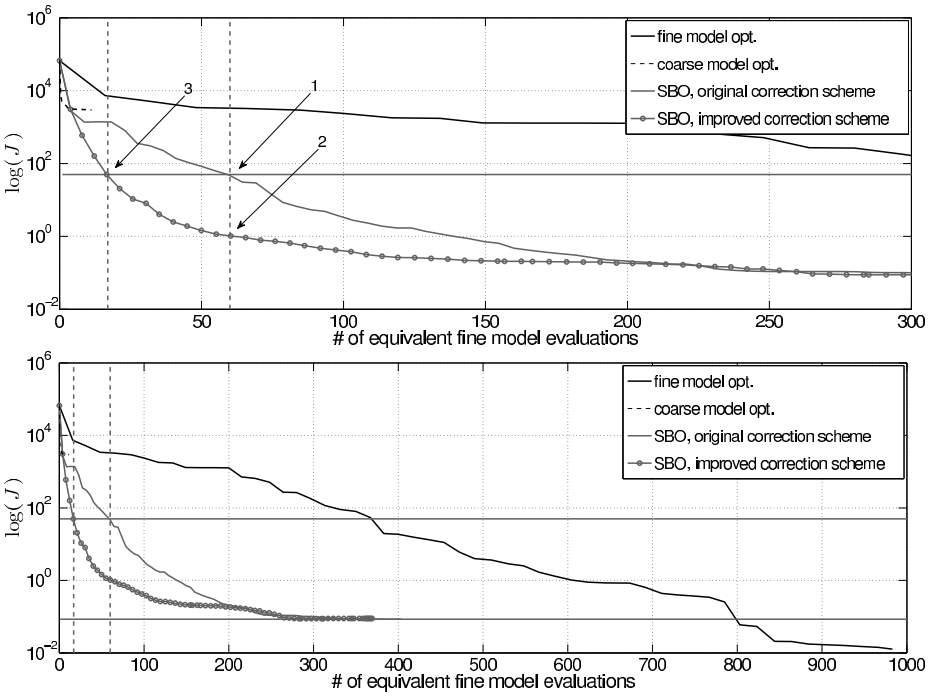


Fig. 4. Values of the cost function J versus the optimization cost measured in *equivalent number* of fine model evaluations for an exemplary SBO run exploiting the original and the improved correction scheme, as well as for a fine and coarse model optimization run. Points 1 and 3 correspond to a termination condition of $J \leq 50$ (upper horizontal line), ensuring good visual agreement between the fine model output and the target. Solution at point 2 in the improved SBO is significantly more accurate and obtained at the same cost as the one at point 1. Overall, SBO converges to a cost function value of to $J \approx 10^{-1}$ (lower horizontal line).

runs that correspond to a termination condition of $J \leq 50$. This particular value was selected as it ensures good visual agreement between the fine model output and the target. Point 2 denotes the solution in the improved SBO run which could be obtained at the same optimization cost as the one at point 1 in the original SBO run.

Figure 5 shows the fine model response at the solutions $\mathbf{u}_{s1}^*$ and $\mathbf{u}_{s2}^*$ (corresponding to points 1 and 2 in Figure 4) obtained using the SBO algorithm with the original and improved response correction scheme (cf. Sections 4.1 and 4.3) as well the responses at the solutions $\mathbf{u}_f^*, \mathbf{u}_c^*$ of a direct fine and coarse model optimization. For illustration, responses for two representative tracers and for a selected depth level and time interval are shown. Corresponding parameter values are provided in Table 1.

It can be observed that coarse model optimization yields a solution far away from the target and a rather inaccurate parameter match (cf. Table 1), whereas the optimization cost of only 11 equivalent fine model evaluations is very low. However, results indicate that the accuracy of the coarse model is not sufficient to use this very model directly in an optimization.

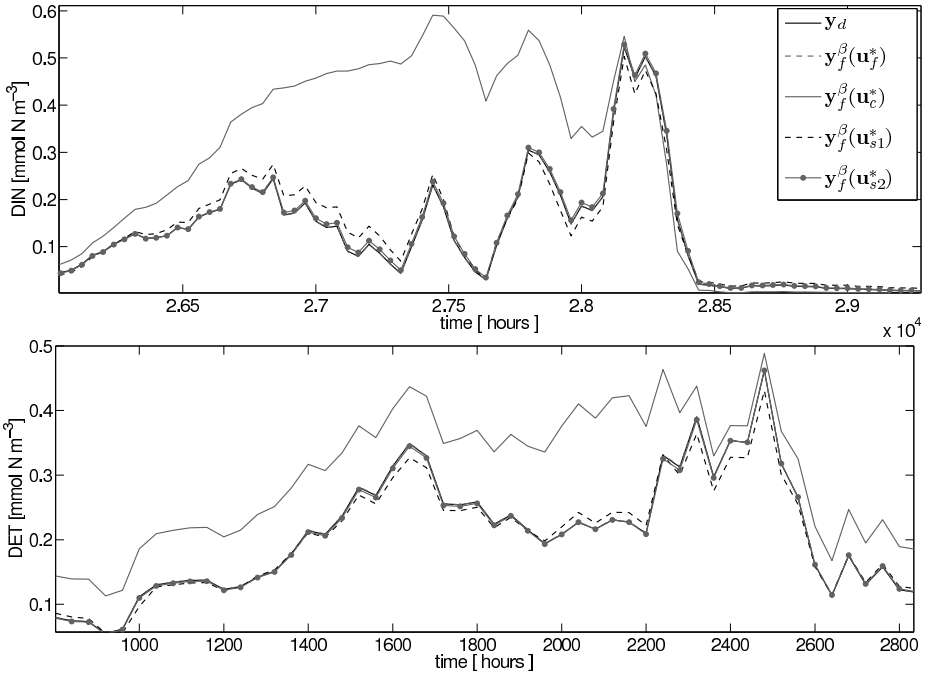


Fig. 5. Synthetic target data $\mathbf{y}_d$ at optimal parameters $\mathbf{u}_d$ and fine model response $\mathbf{y}_f^\beta$ (down-sampled) for two illustrative tracers and at the uppermost depth layer for the solutions $\mathbf{u}_f^*$, $\mathbf{u}_c^*$, $\mathbf{u}_{s1}^*$ and $\mathbf{u}_{s2}^*$ of a direct fine and coarse model optimization as well as of a SBO run exploiting the original and the improved correction scheme. Solutions $\mathbf{u}_{s1}^*$ and $\mathbf{u}_{s2}^*$ correspond to points 1 and 2 in Figure 4

On the other hand, direct fine model optimization yields a solution $\mathbf{u}_f^*$ with an almost perfect fit of the target data (cf. Figure 5) and of the optimal parameters $\mathbf{u}_d$ (cf. Table 1), corresponding to a very low cost function of $J \approx \cdot 10^{-2}$. However, the optimization cost is substantially higher: about 980 fine model evaluations.

In [14], we demonstrated that in a exemplary SBO run based on the original response correction scheme, a reasonably accurate solution $\mathbf{u}_{s1}^*$ could be obtained at the cost of approximately 60 equivalent fine model evaluations only (point 1 in Figure 4). This resulted in a significant reduction of the total optimization cost of about 84% when compared to the direct fine model optimization (correspondingly, 375 evaluations were required in the fine model optimization to reach this cost function value, cf. Figure 4).

Exploiting the improved scheme, a similarly accurate solution – both in terms of parameter match and optimal fit of the target data – can be obtained at a remarkably lower cost of only 17 equivalent fine model evaluations (point 3 in Figure 4). This is over three times less than for the original response correction scheme corresponding to a reduction of the total optimization cost of about 96%. Specific parameter values and model responses of this solution are omitted here, since they are similar to those of the original solution $\mathbf{u}_{s1}^*$.

Table 1. Solutions $\mathbf{u}_c^*$, $\mathbf{u}_f^*$, $\mathbf{u}_{s1}^*$ and $\mathbf{u}_{s2}^*$ of an illustrative coarse, fine model optimization and of a SBO run, exploiting the original and the improved correction scheme. Solutions $\mathbf{u}_{s1}^*$ and $\mathbf{u}_{s2}^*$ correspond to points 1 and 2 in Figure 4.

iterate	u_1	u_2	...									u_{12}
	SBO (original and improved scheme)											
$\mathbf{u}_{s1}^*$	0.705	0.626	0.044	0.015	0.060	0.937	1.908	0.016	0.147	0.020	0.629	4.237
$\mathbf{u}_{s2}^*$	0.738	0.604	0.028	0.010	0.036	1.024	1.678	0.010	0.206	0.020	0.541	4.318
	Coarse model optimization											
$\mathbf{u}_c^*$	0.300	1.066	0.036	0.065	0.064	0.025	0.040	0.065	0.010	0.012	0.730	3.448
	Fine model optimization											
$\mathbf{u}_f^*$	0.747	0.596	0.025	0.010	0.030	0.999	2.046	0.010	0.203	0.020	0.493	4.310
$\mathbf{u}_d$	0.750	0.600	0.025	0.010	0.030	1.000	2.000	0.010	0.205	0.020	0.500	4.320

On the other hand, when exploiting the improved correction scheme, a solution $\mathbf{u}_{s2}^*$ (point 2 in Figure 4) with a significantly higher accuracy – again both in terms of parameter match and optimal fit of the target data – can be obtained (cf. Figure 5 and Table 1) at the same cost as were required for the original one $\mathbf{u}_{s1}^*$, i.e., 60 equivalent fine model evaluations.

It should be emphasized that the surrogate model utilized in this work only satisfies zero-order consistency with the fine model. Still, as demonstrated in this section, the performance of our surrogate-based optimization process is satisfactory, particularly in terms of obtaining a good match between the model response and a given target output. Improved matching between the optimized model parameters and those corresponding to the target output could be obtained by executing larger number of SBO iterations (cf. Figure 4), which is mostly because of low sensitivity of the model with respect to some of the parameters. Also, the use of derivative information together with the trust-region convergence safeguards [4, 8] would bring further improvement in terms of matching accuracy. Clearly, the trade-offs between the accuracy of the solution and the extra computational overhead related to sensitivity calculation has to be investigated. The aforementioned issues will be the subject of future research.

6 Conclusions

Parameter identification in climate models can be computationally very expensive or even beyond the capabilities of modern computer power. Before a transient simulation of a model (e.g., used for predictions) is possible, the latter has to be calibrated, i.e., relevant parameters have to be identified using measurement data. This is the point where large-scale optimization methods become crucial for a climate system forecast.

Using the high-fidelity (or fine) model under consideration in conventional optimization algorithms that require large number of model evaluations is often infeasible. Therefore, the development of faster methods that aim at reducing the optimization cost,

such as surrogate-based optimization (SBO) techniques, are highly desirable. The idea of SBO is to replace the high-fidelity model in the optimization run by a surrogate, its computationally cheap and yet reasonably accurate representation.

As a case study, we have investigated parameter optimization of a representative of the class of one-dimensional marine ecosystem models. As demonstrated in our previous work, a simple multiplicative response correction applied to a temporally coarser discretized physics-based low-fidelity (coarse) model of the system of interest is sufficient to create a reliable surrogate of the original, high-fidelity ecosystem model, which can be used as a prediction tool to calibrate the latter. This approach allowed us to yield remarkably good results, both in terms of the quality of the final solution and, most importantly, in terms of the relative reduction in the total optimization cost, about 84% when compared to the direct fine model optimization.

In this paper, we demonstrated that the correction scheme can be enhanced to alleviate the difficulties of its original version, which results in further improvement of the surrogate model accuracy and overall performance of the optimization algorithm utilizing this surrogate. The optimization cost was reduced by a factor of three (from 16% to 5% of the direct high-fidelity model optimization optimization cost), which corresponds to the cost savings of 95%.

Improvements of the present approach by utilizing additionally sensitivity information of the low- and the high-fidelity model in the alignment of the low-fidelity model as well as trust-region convergence safeguards applied to enhance the optimization process are expected to further improve the robustness of the algorithm and the accuracy of the solution. The trade-offs between the accuracy and extra costs due too sensitivity evaluation will have to be inspected.

Acknowledgements. The authors would like to thank Andreas Oschlies, IFM Geomar, Kiel. This research was supported by the DFG Cluster of Excellence Future Ocean.

References

1. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Søndergaard, J.: Space mapping: The state of the art. IEEE T. Microw. Theory 52(1) (2004)
2. Banks, H.T., Kunisch, K.: Estimation Techniques for Distributed Parameter Systems. Birkhäuser (1989)
3. Bucker, H.M., Fortmeier, O., Petera, M.: Solving a parameter estimation problem in a three-dimensional conical tube on a parallel and distributed software infrastructure. Journal of Computational Science 2(2), 95–104 (2011); Simulation Software for Supercomputers
4. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-region methods. Society for Industrial and Applied Mathematics, Philadelphia (2000)
5. Fennel, W., Neumann, T.: Introduction to the Modelling of Marine Ecosystems. Elsevier (2004)
6. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. 45(1-3), 50–79 (2009)
7. Gill, A.E.: Atmosphere - Ocean Dynamics. International Geophysics Series, vol. 30. Academic Press (1982)
8. Koziel, S., Bandler, J.W., Cheng, Q.S.: Robust trust-region space-mapping algorithms for microwave design optimization. IEEE T. Microw. Theory 58(8), 2166–2174 (2010)

9. Leifsson, L., Koziel, S.: Multi-fidelity design optimization of transonic airfoils using physics-based surrogate modeling and shape-preserving response prediction. Journal of Computational Science 1(2), 98–106 (2010)
10. Majda, A.: Introduction to PDE's and Waves for the Atmosphere and Ocean. AMS (2003)
11. Marchuk, G.I.: Methods of Numerical Mathematics, 2nd edn. Springer (1982)
12. McGuffie, K., Henderson-Sellers, A.: A Climate Modelling Primer, 3rd edn. Wiley (2005)
13. Oschlies, A., Garcon, V.: An eddy-permitting coupled physical-biological model of the north atlantic. 1. sensitivity to advection numerics and mixed layer physics. Global Biogeochem. Cy. 13, 135–160 (1999)
14. Prieß, M., Koziel, S., Slawig, T.: Surrogate-based optimization of climate model parameters using response correction. Journal of Computational Science (2011) (in press)
15. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P.K.: Surrogate-based analysis and optimization. Prog. Aerosp. Sci. 41(1), 1–28 (2005)
16. Rückelt, J., Sauerland, V., Slawig, T., Srivastav, A., Ward, B., Patvardhan, C.: Parameter optimization and uncertainty analysis in a model of oceanic CO_2-uptake using a hybrid algorithm and algorithmic differentiation. Nonlinear Analysis B Real World Applications 10(1016), 3993–4009 (2010)
17. Sarmiento, J.L., Gruber, N.: Ocean Biogeochemical Dynamics. Princeton University Press (2006)
18. Simpson, T.W., Poplinski, J.D., Koch, P.N., Allen, J.K.: Metamodels for computer-based engineering design: Survey and recommendations. Eng. Comput. 17, 129–150 (2001), 10.1007/PL00007198
19. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. 14, 199–222 (2004), 10.1023/B:STCO.0000035301.49549.88
20. Søndergaard, J.: Optimization using surrogate models - by the space mapping technique. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, Supervisor: Kaj Madsen (2003)
21. Tarantola, A.: Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM (2005)

Hydrodynamic Shape Optimization of Axisymmetric Bodies Using Multi-fidelity Modeling

Leifur Leifsson, Slawomir Koziel, and Stanislav Ogurtsov

Engineering Optimization & Modeling Center, School of Science and Engineering, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland
{leifurth,koziel,stanislav}@ru.is

Abstract. Hydrodynamic shape optimization of axisymmetric bodies is presented. A surrogate-based optimization algorithm is described that exploits a computationally cheap low-fidelity model to construct a surrogate of an accurate but CPU-intensive high-fidelity model. The low-fidelity model is based on the same governing equations as the high-fidelity one, but exploits coarser discretization and relaxed convergence criteria. A multiplicative response correction is applied to the low-fidelity CFD model output to yield an accurate and reliable surrogate model. The approach is implemented for both direct and inverse design. In the direct design approach the optimal hull shape is found by minimizing the drag, whereas in the inverse approach a target pressure distribution is matched. Results show that optimized designs are obtained at substantially lower computational cost (over 94%) when compared to the direct high-fidelity model optimization.

Keywords: Shape Optimization, Surrogate-based Optimization, Multi-fidelity Modeling, Axisymmetric Body, CFD, Direct Design, Inverse Design.

1 Introduction

Autonomous underwater vehicles (AUVs) are becoming increasingly important in various marine applications, such as oceanography, pipeline inspection, and mine counter measures [1]. Endurance (speed and range) is one of the more important attribute of AUVs [2]. Vehicle drag reduction and/or an increase in the propulsion system efficiency will translate to a longer range for a given speed (or the same distance in a reduced time). A careful hydrodynamic design of the AUVs, including the hull shape, the protrutions, and the propulsion system, is therefore essential.

The fluid flow around an underwater vehicle with appendages is characterized by flow features such as thick boundary layers, vortices and turbulent wakes generated due to the hull and the appendages [3]. These flow features can have adverse effects on, for example, the performance of the propulsion system and the control planes. Moreover, the drag depends highly on the vehicle shape, as well as on the aforementioned flow features. Consequently, it is important to account for these effects during the design of the AUVs.

The prediction of the flow past the full three-dimensional configuration of the AUVs requires the use of computational fluid dynamics (CFD). Numerous

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 209–223.
DOI: 10.1007/978-3-642-34336-0_14

applications of CFD methods to the flow past AUVs and other underwater vehicles are in the literature, see, e.g., [4-6]. The purpose of these investigations is to predict properties such as added masses, pressure and friction distributions, drag, normal force and moment coefficients, wake field, and stability derivatives. Comparison with experimental measurements show that CFD is reliable and can yield accurate results.

Numerous studies on underwater vehicle design and optimization have been reported which focus on the clean hull only, i.e., the appendages and the propulsion system are neglected and the flow is taken to be past an axisymmetric body at a zero angle of attack. Examples of such numerical studies can be found in [7-13]. Allen et al. [2], on the other hand, report an experimental investigation of propulsion system enhancements and drag reduction of an AUV.

The hydrodynamic design optimization of AUVs in full configuration, taking into account the appendages and the propulsion system, is still an open problem. One of the main challenges involved is the high computational cost of a CFD simulation. A single CFD simulation of the three-dimensional flow past an AUV can take a few hours up to several days, depending on the computational power, the grid density, and the flow conditions. Therefore, the direct optimization, as shown in Fig. 1(a), can be impractial, especially using conventional gradient-based methods.

An important research area in the field of aerodynamic optimization is focused on employing the surrogate-based optimization (SBO) techniques [14,15]. One of the major objectives is to reduce the number of high-fidelity model evaluations, and thereby making the optimization process more efficient. In SBO, the accurate, but computationally expensive, high-fidelity CFD simulations are replaced—in the optimization process—by a cheap surrogate model (Fig. 1(b)). SBO has been successfully applied to the aerodynamic design optimization of various aerospace components, such as airfoils [16], aircraft wings [17], and turbine blades [18].

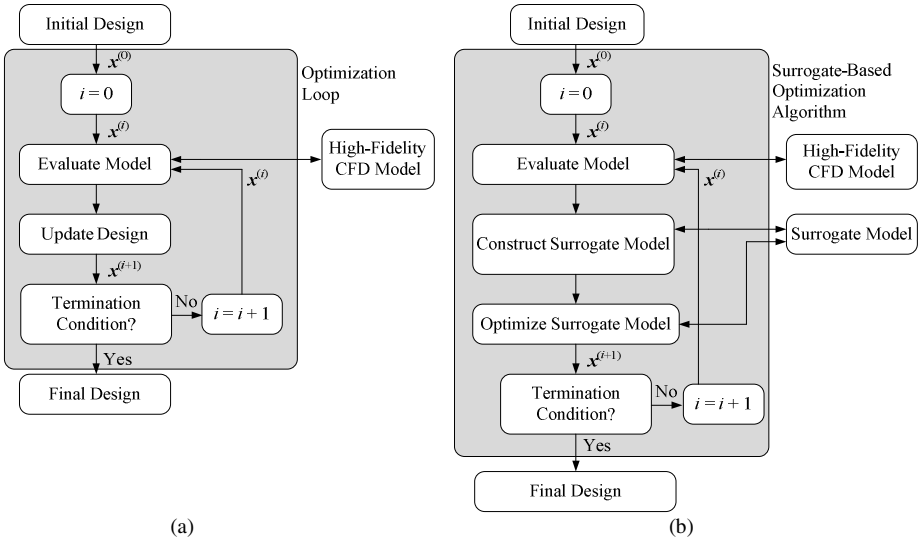


Fig. 1. (a) Direct optimization, (b) surrogate-based optimization. Here, $\boldsymbol{x}^{(i)}$ denotes the design variable vector at iteration i.

The surrogate models can be created either by approximating the sampled high-fidelity model data using regression (so-called function approximation surrogates) (see for example [14]), or by correcting physics-based low-fidelity models which are less accurate but computationally cheap representations of the high-fidelity models (see, e.g., [17] and [19]). The latter models are typically more expensive to evaluate. However, less high-fidelity model data is normally needed to obtain a given accuracy level. SBO with physics-based low-fidelity models is called multi- or variable-fidelity optimization.

In this paper, we present a hydrodynamic shape optimization methodology based on the SBO concept for AUVs. In particular, we adopt the multi-fidelity approach with the high-fidelity model based on the Reynolds-Averaged Navier-Stokes (RANS) equations, and the low-fidelity model based on the same equations, but with coarse discretization and relaxed convergence criteria. We use a simple response correction to create the surrogate. Here, we choose to focus on the clean hull design, which is a convenient case study to implement and test our design approach.

2 Hydrodynamic Shape Optimization

In this work, we focus on efficient shape optimization involving computationally heavy high-fidelity CFD simulations. This section describes the problem formulation and outlines the solution approach.

2.1 Problem Formulation

The goal of hydrodynamic shape optimization is to find an optimal—with respect to given objectives—hull shape, so that given design constraints are satisfied. The general design problem can be formulated as a nonlinear minimization problem, i.e.,

$$\begin{aligned} &\min_{\mathbf{x}} f(\boldsymbol{x}) \\ &s.t.\ g_j(\boldsymbol{x}) \le 0,\ j = 1,\ldots,M \\ &\quad h_k(\boldsymbol{x}) = 0,\ k = 1,\ldots,N \\ &\quad \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u} \end{aligned} \tag{1}$$

where $f(\boldsymbol{x})$ is the objective function, $\boldsymbol{x}$ is the design variable vector, typically containing relevant geometry parameters of the fluid system under consideration, $g_j(\boldsymbol{x})$ are the inequality constraints, M is the number of the inequality constraints, $h_k(\boldsymbol{x})$ are the equality constraints, N is the number of the equality constraints, and $\boldsymbol{l}$ and $\boldsymbol{u}$ are the lower and upper bounds of the design variables, respectively.

There are two main approaches two solving (1) when considering shape design. One is to adjust the geometrical shape to maximize performance. This is called direct design, and a typical design goal is drag minimization. An alternative approach is to define a priori a specific flow behavior that is to be attained. This is called inverse design, and, typically in hydrodynamic design, a target velocity distribution is prescribed [10]. Instead, a target pressure distribution can be prescribed a priori, which is more common in aerodynamic design [20]. Typically, inverse design minimizes the

norm of the difference between the target and design distributions. The main difficulty in this approach is to define the target distribution.

In this paper, we apply both the direct and the inverse approaches to the hydrodynamic shape optimization of axisymmetric bodies. The direct approach involves the minimization the drag coefficient (C_D) and we set $f(\boldsymbol{x}) = C_D(\boldsymbol{x})$. The drag coefficient and other characteristic variables are defined in Sec. 3.4. In the inverse approach, a target pressure distribution ($C_{p,t}$) is prescribed a priori and the function $f(\boldsymbol{x}) = \|C_p(\boldsymbol{x}) - C_{p,t}\|^2$, where $C_p(\boldsymbol{x})$ is the hull surface pressure distribution, is minimized. No constraints are considered in this work, aside from the design variable bounds.

2.2 Solution Approach

The design problem (1) can be solved in a straightforward way using any available algorithm to directly optimize the high-fidelity CFD model (Fig. 1(a)). In many cases, this is impractical due to high computational cost of an accurate CFD simulation and the fact that conventional optimization methods (e.g., gradient-based) normally require large number of objective function evaluations to yield an optimized design [21]. This problem can be partially alleviated using cheap adjoint sensitivity [8], however, they are not always available and the number of required CFD simulations may still be prohibitively large.

Here, the design process is accelerated using surrogate-based optimization (SBO) [14], [15] exploiting physics-based low-fidelity models. The low-fidelity model inherits the knowledge about the system under consideration and it can be used to create a fast and yet accurate representation of the high-fidelity model, a surrogate. The surrogate model can be then used as a prediction tool that yields an approximate high-fidelity model optimum at a low computational cost [22]. The flow diagram of the SBO process is shown in Fig. 1(b).

The surrogate model considered in this paper is constructed using an underlying low-fidelity CFD model and a response correction technique. The low-fidelity CFD model is constructed using the high-fidelity CFD model with a coarser mesh-resolution and relaxed convergence criteria; called variable-resolution modeling [17]. A description of the variable-resolution modeling is given in Section 3. There are various response correction techniques available, and the type appropriate in each case depends on the nature of the response. A multiplicative response correction is used in this work and is described in Section 4.

3 Computational Models

In this section, we describe the major components of the computational model of the axisymmetric hull shape considered in this paper. In particular, we discuss the shape parameterization, the high- and low-fidelity CFD models, as well as calculation of the hull drag force.

3.1 Shape Parameterization

The hull shapes are constrained to the most common AUV hull shape, namely, the torpedo shape, i.e., a three section axisymmetric body with a nose, a cylindrical midsection, and a tail. Figure 2(a) shows a typical torpedo shaped hull with a nose of length a, midsection of length b, overall length L, and maximum diameter of D. The nose and the tail are parameterized using Bézier curves defined as [23]

$$B(t) = \sum_{k=1}^{m} \sum_{i=0}^{n} \frac{n!}{i!(n-i)!} (1-t(k))^{n-i} t(k)^i P(i), \tag{2}$$

where P_i, $i = 0,1,\ldots n$, are the control points, and t is an $1 \times m$ array from 0 to 1. We use five control points for the nose and four for the tail, as shown in Figs. 2(b) and 2(c). Control points number three and eight are free (x- and y-coordinates), while the other points are fixed. We, therefore, have two design variables for the nose and tail curves, a total of four design variables, aside from the hull dimensions a, b, L, and D.

3.2 High-Fidelity CFD Model

The flow past the hull is considered to be steady and incompressible. The Reynolds-Averaged Navier-Stokes (RANS) equations are assumed as the governing flow equations with the two-equation k-ε turbulence model with standard wall functions (see, e.g., [24]).

The solution domain is axisymmetric around the hull centreline axis and extends two body lengths in front of the hull, five body lengths behind it, and two body lengths above it (Fig. 3). At the inlet, there is a velocity boundary condition where the velocity is set parallel to the hull axis, i.e., zero angle of attack. Pressure is prescribed at the outlet (zero gauge pressure).

The CFD computer code FLUENT [25] is used for numerical simulations of the fluid flow. Asymptotic convergence to a steady state solution is obtained for each case. The iterative convergence of each solution is examined by monitoring the overall residual, which is the sum (over all the cells in the computational domain) of the L^2 norm of all the governing equations solved in each cell. In addition to this, the drag force (defined in Sec. 3.4) is monitored for convergence. A solution is considered converged if a residual value of 10^{-6} has been reached for all equations, or the number of iterations reaches 1000.

The computational grid is structured with quadrilateral elements. The elements are clustered around the body and grow in size with distance from the body. The grids are generated using ICEM CFD [26]. A grid convergence study was performed to determine the necessary grid density (Fig. 4). A torpedo shaped body with $L/D = 5$ was used in the study. The inlet speed was 2 *m/s* and the Reynolds number was 2 million. Clearly, the drag coefficient value has converged at the finest grids (number 1 and 2) (Fig. 4(a)). There is, however, a large difference in the simulation time between the two finest grids (Fig. 4(b)). Therefore, we selected grid number 2, with 42,763 elements, to use for the high-fidelity CFD model in the optimization process. The velocity contours, and the pressure and skin friction distributions are shown in Fig. 5 for illustration purposes.

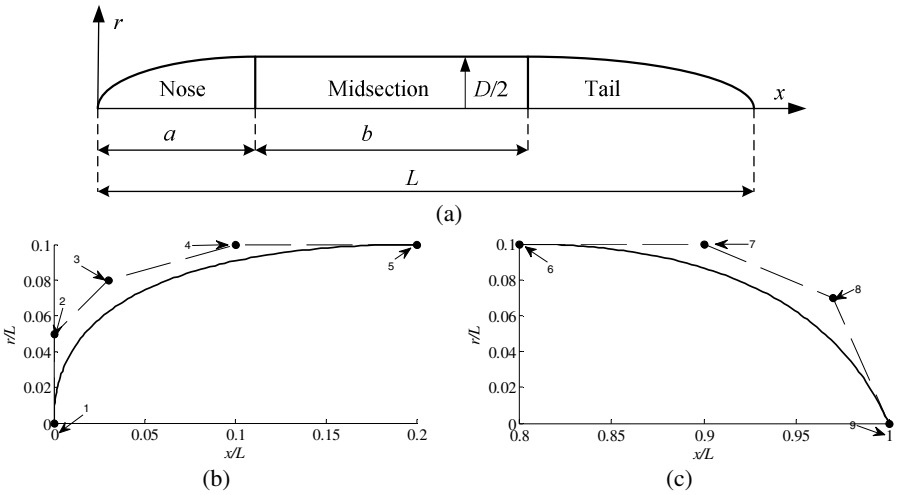


Fig. 2. (a) A sketch of a typical torpedo shaped hull form (axisymmetric with three sections: nose, middle, tail); typically, equipment such as the computer, sensors, electronics, batteries, and payload are housed in the nose and the midsection, whereas the propulsion system is in the tail; (b) Bézier curve representing the nose (5 control points); and (c) Bézier curve representing the tail (4 control points). Control points 3 and 8 are free, while the other points are essentially fixed (depend on L, a, b, and D).

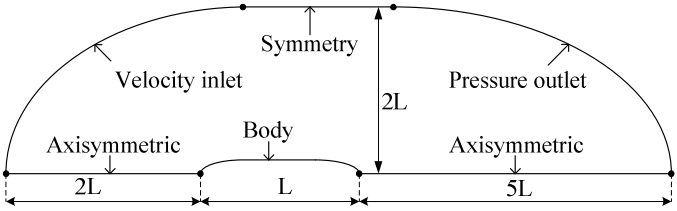


Fig. 3. The computational solution domain and the boundary conditions

3.3 Low-Fidelity CFD Model

The most important component of the design optimization methodology described in this work is a low-fidelity model, which is a simplified representation of the high-fidelity one. The two most important features for the low-fidelity model to be efficiently used in the surrogate-based optimization process is that it should be computationally much cheaper than the high-fidelity model, and, at the same time, contain sufficient knowledge about the latter so that it can be used as a reliable prediction tool to yield an approximate location of the optimum design.

To satisfy the aforementioned requirements, the low-fidelity model is based on the same CFD model as the high-fidelity one. The simulation time is substantially reduced by making the grid coarser (Fig. 4(b)). Grid number 6 needs the lowest simulation time and is the least accurate. A closer look at that grid reveals that it is too coarse (the responses were too "grainy"). Consequently, we selected grid number 5, with 504 elements, to be used for the low-fidelity model.

The simulation time can be reduced further by reducing the number of iterations. Figure 6 shows how the drag coefficient reaches a converged value after approximately 50 iterations. We therefore relax the convergence criteria for the low-fidelity model by setting it to 50 iterations. The ratio of simulation time of the high-fidelity model to the low-fidelity model is around 15. A comparison of the high- and low-fidelity responses is given in Figs. 5(a) and 5(b).

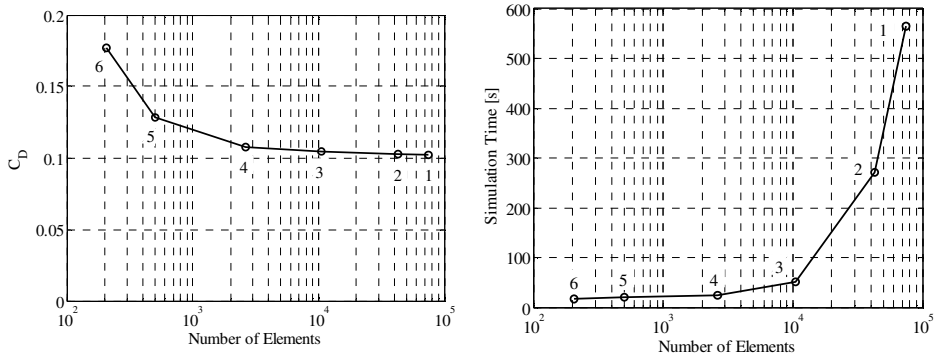


Fig. 4. Grid convergence study for a torpedo shaped hull with length to diameter ratio L/D = 5 at a speed of 2 m/s and Reynolds number of 2 million; (a) the change in the drag coefficient C_D (defined in Section X) with the number of elements; (b) the variation in the simulation time with number of elements

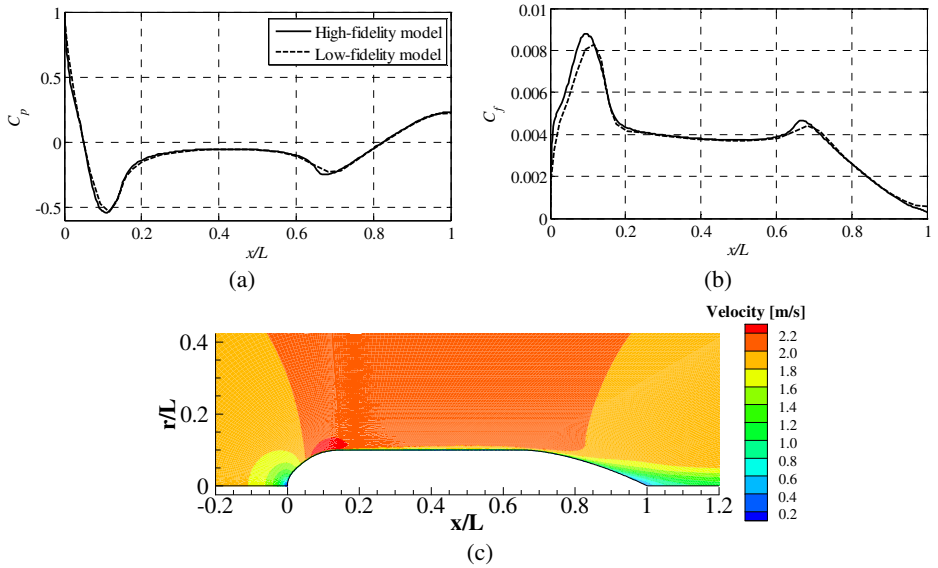


Fig. 5. (a) Comparison of the high- and low-fidelity model responses of the axisymmetric hull of diameter ratio L/D = 5 at a speed of 2 m/s and Reynolds number of 2 million, (a) pressure distributions, and (b) skin friction distributions. (c) Velocity contours of the flow past an axisymmetric torpedo shape obtained by the high-fidelity CFD model.

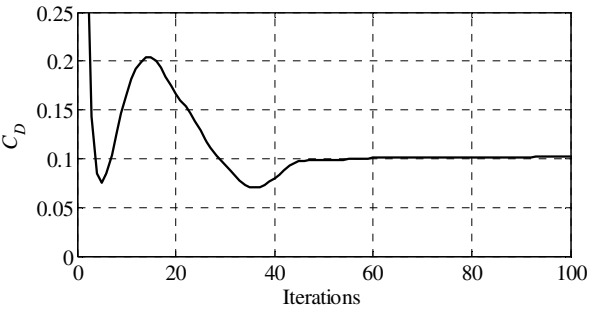


Fig. 6. Variation of the drag coefficient with number of iterations for the case shown in Fig. 5

3.4 Hull Drag Calculation

For a body in incompressible flow, the total drag is due to pressure and friction forces, which are calculated by integrating the pressure (C_p) and skin friction (C_f) distributions over the hull surface. The pressure coefficient is defined as $C_p \equiv (p\text{-}p_\infty)/q_\infty$, where p is the local static pressure, p_∞ is free-stream static pressure, and $q_\infty = (1/2\rho_\infty V_\infty^2)$ is the dynamic pressure, with ρ_∞ as the free-stream density, and the V_∞ free-stream velocity. The skin friction coefficient is defined as $C_f \equiv \tau/q_\infty$, where τ is the shear stress. Typical C_p and C_f distributions are shown in Fig. 5.

The total drag coefficient is defined as $C_D \equiv d/(q_\infty S)$, where d is the total drag force, and S is the reference area. Here, we use the frontal-area of the hull as the reference area. The drag coefficient is the sum of the pressure and friction drag, or

$$C_D = C_{Dp} + C_{Df} \,, \tag{3}$$

where C_{Dp} is the pressure drag coefficient and C_{Df} is the skin friction drag coefficient. The CFD analysis yields static pressure and wall shear stress values (which are non-dimensionalized to give C_p and C_f) at the element nodes (Fig. 7). The pressure acts normal to the surface and the shear stress parallel to it. The pressure drag coefficient is calculated by integrating from the leading-edge of the nose to the trailing-edge of the tail

$$C_{Dp} = 2\pi \int_0^L C_p(x) \sin\theta(x)\, r(x)\, dx \,, \tag{4}$$

where $C_p(x)$ is assumed to vary linear between the element nodes, $\theta(x)$ is angle of each element relative to the x-axis, and L is the length of the hull. Similarly, the skin friction drag coefficient is calculated as

$$C_{Df} = 2\pi \int_0^L C_f(x) \cos\theta(x)\, r(x)\, dx \,. \tag{5}$$

4 Surrogate Modeling and Optimization

In this section, a generic surrogate-based optimization scheme exploited here, as well as the methodology for creating a surrogate model, is described. The specific method utilized to align the responses of the low- and high-fidelity model, both the hull pressure and skin friction distributions, is a multiplicative response correction.

4.1 Surrogate-Based Optimization Scheme

Our design task is formulated as a nonlinear minimization problem of the form

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{l}\le\boldsymbol{x}\le\boldsymbol{u}} f(\boldsymbol{x}) \tag{6}$$

where $f(\boldsymbol{x})$ is the objective function, $\boldsymbol{x}$ is the design variable vector, whereas $\boldsymbol{l}$ and $\boldsymbol{u}$ are the lower and upper bounds, respectively. The high-fidelity model evaluation is CPU-intensive so that solving the problem (6) directly, by plugging in the high-fidelity model into the optimization loop, may be impractical. Instead, we exploit surrogate-based optimization (SBO) [14,15] that shifts the optimization burden into the computationally cheap surrogate, and, thus, allows us to solve (6) at a low computational cost.

The generic SBO optimization scheme is the following

$$\boldsymbol{x}^{(i+1)} = \arg\min_{\boldsymbol{x}} s^{(i)}(\boldsymbol{x}) \tag{7}$$

where $\boldsymbol{x}^{(i)}$, $i = 0, 1, ...$, is a sequence of approximate solutions to (6), whereas $s^{(i)}$ is the surrogate model at iteration i. If the surrogate model is sufficiently good representation of the high-fidelity model f, the number of iterations required to find a satisfactory design is small [27].

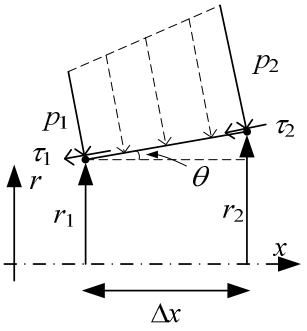


Fig. 7. Edge of an element on the hull surface at radius r. The element length is Δx and it makes an angle θ to the x-axis. Pressure p acts normal to the hull surface. Shear stress τ acts parallel to the surface.

4.2 Multiplicative Response Correction

The surrogate model can be constructed either from sampled high-fidelity model data using an appropriate approximation technique [28], or by utilizing a physically-based low-fidelity model [19]. Here, we exploit the latter approach as we have a reliable low-fidelity model at our disposal (see Sec. 3.3). Also, good physically-based surrogates can be constructed using a fraction of high-fidelity model data necessary to build accurate approximation models [29].

There are several methods of constructing the surrogate from a physically-based low-fidelity model. They include, among others, space mapping (SM) [19], various response correction techniques [30], manifold mapping [31], and shape-preserving response prediction [32]. In this paper, the surrogate model is created using a simple multiplicative response correction, which turns out to be sufficient for our purposes. An advantage of such an approach is that the surrogate is constructed using a single high-fidelity model evaluation, and it is very easy to implement.

Recall that $C_{p.f}(\boldsymbol{x})$ and $C_{f.f}(\boldsymbol{x})$ denote the pressure and skin friction distributions of the high-fidelity model. The respective distributions of the low-fidelity model are denoted as $C_{p.c}(\boldsymbol{x})$ and $C_{f.c}(\boldsymbol{x})$. We will use the notation $C_{p.f}(\boldsymbol{x}) = [C_{p.f.1}(\boldsymbol{x})\ C_{p.f.2}(\boldsymbol{x})\ ...\ C_{p.f.m}(\boldsymbol{x})]^T$, where $C_{p.f.j}(\boldsymbol{x})$ is the jth component of $C_{p.f}(\boldsymbol{x})$, with the components corresponding to different coordinates along the x/L axis.

At iteration i, the surrogate model $C_{p.s}^{(i)}$ of the pressure distribution $C_{p.f}$ is constructed using the multiplicative response correction of the form:

$$C_{p.s}^{(i)}(\boldsymbol{x}) = [C_{p.s.1}^{(i)}(\boldsymbol{x})\ \ C_{p.s.2}^{(i)}(\boldsymbol{x})\ \ ...\ \ C_{p.s.m}^{(i)}(\boldsymbol{x})]^T \tag{8}$$

$$C_{p.s.j}^{(i)}(\boldsymbol{x}) = A_{p.j}^{(i)} \cdot C_{p.c.j}(\boldsymbol{x}) \tag{9}$$

where $j = 1, 2, ..., m$, and

$$A_{p.j}^{(i)} = \frac{C_{p.f.j}^{(i)}(\boldsymbol{x}^{(i)})}{C_{p.c.j}^{(i)}(\boldsymbol{x}^{(i)})} \tag{10}$$

Similar definition holds for the skin friction distribution model $C_{f.s}^{(i)}$. Note that the formulation (8)-(10) ensures zero-order consistency [33] between the surrogate and the high-fidelity model, i.e., $C_{p.f}(\boldsymbol{x}^{(i)}) = C_{p.s}^{(i)}(\boldsymbol{x}^{(i)})$. Rigorously speaking, this is not sufficient to ensure the convergence of the surrogate-based scheme (7) to the optimal solution of (5). However, because of being constructed from the physically-based low-fidelity model, the surrogate (8)-(10) exhibits quite good generalization capability. As demonstrated in Sec. 5, this is sufficient for good performance of the surrogate-based design process.

One of the issues of model (8)-(10) is that (10) is not defined whenever $C_{p.c.j}(\boldsymbol{x}^{(i)})$ equals zero, and that the values of $A_{p.j}^{(i)}$ are very large when $C_{p.c.j}(\boldsymbol{x}^{(i)})$ is close to zero. This may be a source of substantial distortion of the surrogate model response as illustrated in Fig. 8. In order to alleviate this problem, the original surrogate model response is "smoothened" in the vicinity of the regions where $A_{p.j}^{(i)}$ is large (which indicates the problems mentioned above). Let $j_{\max}$ be such that $|A_{p.j\max}^{(i)}| >> 1$ assumes (locally) the largest value. Let Δj be the user-defined index range (typically, $\Delta j = 0.01 \cdot m$). The original values of $A_{p.j}^{(i)}$ are replaced, for $j = j_{\max}-\Delta j, ..., j_{\max}-1, j_{\max}, j_{\max}+1, ..., j_{\max}+\Delta j$, by the interpolated values:

$$\begin{aligned}\bar{A}^{(i)}_{p.j} = I(\{[j_{\max} - 2\Delta j \ldots j_{\max} - \Delta j - 1] \cup \\ \cup [j_{\max} + \Delta j - 1 \ldots j_{\max} + 2\Delta j]\}, \\ \{[A^{(i)}_{p.j_{\max}-2\Delta j} \ldots A^{(i)}_{p.j_{\max}-\Delta j-1}] \cup \\ \cup [A^{(i)}_{p.j_{\max}-2\Delta j} \ldots A^{(i)}_{p.j_{\max}-\Delta j-1}]\}, j)\end{aligned} \tag{11}$$

where $I(X,Y,Z)$ is a function that interpolates the function values Y defined over the domain X onto the set Z. Here, we use cubic splines. In other words, the values of $A_{p.j}^{(i)}$ in the neighbourhood of $j_{\max}$ are "restored" using the values of $A_{p.j}^{(i)}$ from the surrounding of $j = j_{\max}-\Delta j, \ldots, j_{\max}+\Delta j$. Figure 9(a) shows the "smoothened" surrogate model response corresponding to that of Fig. 8. Figure 9 shows the surrogate and the high-fidelity model responses, both C_p and C_f, at $\boldsymbol{x}^{(i)}$ and at some other design $\boldsymbol{x}$.

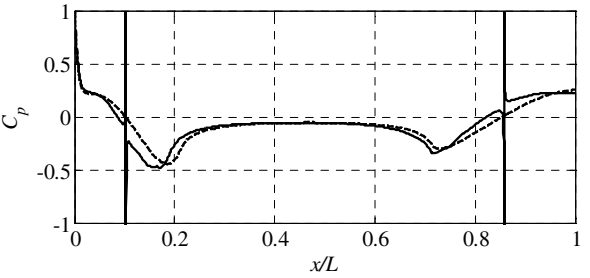


Fig. 8. Surrogate model $C_{p.s}^{(i)}$ (7)-(9) at $x^{(i)}$ (- - -), and at some other design x (—). By definition, $C_{p.s}^{(i)}(x^{(i)}) = C_{p.f}(x^{(i)})$. Note that $C_{p.s}^{(i)}(x)$ has large spikes around the points where $C_{p.s}^{(i)}(x^{(i)})$ is close to zero.

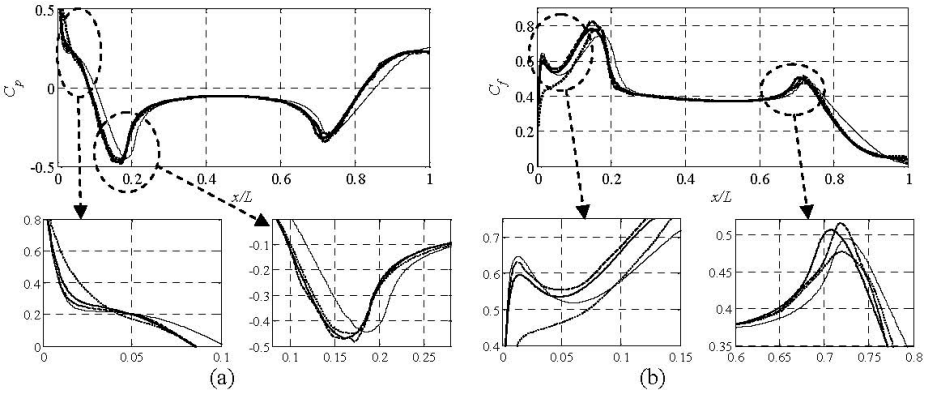


Fig. 9. (a) Smoothened surrogate model (7)-(10) $C_{p.s}^{(i)}(\boldsymbol{x}^{(i)}) = C_{p.f}(\boldsymbol{x}^{(i)})$ (—), $C_{p.s}^{(i)}(\boldsymbol{x})$ (- - -), $C_{p.c}(\boldsymbol{x})$ (· · ·), and $C_{p.s}(\boldsymbol{x})$ (—); (b) Smoothened responses $C_{f.s}^{(i)}(\boldsymbol{x}^{(i)}) = C_{f.f}(\boldsymbol{x}^{(i)})$ (—), $C_{f.s}^{(i)}(\boldsymbol{x})$ (- - -), $C_{f.c}(\boldsymbol{x})$ (· · ·), and $C_{f.s}(\boldsymbol{x})$ (—)

5 Numerical Examples

The methodology of Section 4 is applied to the hydrodynamic shape optimization of torpedo-type hulls, involving both the direct and inverse design approaches. Designs are obtained with the surrogate model optimized using the pattern-search algorithm [34]. For comparison purposes, designs obtained through direct optimization of the straightforward high-fidelity model using the pattern-search algorithm [34] are also presented.

For both the direct and the inverse design approaches, the design variable vector is $\boldsymbol{x} = [a\ x_n\ y_n\ x_t\ y_t]^T$, where a is the nose length, (x_n,y_n) and (x_t,y_t) are the coordinates of the free control points on the nose and tail Bézier curves, respectively, i.e., points 3 and 8 in Fig. 2. See Section 3.1 for a description of the shape parameterization. The lower and upper bounds of design variables are $\boldsymbol{l} = [0\ 0\ 0\ 80\ 0]^T$ *cm* and $\boldsymbol{u} = [30\ 30\ 10\ 100\ 10]^T$ *cm*, respectively. Other geometrical shape parameters are, for both cases, L = 100 cm, d = 20 *cm,* and b = 50 *cm*. The flow speed is 2 *m/s* and the Reynolds number is 2 million.

5.1 Direct Design

Numerical results for a direct design case are presented in Table 1. The hull drag coefficient is minimized by finding the appropriate shape and length of the nose and tail sections for a given hull length, diameter, and cylindrical section length. In this case, the drag coefficient is reduced by 6.3%. This drag reduction comes from a reduction in skin friction and a lower pressure peak where the nose and tail connect with the midsection (Figs. 10(a) and 10(b)). These changes are due to a more streamlined nose (longer by 6 cm) and a fuller tail, when compared to the initial design (Fig. 10(c)).

Our approach requires 3 high-fidelity and 300 low-fidelity model evaluations. The ratio of the high-fidelity model evaluation time to the corrected low-fidelity model evaluation time varies between 11 to 45, depending on whether the flow solver converges to the residual limit of 10^{-6}, or the maximum iteration limit of 1000. We express the total optimization cost of the presented method in the equivalent number of high-fidelity model evaluations. For the sake of simplicity, we use a fixed value of 30 as the high- to low-fidelity model evaluation time ratio. The results show that the total optimization cost of the presented approach is around 13 equivalent high-fidelity model evaluations. The direct optimization method, using the pattern-search algorithm [34] yields very similar design, but at the substantially higher computational cost of 282 high-fidelity model evaluations.

5.2 Inverse Design

Inverse design of the hull shape was performed by prescribing a target pressure distribution. The objective is to minimize the norm of the difference between the pressure distribution of the hull design and the target pressure distribution. The numerical results are of the inverse design are presented in Table 1.

The algorithm of Section 4 matched the target pressure distribution (the norm of the distributions is less than 2×10^{-5}) using less than 22 equivalent high-fidelity model evaluations. The direct optimization of the high-fidelity model using the pattern-search algorithm required 401 function calls to yield a comparable matching with the target.

Table 1. Numerical results for design Cases 1 and 2; axisymmetric hull direct drag minimization and inverse design with a target pressure distribution, respectively. In both cases, the flow speed is 2 *m/s* and the Reynolds number is $2 \cdot 10^6$. All the numerical values are from the high-fidelity model. N_c and N_f are the number of low- and high-fidelity model evaluations, respectively. F is the norm of the difference between the target and the design shapes.

	Case 1 (Drag minimization)			Case 2 (Inverse design)		
Variable	Initial	Pattern-search	SBO	Initial	Pattern-search	SBO
a	15.0000	21.8611	20.9945	18.000	24.7407	24.7667
x_n	5.0000	5.6758	5.6676	7.0000	7.3704	6.8333
y_n	5.0000	2.7022	2.7531	8.0000	4.7407	4.5667
x_t	90.0000	98.000	96.6701	85.0000	88.1111	88.6333
y_t	5.0000	0.8214	3.0290	7.0000	5.5926	5.3000
F	-	-	-	0.0204	1.64E-5	1.93E-5
C_D	0.0915	0.0853	0.0857	0.0925	0.0894	0.0893
N_c	-	0	300	-	0	500
N_f	-	282	3	-	401	5
Total cost	-	282	13	-	401	< 22

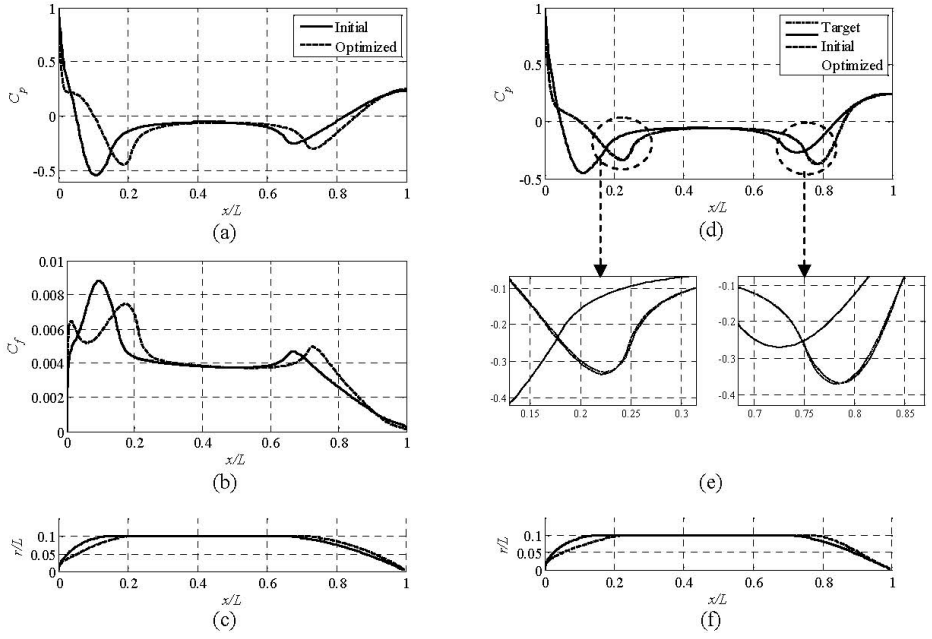


Fig. 10. Case 1 results for the direct hull drag minimization, showing the initial and optimized: (a) pressure distributions (b) skin friction distributions and (c) hull shapes; Case 2 results of the inverse design optimization with a prescribed target pressure distribution: (d) target, initial, and optimized pressure distributions, (e) zoom on two regions of Fig. 15(d), and (f) the initial and optimized hull shapes

6 Conclusions

Computationally efficient simulation-driven multi-fidelity design optimization algorithm for axisymmetric hulls in incompressible fluid flow is discussed. Our algorithm exploits a low-fidelity model, obtained through a coarse-discretization CFD simulation, and a response correction method, to construct a cheap and reliable surrogate of the fluid flow. The algorithm can be applied to both direct and inverse design approaches. We demonstrate that the optimized designs can be obtained at a low computational cost corresponding to a few high-fidelity CFD simulations.

References

1. Yamamoto, I.: Research and development of past, present, and future AUV technologies. In: Proc. Int. Mater-class AUV Technol. Polar Sci. – Soc. Underwater Technol., March 28-29, pp. 17–26 (2007)
2. Allen, B., Vorus, W.S., Prestero, T.: Propulsion system performance enhancements on REMUS AUVs. In: Proceedings MTS/IEEE Oceans 2000, Providence, Rhode Island (2000)
3. Huang, T.T., Liu, H.L., Groves, N.C., Forlini, T.J., Blanton, J.N., Gowing, S.: Measurements of flows over an axisymmetric body with various appendages (DARPA SUBOFF Experiments). In: Proceedings of the 19th Symposium on Naval Hydrodynamics, Seoul, Korea (1992)
4. Yang, C., Löhener, R.: Prediction of flows over an axisymmetric body with appendages. In: The 8th International Conference on Numerical Ship Hydrodynamics, Busan, Korea, September 22-25 (2003)
5. de Barros, E.A., Dantas, J.L.D., Pascoal, A.M., de Sá, E.: Investigation of normal force and moment coefficients for an auv at nonlinear angle of attack and sideslip angle. IEEE Journal of Oceanic Engineering 33(4), 538–549 (2008)
6. Jagadeesh, P., Murali, K., Idichandy, V.G.: Experimental investigation of hydrodynamic force coefficients over auv hull form. Ocean Engineering 36, 113–118 (2009)
7. Goldschmied, F.R.: Integrated hull design, boundary-layer control, and propulsion of submerged bodies. J. Hydronautics 1(1), 2–11 (1966)
8. Parsons, J.S., Goodson, R.E., Goldschmied, F.R.: Shaping of axisymmetric bodies for minimum drag in incompressible flow. J. Hydronautics 8(3), 100–107 (1974)
9. Myring, D.F.: A theoretical study of body drag in subcritical axisymmetric flow. Aeronautical Quarterly 28, 186–194 (1976)
10. Dalton, C., Zedan, M.F.: Design of low-drag axisymmetric shapes by the inverse method. J. Hydronautics 15(1), 48–54 (1980)
11. Lutz, T., Wagner, S.: Numerical shape optimization of natural laminar flow bodies. In: Proceedings of the 21st ICAS Congress, Melbourne, Australia, September 13-18 (1998)
12. Alvarez, A., Bertram, V., Gualdesi, L.: Hull hydrodynamic optimization of autonomous underwater vehicles operating at snorkelling depth. Ocean Engineering 36, 105–112 (2009)
13. Solov'ev, S.A.: Determining the shape of an axisymmetric body in a viscous incompressible flow on the basis of the pressure distribution on the body surface. J. of Applied Mechanics and Technical Physics 50(6), 927–935 (2009)
14. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidynathan, R., Tucker, P.K.: Surrogate-Based Analysis and Optimization. Progress in Aerospace Sciences 41(1), 1–28 (2005)

15. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Progress in Aerospace Sciences 45(1-3), 50–79 (2009)
16. Leifsson, L., Koziel, S.: Multi-fidelity design optimization of transonic airfoils using physics-based surrogate modeling and shape-preserving response prediction. Journal of Computational Science 1(2), 98–106 (2010)
17. Alexandrov, N.M., Lewis, R.M., Gumbert, C.R., Green, L.L., Newman, P.A.: Optimization with variable-fidelity models applied to wing design. In: 38th Aerospace Sciences Meeting & Exhibit., Reno, NV, AIAA Paper 2000-0841 (2000)
18. Braembussche, R.A.: Numerical Optimization for Advanced Turbomachinery Design. In: Thevenin, D., Janiga, G. (eds.) Optimization and Computational Fluid Dynamics, pp. 147–189. Springer (2008)
19. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Søndergaard, J.: Space Mapping: The State of the Art. IEEE Trans. Microwave Theory Tech. 52(1), 337–361 (2004)
20. Dulikravich, G.S.: Aerodynamic shape design and optimization. In: 29th AIAA Aerospace Sciences Meeting, Reno, NV (1991)
21. Leoviriyakit, K., Kim, S., Jameson, A.: Viscous Aerodynamic Shape Optimization of Wings including Planform Variables. In: 21st Applied Aerodynamics Conference, Orlando, Florida, June 23-26 (2003)
22. Koziel, S., Cheng, Q.S., Bandler, J.W.: Space mapping. IEEE Microwave Magazine 9(6), 105–122 (2008)
23. Lepine, J., Guibault, F., Trepanier, J.-Y., Pepin, F.: Optimized nonuniform rational b-spline geometrical representation for aerodynamic design of wings. AIAA Journal 39(11), 2033–2041 (2001)
24. Tannehill, J.A., Anderson, D.A., Pletcher, R.H.: Computational fluid mechanics and heat transfer, 2nd edn. Taylor & Francis (1997)
25. FLUENT, ver. 12.1, ANSYS Inc., Southpointe, 275 Technology Drive, Canonsburg, PA 15317 (2006)
26. ICEM CFD, ver. 12.1, ANSYS Inc., Southpointe, 275 Technology Drive, Canonsburg, PA 15317 (2006)
27. Koziel, S., Bandler, J.W., Madsen, K.: A Space Mapping Framework for Engineering Optimization: Theory and Implementation. IEEE Trans. Microwave Theory Tech. 54(10), 3721–3730 (2006)
28. Simpson, T.W., Peplinski, J., Koch, P.N., Allen, J.K.: Metamodels for computer-based engineering design: survey and recommendations. Engineering with Computers 17, 129–150 (2001)
29. Koziel, S., Bandler, J.W.: Recent advances in space-mapping-based modeling of microwave devices. International Journal of Numerical Modelling 23(6), 425–446 (2010)
30. Søndergaard, J.: Optimization using surrogate models – by the space mapping technique. Ph.D. Thesis, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby (2003)
31. Echeverría, D., Hemker, P.W.: Manifold mapping: a two-level optimization technique. Computing and Visualization in Science 11, 193–206 (2008)
32. Koziel, S.: Shape-preserving response prediction for microwave design optimization. IEEE Trans. Microwave Theory and Tech. 58(11), 2829–2837 (2010)
33. Alexandrov, N.M., Lewis, R.M.: An overview of first-order model management for engineering optimization. Optimization and Engineering 2, 413–430 (2001)
34. Koziel, S.: Multi-fidelity multi-grid design optimization of planar microwave structures with Sonnet. In: International Review of Progress in Applied Computational Electromagnetics, Tampere, Finland, April 26-29, pp. 719–724 (2010)

Analysis of Bulky Crash Simulation Results: Deterministic and Stochastic Aspects

Tanja Clees, Igor Nikitin, Lialia Nikitina, and Clemens-August Thole

Fraunhofer Institute for Algorithms and Scientific Computing, Schloss Birlinghoven
53754 Sankt Augustin, Germany
{Tanja.Clees,Igor.Nikitin,Lialia.Nikitina,
Clemens-August.Thole}@scai.fraunhofer.de

Abstract. Crash simulation results show both deterministic and stochastic behavior. For optimization in automotive design it is very important to distinguish between effects caused by variation of simulation parameters and effects triggered, for example, by buckling phenomena. We propose novel methods for the exploration of a simulation database featuring non-linear multidimensional interpolation, tolerance prediction, sensitivity analysis, robust multiobjective optimization as well as reliability and causal analysis. The methods are highly optimized for handling bulky data produced by modern crash simulators. The efficiency of these methods is demonstrated for industrially relevant benchmark cases.

Keywords: Surrogate Models, Bulky Data, Multiobjective Optimization, Stochastic Analysis.

1 Introduction

Simulation is an integral component of virtual product development today. The task of simulation consists mainly in solution of physical models in the form of ordinary or partial differential equations. From the viewpoint of product development the real purpose is product optimization, and the simulation is "only" means for the purpose. Optimization means searching for the best possible product with respect to multiple objectives (multiobjective optimization), e.g. total weight, fuel consumption and production costs, while the simulation provides an evaluation of objectives for a particular sample of a virtual product.

The optimization process usually requires a number of simulation runs, the results form a simulation dataset. To keep simulation time as short as possible, "Design of Experiments" (DoE, [1]) is applied, where a space of design variables is sampled by a limited number of simulations. On the basis of these samples, a surrogate model is constructed, e.g. a response surface [2], which describes the dependence between design variables and design objectives. Modern surrogate models [3, 4, 12-15] describe not only the value of a design objective but also its tolerance limits, which allow to control precision of the result. Moreover, not only scalar design objectives but whole simulation results, even highly resolved in space/time, (”bulky” dataset) can be modeled [12-15].

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 225–237.
DOI: 10.1007/978-3-642-34336-0_15

In this paper we will concentrate on the stochastic aspects of simulation processes. Industrial simulations, e.g. virtual crash tests, often possess a random component, related to physical and numerical instabilities of the underlying simulation model and uncertainties of its control parameters. Under these conditions the user is interested not only in the mean value of an optimization criterion, e.g. crash intrusion, but also in its scatter over simulations. In practice, it is required to fulfil optimization objectives with a certain confidence, e.g. 6-sigma. This task belongs to the scope of robustness or reliability analysis.

Often, the confidence intervals are so large that one has to reduce scatter before optimization. There is a part of scatter deterministically related to the variation of design variables, which can be found by means of sensitivity analysis. The other part is purely stochastic. It can be triggered by microscopic variations of design variables and - even if they are fixed - by the numerical process itself, e.g. by random scheduling in multiprocessing simulation. These microscopic sources are then amplified by inherent physical instabilities of the model related e.g. to buckling, contact phenomena or material failure. Stochastic analysis allows to track the sources of scatter, to reconstruct causal chains and to identify hidden parameters describing chaotic behavior of the model. If uncontrolled, these parameters propagate scatter to the optimization objectives. The challenge is to put them under control, at least partially, e.g. by predeformation of buckling parts, adjustment of contact conditions, placement of additional welding points etc. In this way the scatter of simulation can be suppressed and optimization becomes more robust.

In Sec.2 we will overview the methods for metamodeling of bulky simulation results; in Sec.3 we describe stochastic methods for reliability and causal analysis; Sec.4 presents applications of these methods to real-life examples in automotive design. The approaches presented in this paper have been implemented in software tools DiffCrash [9-11] and DesParO [12-15] and have been subjects of international patent applications (DPMA 10 2009 057295.3 and PCT/ EP2010/061439).

2 RBF Metamodel

Numerical simulations define a mapping $y=f(x)$: $R^n \rightarrow R^m$ from n-dimensional space of simulation parameters to m-dimensional space of simulation results. In crash test simulation the dimensionality of simulation parameters x is moderate (n~10-30), while simulation results y are dynamical fields sampled on a large grid, typically containing millions of nodes and hundreds of time steps, resulting in values of $m \sim 10^8$. High computational complexity of crash test models restricts the number of simulations available for analysis (typically $Nexp<10^3$) which is preferred to be as small as possible.

Metamodeling with radial basis functions (RBF) is a representation of the form

$$f(x)=\sum_{i=1..\,Nexp} c_i \, \Phi(|x-x_i|), \qquad (1)$$

where $\Phi()$ are special functions, depending only on the Euclidean distance between the points x and x_i. The coefficients c_i can be obtained by solving a linear system

$$y_i=\sum_j c_j \Phi(|x_i-x_j|), \quad (2)$$

where $y_i=f(x_i)$. The solution can be found by direct inversion of the moderately sized Nexp*Nexp system matrix $\Phi_{ij}=\Phi(|x_i-x_j|)$. The result can be written in a form of weighted sum $f(x)=\sum_i w_i(x)y_i$, with the weights

$$w_i(x)=\sum_j {\Phi^{-1}}_{ij}\Phi(|x-x_j|). \quad (3)$$

A suitable choice for the RBF, providing non-degeneracy of Φ-matrix for all finite datasets of distinct points and all dimensions n, is the multi-quadric function [5] $\Phi(r)=(b^2+r^2)^{1/2}$, where b is a constant defining smoothness of the function near data point $x=x_i$. RBF interpolation can also be combined with polynomial detrending, adding a polynomial part p():

$$f(x)=\sum_{i=1..Nexp} c_i \Phi(|x-x_i|)+p(x). \quad (4)$$

This allows reconstructing exactly polynomial (including linear) dependencies and generally improving precision of interpolation. The precision can be controlled via the following cross-validation procedure: the data point is removed, data are interpolated to this point and compared with the actual value at this point. For an RBF metamodel this procedure leads to a direct formula [13-15]

$$err_i = f_{interpol}(x_i)-f_{actual}(x_i) = -c_i/(\Phi^{-1})_{ii}. \quad (5)$$

Specifics of Bulky Data: although RBF metamodel is directly applicable for interpolation of multidimensional data, it contains one matrix-vector multiplication f(x)=yw(x), comprising O(mNexp) floating point operations per every interpolation. Here y_{ij}, i=1..m, j=1..Nexp is the data matrix, where every column forms one experiment, every row forms a data item varied in experiments.

Dimensional reduction technique applicable for acceleration of this computation is provided by principal component analysis (PCA). At first, an average value is row-wise subtracted, forming centered data matrix $dy_{ij} = y_{ij} - \langle y_i \rangle$. For this matrix a singular value decomposition (SVD) is written: $dy=U\Lambda V^T$, where Λ is a diagonal matrix of size Nexp*Nexp, U is a column-orthogonal matrix of size m*Nexp, V an orthogonal square matrix of size Nexp*Nexp:

$$U^TU=1,\ V^TV=VV^T=1. \quad (6)$$

A computationally efficient method [14] for this decomposition in the case m>>Nexp is to find Gram matrix $G=dy^Tdy$, to perform its spectral decomposition $G=V\Lambda^2V^T$, and to compute the remaining U-matrix with post-multiplication $U=dyV\Lambda^{-1}$. The Λ values are non-negative and sorted in non-ascending order. If all these values in the range k>Nmod are omitted (set to zero), the resulting reconstruction of y-matrix will have a deviation δy. L_2-norm of this deviation gives

$$err^2 = \sum_{ij} {\delta y_{ij}}^2 = \sum_{k>Nmod} {\Lambda_k}^2 \quad (7)$$

(Parseval's criterion). This formula allows controlling precision of reconstructed y-matrix. Usually Λ_k rapidly decreases with k, and a few first Λ values give sufficient

precision. The result of interpolation is represented as a product df=Ψg of SVD modes Ψ=UΛ (principal components) to SVD-transformed RBF weights g=V^Tw. Finally one has f(x)=<y>+df(x), computational cost of interpolation is reduced to O(m Nmod), plus once-charged O(m $Nexp^2$) cost of SVD. This method is convenient when interpolation should be performed many times (>>Nexp), e.g. for interactive exploration of database.

More generally, for representation of bulky data one can use clustering techniques [14]. They also decompose bulky data over a few basis vectors (modes) and accelerate linear algebra operations with them.

3 Reliability Analysis

The purpose of reliability analysis is an estimation of confidence limits (CL) for simulation results: P(y<CL)=C, where P is probability measure and C is a user specified confidence level. For example, median corresponds to 50% CL, i.e. P(y<med)=0.5; while 68% CL corresponds to confidence interval [CLmin,CLmax), where P(y<CLmin)=0.16, P(y≥CLmax)=1-P(y<CLmax)=0.16; etc. Several methods for solution of this task are available.

3.1 First Order Reliability Method (FORM)

FORM is applicable for linear mapping f(x) and normal distribution of simulation parameters:

$$\rho(x)\sim\exp(-(x-x_0)^T cov_x^{-1}(x-x_0)/2). \tag{8}$$

Here x_0=<x> is mean value of x and

$$(cov_x)_{ij} = <(x-x_0)_i(x-x_0)_j> \tag{9}$$

is covariance matrix of x. In this case y is also normally distributed, with mean value

$$y_0 = <y> = med(y) = f(x_0) \tag{10}$$

and covariance matrix

$$cov_y = J\, cov_x\, J^T, \tag{11}$$

where $J_{ij} = \partial f_i / \partial x_j$ is Jacoby matrix of f(x), called also sensitivity matrix. The diagonal part of cov_y gives standard deviations σ_y^2 directly defining CL(y), e.g.

$$CL_{min/max}(68\%) = <y> \pm \sigma_y, \tag{12}$$

$$CL_{min/max}(99.7\%) = <y> \pm 3\sigma_y. \tag{13}$$

In particular, when simulation parameters are independent random values, cov_x becomes diagonal: $cov_x = diag(\sigma_x^2)$, and

$$\sigma_{yi}^2 = \sum_{i=1..n} (\partial f_i / \partial x_i)^2 \sigma_{xi}^2. \tag{14}$$

A finite difference scheme used to compute Jacoby matrix of f(x) requires Nexp=O(n) simulations, e.g. 2n for central difference scheme plus one experiment at x_0, Nexp=2n+1. The algorithm possesses computational complexity O(nm) and can be implemented efficiently as reading data from Nexp simultaneously open data streams and writing CL to a single output data stream. In this way the memory requirements can be minimized and parallelization can be done straightforwardly.

3.2 Second Order Reliability Method (SORM)

SORM is applicable for slightly non-linear mapping f(x), which can be approximated by quadratic functions. The distributions ρ(x) are normal or can be cast to normal ones by a suitable transformation of parameter. For quadratic approximations CL can be explicitly computed [6] using main curvatures in the space of normalized variables $z_i=(x-x_0)_i/\sigma_{xi}$, i.e. eigenvalues of Hesse matrix $H^i_{jk}=\partial^2 f_i/\partial z_j \partial z_k$. These eigenvalues can be also used to estimate non-linearity of the mapping f(z), by maximizing the 1st and 2nd Taylor's terms over a ball of radius R:

$$\max_{|z|\le R} Jz=|J|R, \quad \max_{|z|\le R} |z^T Hz/2|=Hmax\ R^2/2, \tag{15}$$

so that the linear term prevails over quadratic one, in this ball, iff |J|>>Hmax R/2. Here

$$J_{ij}=\partial f_i/\partial z_j,\ |J|=(\sum_j J_{ij}^2)^{1/2} \tag{16}$$

and Hmax is maximal absolute eigenvalue of H. Both this criterion and estimation of main curvatures require full Hesse matrix, i.e. Nexp=O(n^2) simulations. Practically, the usability of SORM is limited, because strongly non-linear functions would involve higher order terms and because distributions of simulation parameters can strongly deviate from normal ones.

3.3 Confidence Limits Determination with Monte Carlo Method (CL-MC)

In the case of non-linear mapping f(x) and arbitrary distribution ρ(x) general Monte Carlo method is applicable. The method is based on estimation of probability

$$P_N(y<CL) = \text{num.of } (y_n<CL)/N \tag{17}$$

for a finite sample $\{y_1,...,y_N\}$. By the law of large numbers [7], $F_N=P_N(y<CL)$ is consistent unbiased estimator for F=P(y<CL), i.e. $F_N \to F$ with probability 1, when $N\to\infty$ and $\langle F_N\rangle = F$ for all finite N. By the central limit theorem [7], the error of such estimation $err_N=F_N-F$ at large N is distributed normally with zero mean and standard deviation $\sigma\sim(F(1-F)/N)^{1/2}$. Algorithmically the method consists of three phases:

(CL1) generation of N random points in parameter space according to user specified distribution ρ(x),
(CL2) numerical simulations for given parameter values,
(CL3) determination of confidence limits by one-pass reading of simulation results, sorting m samples $\{y_1,...,y_N\}$ and selection of k-th item in every sample with k=[(N-1)F+1] as a representative for CL.

The analysis phase of the algorithm possesses computational complexity O(mN log N) and can be efficiently implemented using data stream operations similar to FORM. Precision of the method is estimated using standard deviation formula above. Remarkably, the precision depends neither on dimension of parameter space n, nor on the length of simulation result m, but only on sample size N=Nexp and user-specified confidence level F=C. For instance, CL determination at the level 68% (F=0.16) with 4% precision requires Nexp=84, while for 68%±1% one needs Nexp=1344.

3.4 Monte Carlo Combined with RBF Metamodel (MC-RBF)

Large sample size is required for precise determination of CL with Monte Carlo method. To reduce the number of required simulations, RBF metamodel can represent the mapping f(x) during analysis phase of CL-MC. While a metamodel can be constructed using a moderate number of simulations, e.g. Nexp~100, determination of CL can be done with N>>Nexp. Application of RBF metamodel for CL computation proceeds similarly to CL-MC. The only difference is that Nexp parameter points generated at phase (CL1) are used as input for the metamodel. They should not necessarily possess user specified distribution, but one providing better precision of metamodel, i.e. better covering "the corners" of parameter space. It is especially important for populating tails of distribution, corresponding to high confidence e.g. 99.7% CL. Uniform distribution is suitable for this purpose. Then, after numerical simulations at phase (CL2), and after filtering out failed experiments, the actual distribution $\rho(x)$ is used to generate N parameter points, and construct RBF weight matrix $w_{ij}=w_i(x_j)$, i=1..Nexp, j=1..N. This matrix is used in phase (CL3) for multiplication with simulation results y_{ik}, k=1..m, comprising O(m N Nexp) operations, which usually prevails over O(m N log N) operations needed for sorting of interpolated samples.

4 Causal Analysis

Causal analysis is determination of cause-effect relationships between events. In context of crash test analysis, this usually means identification of events or properties causing the scatter of the results. This allows to find sources of physical or numerical instabilities of the system and helps to reduce or completely eliminate them.

Causal analysis is generally performed by means of statistical methods, particularly, by estimation of correlation of events. It is commonly known that correlation does not imply causation (this logical error is often referred as "cum hoc ergo propter hoc": "with this, therefore because of this"). Instead, strong correlation of two events does mean that they belong to the same causal chain. Two strongly correlated events either have direct causal relation or they have a common cause, i.e. a third event in the past, triggering these two ones. This common cause will be revealed, if the whole causal chain i.e. a complete sequence of causally related events will be reconstructed. Practical application of causal analysis requires formal methods for reconstruction of causal chains.

A practical problem of causal analysis in crash-test simulations is often not a removal of a prime cause of scatter, which is the crash event itself. It is more an observation of propagation paths of the scatter, with a purpose to prevent this propagation, by finding regions where scatter is amplified (e.g. break of a welding point, pillar buckling, slipping of two contact surfaces etc). Since a small cause can have large effect, formally earliest events in the causal chain can have a microscopic amplitude ("butterfly effect"). Therefore it is reasonable to search for amplifying factors and try to eliminate them, not the microscopic sources.

As input for causal analysis the centered data matrix dy_{ij}, i=1..m, j=1..Nexp is used. Here every column forms one experiment, every row forms a data item varied in experiments, and the mean value <y> is row-wise subtracted from the matrix. Then every data item is transformed to a z-score vector [8]:

$$z_{ij}=dy_{ij}/|dy_i|,\ |dy_i|=\mathrm{sqrt}(\sum_j dy_{ij}^2), \tag{18}$$

or by means of the equivalent alternative formula

$$z_{ij}=dy_{ij}/(s(y_i)(N_{exp})^{1/2}),\ s(y_i)=(\sum_j dy_{ij}^2/N_{exp})^{1/2}. \tag{19}$$

Here $s(y_i)$ is the root mean square deviation of the i-th data item, which can serve as a measure of scatter. In this way the data items are transformed to m vectors in Nexp-dimensional space. All these z-vectors belong to an (Nexp-2)-dimensional unit-norm sphere, formed by intersection of a sphere |z|=1 with a hyperplane $\sum_j z_{ij}=0$. The scalar product of two z-vectors is equal to Pearson's correlator of data items:

$$(z_1,z_2) = \sum_j z_{1j} z_{2j} = \mathrm{corr}(y_1,y_2). \tag{20}$$

An important role of this representation is the following. Strongly correlated data items correspond either to coincident ($z_1=z_2$) or opposite ($z_1=-z_2$) z-vectors. If not the sign but only the fact of dependence is of interest, one can glue opposite points together formally considering a sphere of z-vectors as projective space. Using this representation, one can apply [13,14] general purpose clustering methods such as k-means to group data items distributed on this sphere to a few strongly correlated components.

In spite of their numerical efficiency, these clustering methods neglect temporal ordering of events, while in causal analysis the task is to find an earliest physically significant event in the causal chain. In crash test simulation such events correspond to bifurcation points, where the scatter appears "ex nihilo". Such points are clearly visible as spikes in dynamical scatter plots s(y), the problem is that there are too many of them. Although decision between potential candidates by a formal algorithm can be difficult, an engineering knowledge allows narrowing the search to significant parts where scatter propagation can be really initiated by physical effects, such as buckling of longitudinal, break of welding point etc. The other problem is that in bifurcation points new scatter is just appeared and it is generally hidden under the consequences of previous effects. At first one needs to separate scatter contributions.

Considering two data items dy(a) and dy(b), one can define contribution relevant to the data item (a) in (b) as follows:

$$dy|_a(b) = corr(a,b)s(b)z(a). \tag{21}$$

After subtraction of this contribution a residual $dy(b)-dy|_a(b)$ does not correlate with $dy(a)$, and the scatter

$$s^2|_a(b)=<(dy(b)-dy|_a(b))^2>=s^2(b)(1-corr^2(a,b)) \tag{22}$$

does not increase.

Armed with this subtraction procedure, we propose the following algorithm for causal analysis.

Temporal clustering:
(T1) visualize scatter state-by-state;
(T2) isolate bifurcation point;
(T3) subtract its contribution from the scatter in consequent states;
(T4) if scatter is still remaining, goto (T1).

Here subtraction of scatter from previous bifurcations reveals new bifurcations hidden under the consequences of previous ones. The remaining scatter monotonously falls during the iterations, and the iterations can be stopped when the scatter becomes small everywhere or in the regions of interest.

The geometrical meaning of subtraction procedure: $b-b|_a=b-(a,b)/(a,a)a$ is an orthogonal projection in the space of data items and the whole sequence is Gram-Schmidt (GS) orthonormalization procedure applied in the order of appearance of bifurcation points a_i. The obtained orthonormal basis $g_i=GS(a_i)$ can be used for reconstruction of all data by the formula:

$$dy=\sum_i \Psi_i g_i+res,\ \Psi_i=(dyg_i). \tag{23}$$

The norm of residual is controlled by remaining scatter, which is small according to our stop criterion:

$$|res|^2/Nexp=s_\perp^2(y)=s^2(y)-\sum_i \Psi_i^2/Nexp. \tag{24}$$

Algorithmically every i-th iteration one computes a scalar field Ψ_i describing contribution of i-th bifurcation point to scatter of the model and a scalar field $s_{i\perp}^2(y)$ used for determination of the next bifurcation point a_{i+1}, or for stop criterion $s_{i\perp}^2(y)<threshold$. This requires O(mNexp) floating point operations per iteration.

Matrix decomposition of the form $dy=\Psi g$ is similar to PCA described above, with the other meaning of the modes Ψ. Like in PCA, Ψ are scalar fields distributed over dynamical model which are common for all experiments. They have the other temporal profile than PCA modes, reflecting causal structure of scatter: they start at corresponding bifurcation points and propagate forward in time. Differently from PCA modes, they are not orthogonal columnwise, i.e. with respect to scalar product over the model. g-coefficients form Nexp*Nmod columnwise orthonormal matrix. Like corresponding matrix in PCA, they define an orthonormal basis in the space of experiments, with respect to the scalar product coincident with Pearson's correlator.

The scatter associated with design variables can be treated by the same method, if one puts data items containing variation of design variables as the first candidates for bifurcation points. The corresponding Ψ-modes will represent sensitivities of simulation results to variation of parameters. The remaining scatter represents indeterministic part of the dependence. The corresponding Ψ-modes are bifurcation profiles and their g-coefficients are those hidden variables which govern purely stochastic behavior of the model. One can either take hidden variables into account when performing reliability analysis, or try to put them under control for reducing scatter of the model.

5 Examples

5.1 Audi B-Pillar Crash Test

The model shown on Fig.1 contains 10 thousand nodes, 45 timesteps, 101 simulations. Two parameters are varied representing thicknesses of two layers composing a part of a B-pillar. The purpose is to find a Pareto-optimal combination of parameters simultaneously minimizing the total mass of the part and crash intrusion in the contact area. To solve this problem, we have applied the methods described in Sec.2, namely RBF metamodeling of target criteria for multiobjective optimization and PCA for compact representation of bulky data. Based on these methods, our interactive optimization tool DesParO supports real-time interpolation of bulky data, with response times in the range of milliseconds. As a result, the user can interactively change parameter values and immediately see variations of complete simulation result, even on an ordinary laptop computer.

In more details, Fig.1 shows the optimization problem loaded in the Metamodel Explorer, where design variables (thicknesses1, 2) are presented at the left and design objectives (intrusion and mass) at the right. First, the user imposes constraints on design objectives, trying to minimize intrusion and mass simultaneously, as indicated by red ovals on Fig.1 (upper part). As a result, “islands” of available solutions become visible along the axes of design variables. Exploration of these islands by moving corresponding sliders shows that there are two optimal configurations, related cross-like, as indicated on Fig.1 (middle). For these configurations, both constraints on mass and intrusion are satisfied, while they correspond to physically different solutions, distinguished by an auxiliary velocity criterion. For every criterion also its tolerance is shown corresponding to 1-sigma confidence limits, as indicated by horizontal bars under the corresponding slider as well as +/- errors in the value box. This indication allows satisfying constraints with 3-sigma (99.7%) confidence, as shown on the images. The Geometry Viewer, shown at the bottom of Fig.1, allows to inspect the optimal design in full details.

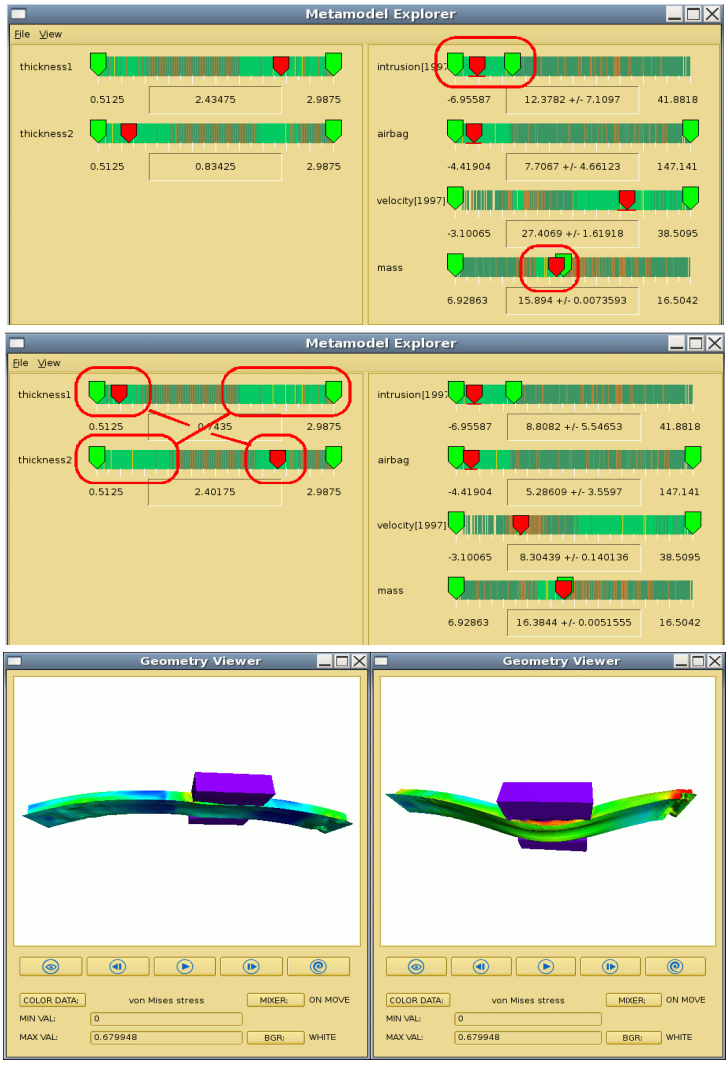


Fig. 1. Audi B-Pillar crash test in DesParO Metamodel Explorer. *(top):* constraints on intrusion and mass are imposed. *(center):* two optimal designs are found. *(bottom)*: inspection of optimal design in DesParO Geometry Viewer.

E.g. on the two images at the bottom, one can see the difference between small and large thickness values resulting in softer or stiffer crash behavior.

While performing constraint optimization, the user immediately sees how small mass solutions disappear when intrusion is minimized. This gives an intuitive feeling for the trade-off (Pareto behavior) between optimization objectives. With these capabilities and complementary information such as auxiliary criteria and interactive interpolation of bulky simulation results, “the” optimal solution, i.e. a single representative on the Pareto front, can be selected by a user decision.

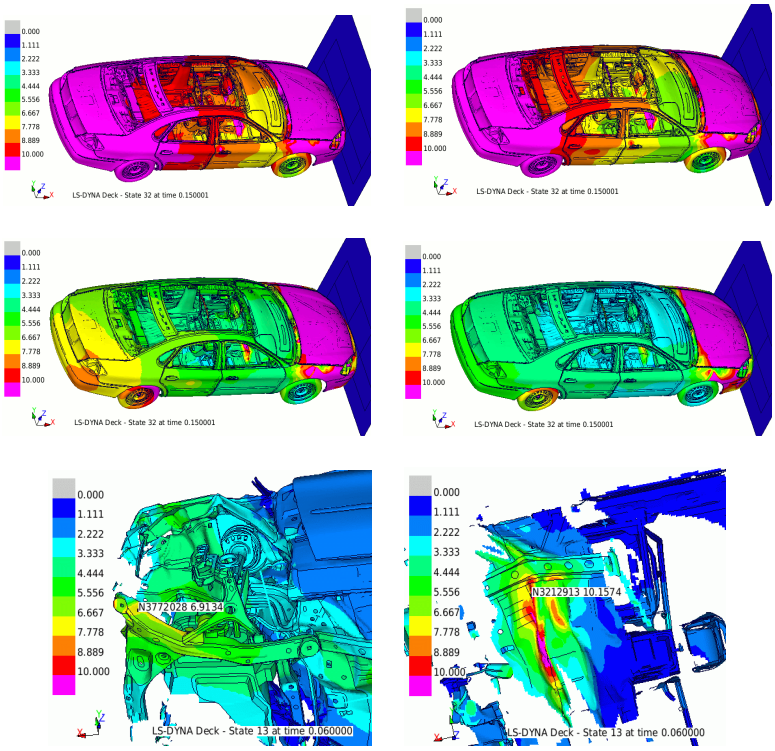


Fig. 2. Temporal clustering of Ford Taurus crash test using DiffCrash. (*upper left):* original scatter in mm. (*upper right – center left – center right):* consequent iterations of scatter subtraction. *(bottom):* two major bifurcations found.

5.2 Ford Taurus Crash Test

The crash model shown on Fig.2 contains 1 million nodes, 32 timesteps, 25 simulations. Processing of this model with the temporal clustering algorithm described above has been performed on a 16-CPU Intel-Xeon 2.9GHz workstation with 24GB main memory. It required 3min per iteration and converged in 4 iterations.

Crash intrusions in the foot room of the driver and passenger are commonly considered as critical safety characteristics of car design. These characteristics possess numerical uncertainties, the analysis of which falls in the subject of Sec.3-4. The upper left part of Fig.2 shows the scatter measure s(y), in mm, distributed on the model. The scatter in the foot room is so large (>10mm) that direct minimization of intrusion is impossible. Temporal clustering allows to identify sources of this scatter and to subtract relevant contributions. Further images show how the scatter decreases in these subtractions. After the 4th iteration the scatter in the foot room reaches a safe level (<3mm). Several bifurcation points have been identified and subtracted per iteration; in this way the performance of the algorithm has been optimized. The two major bifurcations found are shown on the bottom part of Fig.2. They represent

buckling phenomena on the left longitudinal rail and a fold on the floor of the vehicle, which appear in earlier time steps. In total, 15 bifurcation points have been identified, representing statistically independent sources of scatter. The whole scatter in the model can be decomposed over the corresponding basis functions $\Psi(y)$. In this way the dimensionality of the problem is reduced to 15 variables (g-coefficients) completely describing the stochastic behavior of the model.

6 Conclusions

We have presented and discussed methods for nonlinear metamodeling of a simulation database featuring continuous exploration of simulation results, tolerance prediction and rapid interpolation of bulky FEM data. For the purpose of robust optimization, the approach has been extended by the methods of reliability and causal analysis. The efficiency of the methods has been demonstrated for several application cases from automotive industry.

Further plans include to use the results of causal analysis as a basis for modifications of a simulation model for improving its stability. We also plan to consider non-linear relationships between stochastic variables. Linear methods such as PCA and GS determine only a linear span over principal components, while some stochastic variables can become non-linear functions of others. For determination of such dependencies the methods of curvilinear component analysis (CCA) can be applied.

Acknowledgements. Many thanks to the team of Prof. Steve Kan at the NCAC and the Ford company for providing the publically available LS-DYNA models and to Andreas Hoppe and Josef Reicheneder at AUDI for providing PamCrash B-pillar model. The availability of these models fosters method developments for crash simulation. We are also grateful to Michael Taeschner and Georg Dietrich Eichmueller (VW) and to Thomas Frank and Wolfgang Fassnacht (Daimler) for fruitful discussions.

References

1. Tukey, J.W.: Exploratory Data Analysis. Addison-Wesley, London (1997)
2. Donoho, D.L.: High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality, Lecture on August 8, 2000 to the American Mathematical Society "Math Challenges of the 21st Century" (2000), http://www-stat.stanford.edu/~donoho/Lectures/AMS2000/AMS2000.html
3. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient Global Optimization of Expensive Black-Box Functions. J. Glob. Opt. 13, 455–492 (1998)
4. Keane, A.J., Leary, S.J., Sobester, A.: On the Design of Optimization Strategies Based on Global Response Surface Approximation Models. J. Glob. Opt. 33, 31–59 (2005)
5. Buhmann, M.D.: Radial Basis Functions: Theory and Implementations. Cambridge University Press, Cambridge (2003)

6. Cizelj, L., Mavko, B., Riesch-Oppermann, H.: Application of First and Second Order Reliability Methods in the Safety Assessment of Cracked Steam Generator Tubing. Nucl. Eng. Des. 147, 359–368 (1994)
7. van der Vaart, A.W.: Asymptotic Statistics. Cambridge University Press, Cambridge (1998)
8. Larsen, R.J., Marx, M.L.: An Introduction to Mathematical Statistics and Its Applications. Prentice Hall, Upper Saddle River (2001)
9. Thole, C.-A., Mei, L.: Reason for Scatter in Simulation Results. In: 4th European LS-DYNA User Conference, vol. B-III, pp. 11–20. Dynamore Press, Ulm (2003)
10. Thole, C.-A.: Compression of LS-DYNA Simulation Results. In: 5th European LS-DYNA User Conference, vol. 6b, pp. 86–91. Dynamore Press, Birmingham (2005)
11. Thole, C.-A., Mei, L.: Data Analysis for Parallel Car-Crash Simulation Results and Model Optimization. Simulation Modelling in Practice and Theory 16(3), 329–337 (2008)
12. Stork, A., Thole, C.-A., Klimenko, S., Nikitin, I., Nikitina, L., Astakhov, Y.: Towards Interactive Simulation in Automotive Design. Vis. Comput. J. 24, 947–953 (2008)
13. Nikitin, I., Nikitina, L., Clees, T., Thole, C.-A.: Advanced Mode Analysis for Crash Simulation Results. In: 9th LS-DYNA User Forum, vol. SI1, pp. 11–20. Dynamore Press, Bamberg (2010)
14. Nikitin, I., Nikitina, L., Clees, T.: Stochastic analysis and nonlinear metamodeling of crash test simulations and their application in automotive design. In: Browning, J.E., McMann, A.K. (eds.) Computational Engineering: Design, Development and Applications. Nova Science, New York (2011) (to be publ.)
15. Nikitin, I., Nikitina, L., Clees, T.: Nonlinear metamodeling, multiobjective optimization and their application in automotive design. In: Günther, M., Bartel, A., Brunk, M., Schoeps, S., Striebel, M. (eds.) Progress in Industrial Mathematics at ECMI 2010, vol. 17. Springer (2010) (to be publ. 2012)

Integration of Optimization to the Design of Pulp and Paper Production Processes

Mika Strömman, Ilkka Seilonen, Jukka Peltola, and Kari Koskinen

School of Electrical Engineering, Aalto University, Aalto, Finland
{mika.stromman,ilkka.seilonen,
jukka.peltola,kari.o.koskinen}@tkk.fi

Abstract. Adopting new methodology in a design process requires changes in organization, business process, roles, knowledge, data transfer and tools. The influence of the new methodology has to be evaluated and the costs of changes calculated before the change is possible. In pulp and paper industry the non-growing market situation has tightened the competition that much that cutting the design costs by integrating design activities is not going to be enough. The design itself has to be improved. In this paper, an optimization method is integrated to an existing design process of pulp and paper facilities. The model of a new design process is then assessed through a case study and an interview study to ensure that the design process can be realized in the conceptual design phase of a real delivery project.

Keywords: Multidisciplinary Design, Process Engineering, Optimization.

1 Introduction

The non-growing market situation in pulp and paper industry is setting requirements for the design methods. The design process itself has to be conducted efficiently, but in the last years the costs has already been cut off with better project management and concurrent engineering. One possibility for rationalization lies in the design itself; traditionally, the design of the plant is more oriented into structural design and less to the optimal combination of operational and structural design. The design problem can be formulated as a bi-level multi-objective optimization problem (acronym: BLMOO). Mathematical methods for solving BLMOO problems exists and the method have been applied in process facility design in research projects.

However, the utilization of such optimization methods requires enhancement of the engineering process so that the required information for optimization is available on the right time and the results of optimization can be used in design. A design process describing optimizing design of continuous production processes hasn't been thus far presented and it is a necessity for adopting BLMOO-methods in real delivery projects.

This research has been conducted as a part of a larger research project in which the objective is to develop a new optimization based method for designing a process plant. Our part of the research is to define a model for optimizing design process and assess the usability of that model. The research methods of this study include

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 239–254.
DOI: 10.1007/978-3-642-34336-0_16

experimental definition of a business process model, case study (with the model) and interview study evaluating the properties of the model.

In the first chapter the related work and state of the art is discussed. The following chapter presents the new engineering business process which takes into account the optimizing method. Next, a case evaluating the new engineering business process is presented and the observations based on expert interviews are discussed.

2 Process Design and Optimization

2.1 Design of Continuous Production Systems

A process plant design is a multidisciplinary process (process design, automation, software etc.) [1]. Traditionally the process plant design process has been water fall model like linear process with stages ending to document deliveries. In a delivery project, the deadlines are counted backwards from the day that the plant should be operational. The length of work phases are determined based on time needed for work and procurement [2].

The plant engineering process can be divided into steps e.g. problem analysis, conceptual design, detailed engineering, and construction [3]. More business oriented divisions are also possible, for example conceptual phase, pre-feasibility study, feasibility study, investment decision and implementation [4]. Although all the phases are equally important for reaching the goal, the focus in this research is put on the conceptual design phase, because the optimization methods researched in this research project aim to solve problems on conceptual design level. Other phases of the engineering process are relevant to our research in that sense that the tools and methods should be compatible to the proposed changes.

In the conceptual design phase a very small amount of information is available and the time and resources are limited [5]. Still the decisions in this phase fix 80% of the total costs of the project [6]. Decisions in the early phases of the project are also quality-critical, because the costs of changes increase tenfold in each phase (research – process flow – final design – production) [7]. In process plant engineering, the conceptual design phase is led by process design. All the other engineering disciplines are more or less in consulting role. For these reasons, the greatest advantages can be achieved in early phases of the business process. Because of the shortened delivery times, the other engineering disciplines have to begin their work before the process design is ready.

The sub-processes of any process design task are design task definition, process structure design, process operation design an design acceptance. Process structure design and process control design interact and should therefore be designed simultaneously [8]. The existing process design approaches can be divided to heuristic and engineering experience based methods, optimization based methods and case-based reasoning methods [9]. Case based reasoning (CBR) has been applied for design of the pulp process. The main challenge in CBR is the need of extensive database to provide the required knowledge [8]. Outsourcing of the design work is a common practice nowadays. Fathianathan and Panchal [10] have proposed a model to support outsourcing decisions.

2.2 Optimization in Process Design

Current work practices in forest industry process engineering are almost solely based on engineering experience. Simulation and optimization is used in the design of unit processes, but less in the design of the process as whole. Plant wide simulation enables the validation of process structure and control concepts even before selecting suppliers and therefore it reduces risks [11] and gives a deeper understanding of the process [12]. According to the interviews, plant wide simulation is more useful when building a plant with totally new concepts when the "rules of thumb" are not available.

For combining the optimization of plant structure and plant control, there are several options. Optimization strategy can be sequential, iterative, bi-level or simultaneous. [13].

Bi-level optimization has been under an active research lately [14]. Still only a few research is dealing with multi-objective bilevel problems. Eichfelder [15] presents an algorithm for solving bilevel multi-objective problems. The combination of dynamic simulator model and dynamic optimization has been researched for papermaking process [16].

2.3 Information Systems for Process Design

The variety of the Computer Aided Engineering (CAE) tools supporting process systems engineering (PSE) is enormous. One of the interviewed engineering enterprises is using over 50 different engineering tools. A trend, as seen in modern integrated process engineering tools, is the transformation from document-centric design to data-centric design, realized with database technology [17-19].

Major tool vendors have developed, acquired and integrated engineering tools from other engineering disciplines under unified product families. Modern process engineering support systems combine modeling and information management features for engineering of many aspects of plant engineering, e.g. process, piping, electrical and instrumentation, 3D layout, equipment lists, part data sheets, etc, thus comprising an integrated plant information model. This also enables advanced change management, where modification of an object through one view notifies users of other views, looking at the same object. Multi site work flow management is featured for both engineering and commissioning. Integration to external CAE tools is possible through export and import interfaces using standard or proprietary data formats. An important prerequisite for cost efficient integrated engineering is the use of common data models defined in the standards. ISO 15926,"lifecycle data for process plant" is a standard dedicated to the process industry, widely accepted by tool vendors [20]. It has a central role in pursuing information interoperability between engineering systems and it is used in many plant information exchange tool initiatives, such as iRing [21] and XMpLant and even as a native data model of a plant modeling tool [19].

Plant information models and semantic technologies have induced much academic research. For example, POSC Caesar association [22] assembles R&D around the ISO 15926 and modeling methods, such as [23]. However, Wiesner, Morbach and Marquardt. [24] questions whether a single global plant information standard is a realistic goal in the first place and suggest a semantic integration framework OntoCAPE.

3 Model for Optimizing Design Process

Optimizing process design is here modelled in terms of a business process model. The model describes the stakeholders of the optimizing process design and their activities together with the data, knowledge and utilized mathematical models. Based on these the requirements for IT support are identified.

In this research, the workflow, roles, data and tools have been considered without organizational boundaries. In real life, these boundaries can have significant influence of the realization of this business process model. For example the enterprises can use different tools and all the data is not necessarily open for everyone. Also the business models for every stakeholder differ and therefore the different goals have to be taken into account in design process model.

3.1 Process Design as an Optimization Problem

The process design task can be considered as an optimization problem. There are a few general requirements for the process. The process must be operable, reliable and yield products of sufficient quality with minimum operational cost. On the other hand the investment and maintenance cost of the process should be minimized as well. On this basis it is natural to consider and model the design problem as a bi-level multi-objective optimization problem. The mathematical representation of the general bi-level multi-objective optimization problem is:

$$\begin{aligned} &\min_{(x_u, x_i)} F(x) = (F_1(x), \ldots F_M(x)), \\ &\textit{subject to} \quad x_i \in \arg\min_{(x_i)} \left\{ f_1(x), \ldots, f_{m(x))} \,\middle|\, g(x) \geq 0, h(x) = 0 \right\}, \\ &G(x) \geq 0, H(x) = 0, \\ &x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, \ldots, n. \end{aligned} \tag{1}$$

where

F(x) are the upper level objective functions,

f(x) the lower level objective functions,

G(x), **g**(x), **H**(x) and **h**(x) the upper and lower level inequality and equality constraints.[25]

There are multiple methods for solving bi-level multi-objective optimization (BLMOO) problems [15] and [26] and the solution method should be chosen according to the problem itself and the possibilities for interaction with the decision maker [27] In the plant design process, there is a logical division to optimization levels, so that plant structure is the upper level (**F**(x)) and the operation of the plant is the lower level (**G**(x)). The nature of the plant design is also multi-objective; the balancing between design parameters as for example the total cost of the plant, operational costs, production quality, production volume and expected oee-value is difficult and the decision of these values belongs to the plant owner, not the designer. Therefore the gathered requirements should also cover business oriented user preferences.

In this research the solution of the optimization problem was simplified by scalarizing the lower level optimization problem, but this simplification has no affect to this part of the research focusing on the business process of the design.

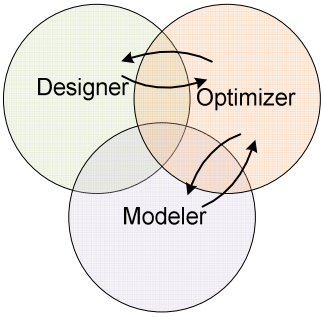


Fig. 1. Stakeholders of optimizing design

3.2 Stakeholders

In the model of optimizing design, new stakeholders, an optimizer and a modeler, are added to the group of stakeholders involved in process design as illustrated in Figure 1. The optimizer is an expert of mathematical optimization whose responsibility is to help the process designer in finding more optimal process designs. The optimizer also needs to cooperate with the modeler in order to be able to take into account the operational aspects of the designed process. These cooperation connections with the optimizer will also change the work of the process designer and modeler. Successful cooperation between the stakeholders is a necessity for useful design optimization.

The role of the optimizer can be described as an analyst [28]. His responsibility is not to make decisions about process designs but to produce useful information for the designer about possibly better designs. In order to do this, the optimizer will need to have expertise in multi-objective optimization and familiarity in process design. The adoption of optimization also changes the roles of pre-existing stakeholders. Designer is the decision-maker of the process design and the client of the optimizer. In optimizing design the designer has to select a part of his design problem for optimization together with the optimizer. In addition to this, the designer also has to cooperate with the optimizer during the optimization process and finally interpret the results and decide how to apply them. Again, the optimizer will become the client of the modeler.

In order to adopt the new business process, all the stakeholders should gain some advantage of the enhanced business process. The process designer gains competitive advantage by offering design that is more tailored and more cost effective along the life cycle of the plant. For the optimizer and modeler, the new model opens a totally new business possibility.

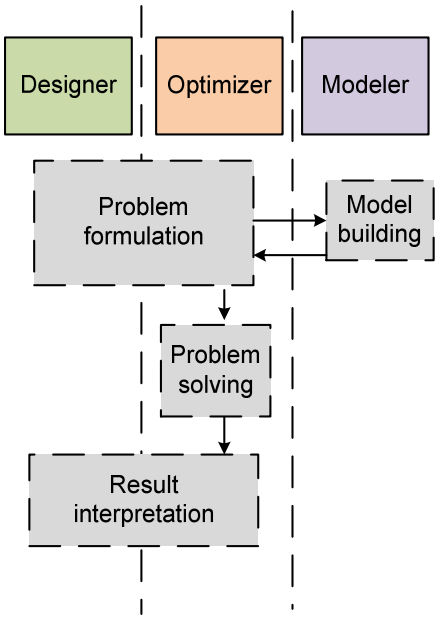


Fig. 2. Activities of optimizing design

3.3 Business Process

In the model of optimizing design the activities of process design have partially changed. The basis for the activities is the existing design processes that are extended and partly modified. The suitable time for optimization is the conceptual design phase. When the designer identifies a need for optimization in his conceptual design, he initiates cooperation with the optimizer. During this cooperation an optimal design balancing both structural and operational aspects of the design are being searched for. This process can be described as expert cooperation in which also the modeler will be included.

The optimization activities take place in a few stages as an extension to conceptual process design phase as illustrated in Figure 2. The process starts from optimization problem definition and continues through optimization problem-solving until result interpretation. During these stages different cooperation patterns between the designer and modeler are needed. The whole process and each of its stages may also be iterative.

The purpose of the optimization problem definition is to define a part of the designer's design problem as BLMOO for the optimizer. This stage is performed by the designer and optimizer together. The designer identifies parts of the overall design problem in which balancing structural and operational aspects of the design is essential. The solvability of the problem is then assessed by the optimizer, designer and modeler together. The assessment requires expertise of all three stakeholders because the result depends not only on the problem itself but e.g. optimization tools, process models and data about the process. Eventually the designer and the optimizer should

agree on a useful and solvable design optimization problem, which the optimizer then formulates as a BLMOO problem.

An important subtask of the optimization problem definition is process operation modeling. Modeling the operational part of the design problem is much more difficult than the structural part due to its dynamic and stochastic nature.of the modeler. In this subtask the designer and the optimizer can rely on the expertise of the modeller. The modeller is expected to have expertise about both mathematical modeling and the designed process itself, i.e. its chemical and physical characteristics. Based on his expertise the modeler should be able to create such operational models that are suitable to be used in optimization. The suitability of the models will be assessed by the optimizer and the designer.

The stage of problem-solving is focused on the optimizer. However, cooperation with the other stakeholders is likely to be needed also in this stage. In the beginning of this stage the data and models required in the optimization are expected to be transferred to the optimizer in a form which he can utilize. Depending on the utilized MOO method, different type and amount of cooperation with designer will be needed also during the actual problem-solving. According to an interview (see chapter "Interviews") industrial experts seem to favor optimization methods which lead to representations of Pareto optimal designs.

The last stage of optimizing design is result interpretation. Also this stage is performed in cooperation between the designer and the optimizer. The optimizer prepares result presentations, which indicate Pareto optimal designs and help the designer evaluate the impact of his preferences on the design. The designer is expected to study the design optimization result, assess its reliability and make decision about possible changes to his design. This is not necessary a straightforward task and is likely to require assistance from the optimizer and the modeler. The reliability of the optimization result is dependent on used operational models and data. Sensitivity analysis of the result might also be needed. In the end, the designer can adopt changes to his design or reject the optimization results and reformulate the optimization problem with the optimizer.

3.4 Data, Knowledge and Models

The optimizing design requires additional knowledge, data and models than the state-of-the-art approaches to process design. The new requirements originate from the need to solve the process design BLMOO problem. The new requirements for knowledge, data and models in optimizing design are summarized in Table 1. In addition to these, the previous requirements are still valid, e.g. designer knowledge for process design, use of design data and design models.

The expertise and knowledge of the stakeholders involved in optimizing design is complementary. The designer has knowledge about industrial processes and their design, customer requirements and evaluation of process designs. Meanwhile, the modeler is expected have knowledge about similar processes and their mathematical modeling.

The knowledge of the optimizer concerns about optimization and acting as an analyst in a decision-making process of MOO. However, during the activities of the optimizing design combination of the knowledge of different stakeholders and

Table 1. Knowledge, model and data requirements in optimizing design

	Knowledge	Data	Models
Designer	Process design, process knowledge, some understanding about optimization	Design data, customer requirements	Flow diagram P&ID Plant Model
Optimizer	Optimization, some understanding about design	Design data and operational data from designer and modeler	Operational and design level problem formulation models for optimization
Modeler	Modeling, Process knowledge	Operational data, some design data	Operational models (e.g break probability model,

knowledge transfer between them is necessary. A partially common understanding of the design problem shared by the stakeholders has to be created [29]. This is may be done according to the BLMOO of the process design.

Mathematical models of the designed process have an important role in optimizing design. Models are needed particularly for modeling the operation of the process. Mathematical models have been used in the design of continuous processes also previously, e.g. in simulations [11], but these models are not necessary suitable to be used in optimizing design. In order to be able to be utilized in optimizing design, the operational models need to have a suitable balance of modeling capability and computational requirements. The computational requirements can be met by modeling only selected parts of the process. More precise models may be utilized after the design in a design validation stage.

The optimizing design requires data transfer between the optimizer and other stakeholders, which is not needed without optimization. The most important data transfer takes place from the designer to the optimizer. The designer has to pass the most of the data describing the design optimization problem to the optimizer, e.g. flow diagrams, dimensions of equipments etc. The other source of data to the optimizer is the modeler. He is expected to deliver to the optimizer the operational models and the data required by them, e.g. model describing the probability of break. This data is intended for algorithmic processing, which indicates a requirement for adequate precision. The final data transfer consists of the optimization results, which are passed from the optimizer to the designer. This data has a form of a document. A major requirement for it is understandability.

3.5 Requirements for Information Systems

The new requirements for the information systems mainly rise from the new data flows between designer, optimizer and modeler. The amount of data from the designer's plant model is moderate, but the iterations may justify the usage of a proper tool for data exchange and version management. The optimizer should have access to the designers plant model tool to be able to import the needed set of design data. A new thing is that the designer should also include the constraints of the design to the model

when applicable. The design data should be transferable to optimizing tool as well as the models that the modeler has created. The support for representing the alternatives to the designer is not that critical, because that document should be kept brief and simple.

4 Assessment of the Model

In this chapter, the business process model of optimizing design is assessed through a small-scale case study. This case study was carried out as a part of a wider research project and the results of the mathematical solution of the BLMOO in this case can be found in our partners' publications [30-33]. The case was evaluated by internal review and expert interviews.

4.1 Case Study

Case Design Problem. The design task in the case study was to dimension six storage towers of a part of a paper-making process and to guarantee the runnability and stability of the process. The dimensioned storage towers include TMP (thermo-mechanical pulp), chemical pulp, wet broke, dry broke, clean water and 0-water. The design problem is illustrated in Figure 3 and further explained in [31].

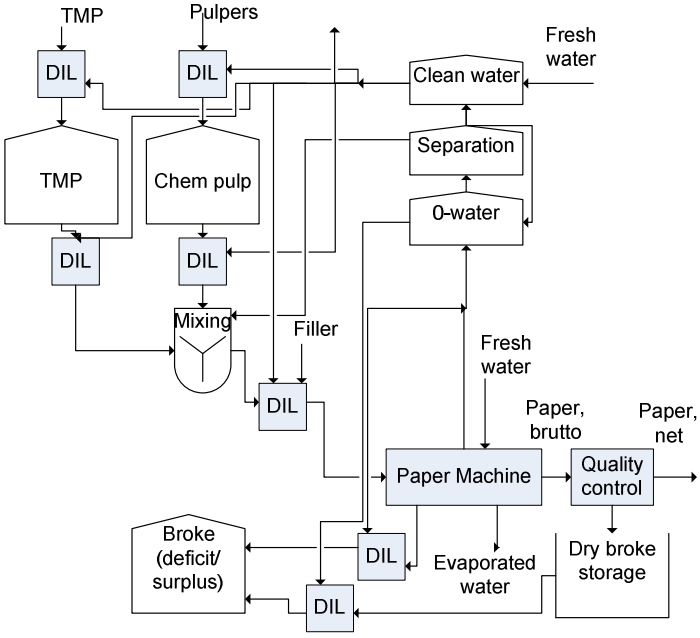


Fig. 3. Flow diagram of the process in case study

Stakeholders. The actors involved in the design process in the case study include the designer, optimizer and modeler. The roles were manned by research teams involved in the project.

The designer had the main responsibility of the project. He carried out the requirement elicitation with the end-customer, proposed a conceptual design and initiated the problem formulation for the optimization. He then had a key role in data acquisition for the model building. After getting the optimization results, he made the decisions according to the end-users preferences.

The optimizer participated in the problem formulation by having an opinion what kind of problems can be solved with optimization. After the problem formulation, the optimizer then asks the modeler to build necessary models for optimization and then chose the right optimization method. Finally, a suitable method for presenting the results was chosen.

The modeler was responsible for creating a model simple enough to be calculated. The modeler was also responsible to make sure that the simplifications do not affect to the problem to be solved.

Business Process. The project could be divided into four main tasks: problem formulation, model building, problem solving (optimization) and result interpretation.

Problem Formulation. At the starting point of the case study a part of the conceptual design was already performed, e.g. the number of storage towers and material flows between them was defined. The designer and optimizer then discussed the possibilities for a manageable optimization problem. They designed that the optimization activity concerns only about operation design and the dimensioning part of the structure design. Also the amount of optimized parameters was reduced in negotiations between the designer and the optimizer. During the optimization activity a mathematical model of the problem was created and used for finding an optimal design under the specified requirements. The design problem was formulated as follow:

$$\min_{d} \begin{Bmatrix} \sum_{i=1}^{4} H(V_{i,\max}) \\ E_{\psi}\left\{(q_{Filler}(n) - q_{0,Filler})^2\right\} \\ E_{\psi}\left\{(q_{bw}(n) - q_{0,bw})^2\right\} \\ E_{\psi}\left\{(q_{strength}(n))^2\right\} \\ E_{\psi}\left\{(u(n+1) - u(n))^2\right\} \\ -E_{\psi}\left\{T_v\right\} \end{Bmatrix} \tag{2}$$

where $H(V_{i,max})$ is the investment cost of the 4 selected tower volumes, T_V is the time till one of the towers goes empty or flows over, and $E_{\Psi}\{\}$ denotes the expectation value of the system performance as Ψ is the stochastic process with applied dosage policy.[31]

The operational problem, i.e. the lower level of the BLMOO was formulated as:

$$\min_{u} \begin{Bmatrix} \sum_{k=1}^{K_H} \gamma(k)(q_{Filler}(n+k) - q_{0,Filler})^2 \\ \sum_{k=1}^{K_H} \gamma(k)(qbw(n+k) - q_{0,bw})^2 \\ \sum_{k=1}^{K_H} \gamma(k)(q_{strength}(n+k) - q_{0,strength})^2 \\ \sum_{k=1}^{K_H} \gamma(k)(u(n+k) - u(n+k-1)^2 \\ \max\limits_{k=1..K_H} \dfrac{p(V_i(n+k) > V_{i,\max})}{p_i^{(up)}(k)} \\ \max\limits_{k=1..K_H} \dfrac{p(V_i(n+k) > V_{i,\min})}{p_i^{(down)}(k)} \end{Bmatrix} \quad (3)$$

$$s.t \quad u_{j,\min} \le u_j \le u_{j,\max}$$

where q_{filler} , q_{BW}, and $q_{strength}$ are the quality variables with q_0s as their nominal values, K_H is the optimization horizon, $\gamma(k)$ a time-wise weighting factor, u is a vector of pulp/water flows to be controlled, $p_i^{(up)}(k)$ and $p_i^{(down)}(k)$ are the accepted risks for a tower overflow/goes empty k time steps from the present time n defined as $p_i^{(up/low)}(k) = 1 - \left(1 - p_i^{(up/low)}\right)^k$, i refereeing to the storage towers for clean water, 0-water, broke, and dry broke. $V_{i,max}$ is the volume of the ith storage tower, i.e. the maximum amount of pulp/water in the tower, and $V_{i,min}$ is the minimum amount of pulp/water in the tower. U is the control variable describing the broke dosage from the broke tower to the system [31].

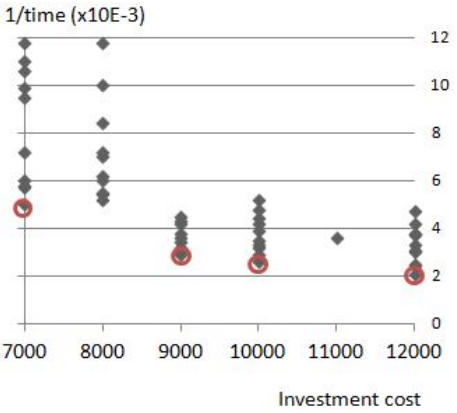


Fig. 4. Design solutions in respect to investment cost and time until production stop. Pareto optimal set of designs circled [31].

Simplified, on the operational level we optimize the variances of the quality attributes of the paper and the broke dosage and the probability of under/overflows. On the design level, we optimize the design according to the sizes of the tanks and expectation values of the system performance.

Model Building. At the same time that the optimizer negotiated with the designer about the problem formulation, he had to discuss with the model builder if a suitable model for the problem can be built. In this discussion there were two main themes: is the physical phenomenon of the problem known or is there enough data to model the problem stochastically and can the model be simple enough that it can be calculated fast enough in the optimization loop.

Optimization and Result Interpretation. In this case example, the tasks of problem formulation, model building and optimization were performed simultaneously and were highly iterative. The main focus of the case example was in optimization. The results of the optimization are described in [30-31].

After the optimization, the results were presented to the designer as two-dimensional Pareto optimal sets. In Fig.4, a Pareto optimal set in respect to the two most important parameters is presented. The designer then made the decisions e.g. between a decent investment cost and an acceptable probability of break.

Data, Knowledge and Models. The designer in this case had a wide experience in process design, paper making, modeling and optimization. The optimizer was mathematically oriented, but had only minor experience on paper making or process design. The modeler was familiar with process modeling and optimization.

The largest data flow in the process was from designer to optimizer. The designer had to communicate the customer requirements, the original design about the structure and operation and the freedoms and limitations for optimization in the design. The main models for this communication were a process flow sheet and steady-state model of the process. Making of these models was mainly a task for the designer. The designer was able to formulate most of the limitations and requirements in numerical form, e.g. the probability of the break may not be greater than Pmax. Due to the nature of a first time project, the data transfer between the optimizer and modeler was also huge.

Modeller was responsible for building three models: dynamic model, predictive model and a validation model. The two first mentioned were used in optimization while the validation model build with different simulation software was used only for one selected design.

Practically, the problem formulation and optimization required simultaneous model development, because there wasn't previous knowledge about feasible models.

The results of the optimization were delivered as a document containing simulation graphs and Pareto optimal sets (one example in Figure 4) of optimization results.

Information Tools. This case example was carried out as a research project, and therefore the engineering tools used didn't match the ones used in industry. MATLAB was used both for the optimization and simulation for optimization. APROS process simulator was used in validating the results of optimization.

4.2 Interviews

In order to get information of the process engineering business process today and to validate the proposed changed to the process in order to adopt a new optimizing design process, a set of interviews were performed. The interviewees represented actors in both chemical and pulp & paper industries and contained process designers, automation designers and IT-system experts in process design companies and engineering enterprises. In addition a simulation expert and an optimization expert were interviewed.

The topics of the interviews were motivation and feasibility of optimizing design, current design practices vs. optimizing design and IT systems vs. requirements of optimizing design.

The following observations could be made about issues concerning *the motivation and feasibility of optimizing process design:*

There are business requirements to decrease the costs of plant design projects. At the same time the quality of the design should be increased and cost decreased. The effect of optimizing process design process on all three aspects (design quality, design cost, project cost) should be taken into account.

The process design practices in different industries are heterogeneous. In paper and pulp industry process design can be characterized as engineering-oriented, i.e. an engineering design system is the primary design tool. As a comparison, in chemical industry process design is quite simulation-oriented, i.e. a simulator is the primary design tool. The design practices of chemical industry are closer to the optimizing process design process than the ones in paper and pulp industry.

The following observations could be made about issues concerning *the differences between current design practices and optimizing design*:

Cooperation between different parties involved in a design project has recently been emphasized by engineering companies. Cooperation is needed for the efficiency of a design process, e.g. finding out the requirements of the customer early enough, ensuring consistency of the designs from different designers and handling the effects of design changes. The optimizing design process should fit to the cooperation practices.

The design of a process is divided to several designers according to different systems or parts of the process. This is done due to the different expertise of the designers and concurrency of the design work. There are usually some buffers in the design between the designs by separate designers. From the optimization viewpoint this division is questionable. The optimizing design process is likely to change the division of work.

The division of work is also reflected to current optimization practices. They are optimizing unit processes rather than the whole process. The optimizing design process should change this practice, too.

The trust of the customer on the feasibility of the process design in a very important issue, which is affected by many factors, e.g. references of the vendor and difference of the design to existing ones. It was mentioned that particularly in the paper and pulp industry customers do not trust simulations as a process design validation tool. Validation of the design results should be a primary concern also in optimizing process design.

The following observations could be made about issues concerning the *differences between requirements for current IT systems and IT systems when using optimizing design*:

The IT-architecture of an engineering company is usually quite heterogeneous, i.e. there are several different IT-systems used during a design project. Sometimes there are even several alternative IT-systems for same design tasks, e.g. due to customer requests. The heterogeneity of IT-systems may hinder the implementation of IT-support optimizing design.

There is a slowly progressing shift from document-centered design paradigm to data-centered design paradigm in plant design. The optimizing process design process should be made to fit the data model -oriented design paradigm because its meaning seems to be increasing in the future. It is also likely to be more suitable basis for optimizing design than the older document-oriented design.

The amount of data which is transferred during the optimization process is moderate but the iterations between the problem formulation, model building, optimization and decision making may justify the usage of a proper tool for data exchange and version management.

5 Conclusions

In this paper a business process model for optimizing design in continuous process facility engineering has been presented. This model was considered from the viewpoints of stakeholders, process, knowledge, data, models and tools. The model for optimizing design was assessed by applying it in an experimental case study and by interviewing experts.

Based on this study, some conclusions can be made. The greatest change is the new roles of optimizer and modeler, which make the process more iterative between optimizer and process designer. The new roles require a shared knowledge, because the work can be described as expert co-operation. The business process in optimizing design is more iterative than in traditional design because of the need for negotiation in the problem formulation and the uncertainties in the modeling. The presentation of results of the optimization should be considered more carefully. The amount of objective functions presented simultaneously may not be too large, but some priorization should be made.

In addition to this, the interviews also illustrate the importance of validation of process designs. Validation of the designs so that the customer will trust them is a primary concern to be observed in future research.

It must be noted that the design business process in this paper is presented at a general level and it must be specified when used as actual process. The description of roles should take the organization boundaries and current responsibilities into account. E.g. the customer organization usually has a business oriented decision maker and technical decision maker roles. The roles in engineering organization should correspond to these roles. In future, the process model is evaluated and specified in a larger case study.

Acknowledgements. This research was supported by Forestcluster Ltd and its Effnet program.

References

1. Watermeyer, P.: Handbook for Process Plant Project Engineers. Wiley (2002)
2. Cziner, K.: Multicriteria process development and design (Internet). Helsinki University of Technology, Espoo (2006), http://lib.tkk.fi/Diss/2006/isbn9512284294/isbn9512284294.pdf
3. Tuomaala, M.: Conceptual approach to process integration efficiency (Internet). Helsinki University of Technology Department of Mechanical Engineering, Espoo (2007), http://lib.tkk.fi/Diss/2007/isbn9789512287888/isbn9789512287888.pdf
4. Diesen, M.: Economics of the Pulp & Paper Industry. Technical Association of the Pulp and Paper (1998)
5. Seuranen, T.: Studies on computer-aided conceptual process design (Internet). Helsinki University of Technology, Espoo (2006), http://lib.tkk.fi/Diss/2006/isbn9512282674/
6. Douglas, J.: Conceptual Design of Chemical Processes, 1st edn. McGraw-Hill Science/Engineering/Math. (1988)
7. Bollinger, R.E., Clark, D.G., Dowell, R.M., Ewbank, R.M., Hendershot, D.C., Lutz, W.K., et al.: Inherently Safer Chemical Processes: A Life Cycle Approach. Wiley-AIChE (1997)
8. Pajula, E.: Studies on Computer Aided Process and Equipment Design in Process Industry (Internet). Helsinki University of Technology, Espoo (2006), http://lib.tkk.fi/Diss/2006/isbn9512284898/isbn9512284898.pdf (cited February 10, 2011)
9. Seuranen, T., Pajula, E., Hurme, M.: Applying CBR and Object Database Techniques in Chemical Process Design. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 731–743. Springer, Heidelberg (2001)
10. Fathianathan, M., Panchal, J.H.: Incorporating design outsourcing decisions within the design of collaborative design processes. Comput. Ind. 60, 392–402 (2009)
11. Ylén, J., Paljakka, M., Karhela, T., Savolainen, J., Juslin, K.: Experiences on utilising plant scale dynamic simulation in process industry. In: Proc. of 19th European Conference on Modeling and Simulation EMCS (2005)
12. Pulkkinen, P., Ihalainen, H., Ritala, R.: Developing Simulation Models for Dynamic Optimization, Vesterås Sweden (2003)
13. Fathy, H.K., Reyer, J.A., Papalambros, P.Y., Ulsov, A.G.: On the coupling between the plant and controller optimization problems. In: American Control Conference, Proceedings of the 2001, vol. 3, pp. 1864–1869 (2001)
14. Dempe, S.: Foundations of bilevel programming, 322 p. Springer (2002)
15. Eichfelder, G.: Multiobjective bilevel optimization. Mathematical Programming 123, 419–449 (2010)
16. Linnala, M., Ruotsalainen, H., Madetoja, E., Savolainen, J., Hämäläinen, J.: Dynamic simulation and optimization of an SC papermaking line - Illustrated with case studies. Nordic Pulp and Paper Research Journal 25(2), 213–220 (2010)
17. Comos Industry Solutions GmbH - Innovation Worldwide (Internet), http://comos.com/14.html?&L=1 (cited February 10, 2011)

18. Intergraph PP&M | Leading global provider of engineering software to the process, power, and marine industries (Internet), http://www.intergraph.com/ppm/ (cited October 20, 2011)
19. Plant Design and Engineering Software Products from Bentley (Internet), http://www.bentley.com/en-US/Products/Plant+Design+and+Engineering/ (cited October 20, 2011)
20. 15926.ORG: HomePage (Internet). http://15926.org/home/tiki-index.php (cited October 20, 2011)
21. iRINGUserGroup (Internet), http://iringug.org/wiki/index.php?title=Main_Page (cited October 20, 2011)
22. POSC Caesar – Trac (Internet), https://www.posccaesar.org/ (cited October 20, 2011)
23. Batres, R., West, M., Leal, D., Price, D., Masaki, K., Shimada, Y., et al.: An upper ontology based on ISO 15926. Computers and Chemical Engineering 31(5-6), 519–534 (2007)
24. Wiesner, A., Morbach, J., Marquardt, W.: Information integration in chemical process engineering based on semantic technologies. Computers and Chemical Engineering 35(4), 692–708 (2011)
25. Deb, K., Sinha, A.: Solving Bilevel Multi-Objective Optimization Problems Using Evolutionary Algorithms. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 110–124. Springer, Heidelberg (2009)
26. Branke, J., Deb, K., Miettinen, K., Slowinski, R.: Multiobjective Optimization: Interactive and Evolutionary Approaches, 1st edn. Springer (2008)
27. Miettinen, K.: Nonlinear Multiobjective Optimization, 1st edn. Springer (1998)
28. Belton, V., Stewart, T.J.: Multiple criteria decision analysis: an integrated approach. Springer (2002)
29. Konda, S., Monarch, I., Sargent, P., Subrahmanian, E.: Shared memory in design: A unifying theme for research and practice. Research in Engineering Design 4(1), 23–42 (1992)
30. Ropponen, A., Ritala, R., Pistikopoulos, E.N.: Broke management optimization in design of paper production systems. In: 20th European Symposium on Computer Aided Process Engineering, pp. 865–870. Elsevier (2010)
31. Ropponen, A., Rajala, M., Ritala, R.: Multiobjective optimization of the pulp/water storage towers in design of paper production system, pp. 612–616 (2011)
32. Ropponen, A., Ritala, R., Pistikopoulos, E.N.: Optimization issues of the broke management system in papermaking. Computers and Chemical Engineering 35(11), 2510–2520 (2011)
33. Eskelinen, P., Ruuska, S., Miettinen, K., Wiecek, M., Mustajoki, J.: A scenario-based interactive multiobjective optimization method for decision making under uncertainty, Coimbra, Portugal (2010)

Variation-Aware Circuit Macromodeling and Design Based on Surrogate Models

Ting Zhu, Mustafa Berke Yelten, Michael B. Steer, and Paul D. Franzon

Department of Electrical and Computer Engineering,
North Carolina State University, Raleigh, NC 27695, U.S.A.
{tzhu,mbyelten,mbs,paulf}@ncsu.edu

Abstract. This paper presents surrogate model-based methods to generate circuit performance models, device models, and high-speed IO buffer macromodels. Circuit performance models are built with design parameters and parametric variations, and they can be used for fast and systematic design space exploration and yield analysis. Surrogate models of the main device characteristics are generated in order to assess the effects of variability in analog circuits. A new variation-aware IO buffer macromodel is developed by integrating surrogate modeling and a physically-based model structure. The new IO model provides both good accuracy and scalability for signal integrity analysis.

Keywords: Surrogate Modeling, Macromodel, Variation-Aware, Circuit, Device Model, Design Exploration, IO Buffer.

1 Introduction

Advances in integrated circuit (IC) technologies have enabled the single-chip integration of multiple analog and digital functions, resulting in complex mixed-signal Systems-on-a-Chip (SoCs). However, as the IC technology further scales, process variations become increasingly critical and lead to large variances in the important transistor parameters. As a result, circuit performance varies significantly, and some circuits may even fail to work. The large process uncertainties have caused significant performance yield loss. In addition, reliability issues and environmental variations (such as supply voltage and temperature) contribute to further yield reduction and make it more challenging to create a reliable, robust design. In handling this problem, it is important to consider the effects of variations in circuit modeling and design analysis at an early stage. However, this is a nontrivial task. In this paper, we apply surrogate modeling to handle the complexities in variation-aware circuit macromodeling, design analysis, and device modeling. We demonstrate the benefits of using surrogate modeling in enhancing the accuracy, flexibility, and efficiency in those applications.

2 Circuit Performance Macromodeling with Variations

2.1 Overview of the Method

Circuit designers are confronted with large design spaces and many design variables whose relationships need to be analyzed. In this situation, tasks such as sensitivity

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 255–269.
DOI: 10.1007/978-3-642-34336-0_17

analysis, design space exploration, and visualization become difficult, even if a single simulation takes only a short period of time. The analyses are getting impractical as when some of the circuit simulations are computationally expensive and time-consuming. Moreover, when variations are considered in a circuit design, the situation becomes even more complex. One way to reduce the design complexities and costs is to build performance models which can be used as replacements for the real circuit performance responses.

In this work, performance models are built by directly approximating circuit performance parameters (e.g. S-parameter, gain, power consumption, noise figure, etc.) with design variables (e.g. transistor size, bias voltage, current, etc.) and parametric variations (e.g. V_{th}, t_{ox}, L_{eff}). The idea is illustrated in Fig. 1. This method is data-driven and black-box by nature, and thus it can be applied to a wide range of circuit design problems.

2.2 Model Construction

Techniques. Global surrogate modeling [1] is used to create performance models with good accuracy over the complete design space. This is different from building local surrogate model for the purpose of optimization [2].

Surrogate modeling accuracy and efficiency are determined by several key factors including the sampling plan, model template, and validation. These factors are the three steps in surrogate modeling. Multiple techniques are available and they need to be carefully selected according to the nature of the problem and computational complexity.

In the first step, the key question in designing the sampling plan is how to efficiently choose samples for fitting models, considering that the number of samples is limited by the computational expense. Traditionally, methods such as Latin Hypercube sampling or orthogonal arrays, is used for one-shot sampling [3]. Recently, adaptive sampling techniques were developed in order to achieve better efficiency in sampling [4, 5]. Adaptive sampling is an iterative sampling process which analyzes the data from previous iterations in order to select new samples in the areas that are more difficult to fit.

In the model template selection step, the surrogate model type needs to be determined. Popular surrogate model types include Rational Functions, Kriging models, Radial Basis Function (RBF) models, Artificial Neural Networks (ANNs), and Support Vector Machines (SVMs). After the model type has been selected, model complexity also needs to be decided. Model complexity is controlled by a set of hyper-parameters which would be optimized during a modeling process.

The step of model validation establishes the predictive capabilities of the models and estimates their accuracy. One popular method is five-fold cross-validation [6] in which the training data are divided into five subsets. A surrogate model is constructed five times, each time four subsets are used for model construction and one subset is used for error measurement. Model error can be measured as a relative error, for example Root Relative Square Error (RRSE), Bayesian Estimation Error Quotient (BEEQ), etc., or an absolute error, e.g. Maximum Absolute Error (MAE), Root Mean Square Error (RMSE), etc.

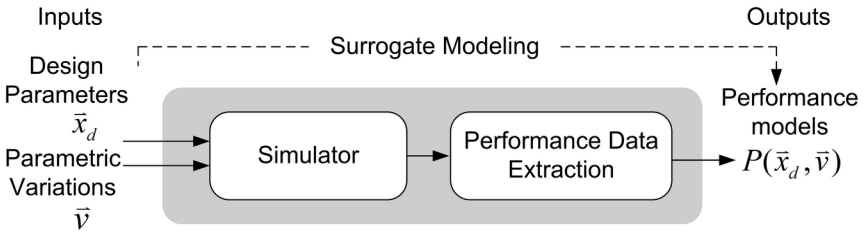


Fig. 1. Circuit performance modeling

Automatic Modeling Flow. In this work, we constructed an automatic modeling flow that is able to generate performance models from transistor-level circuit simulations, as shown in Fig. 2. Before the modeling starts, a set of input and output parameters are defined. The modeling techniques are also configured, including the model template, adaptive sampling strategy, and accuracy measurement. An accuracy target is defined as well. At the beginning of the modeling process, a small set of initial samples are generated. Then transistor-level SPICE simulations are performed using this initial set, and the corresponding responses are collected and used as the modeling data. Surrogate models are then constructed and their parameters optimized. The model accuracy is measured and the optimization continues until only negligible improvements can be made by changing the model parameters. If the desired accuracy is not reached, the adaptive sampling is evoked to add a new set of samples. The process continues until the fit reaches the targeted accuracy. When the process finishes, the model expressions are be exported and used in the follow design steps.

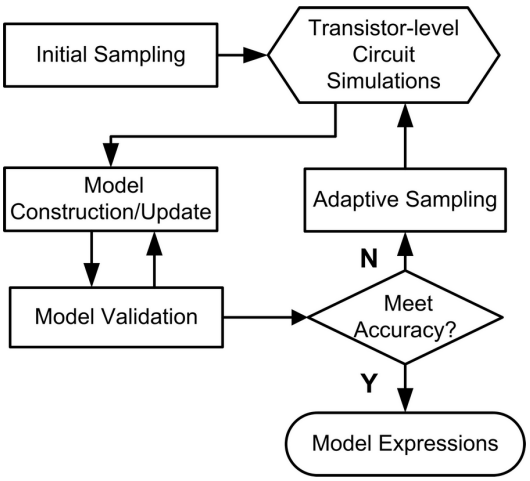


Fig. 2. Automatic adaptive performance surrogate modeling flow

In the examples presented in this section, the modeling techniques are explored using the SUrrogate MOdeling (SUMO) Matlab Toolbox [7]. SUMO is a plug in-based, adaptive platform that can be customized flexibly. The toolbox makes it feasible to test a variety of modeling techniques. Transient circuit simulators, including Cadence Virtuoso Spectre®, and Synopsys HSPICE®, are used here for performing transistor-level circuit simulations.

2.3 Circuit Case Demonstration

Performance models are very helpful for visualizing the design space and gaining insight into circuit behavior. In this section, a low-noise-amplifier (LNA) circuit is designed in a 0.13 μm CMOS process [8], the simplified circuit schematic of which is shown in Fig.3.

In this example, we consider the transistors' driving strength (*tr_strengh*) as a main source of process variations. Transistor strength describes the variations in the tran sistor speed and the current. The data is provided by the foundry and it is set between -3σ and +3σ. Additionally, temperature is considered as an environmental variation, and it varies in the range of -20°C to 60°C. Two design parameters are considered. One is reference current I_{ref} which is used to generate the input DC bias. I_{ref} is set in the range of 50 μA to 200 μA. The other parameter is mlna which is the multiple of the widths of the amplifier transistors M1and M2. mlna is in the range of 0.5 to 2. Here the performance of interest is the voltage gain at the center frequency (*maxlnagain*).

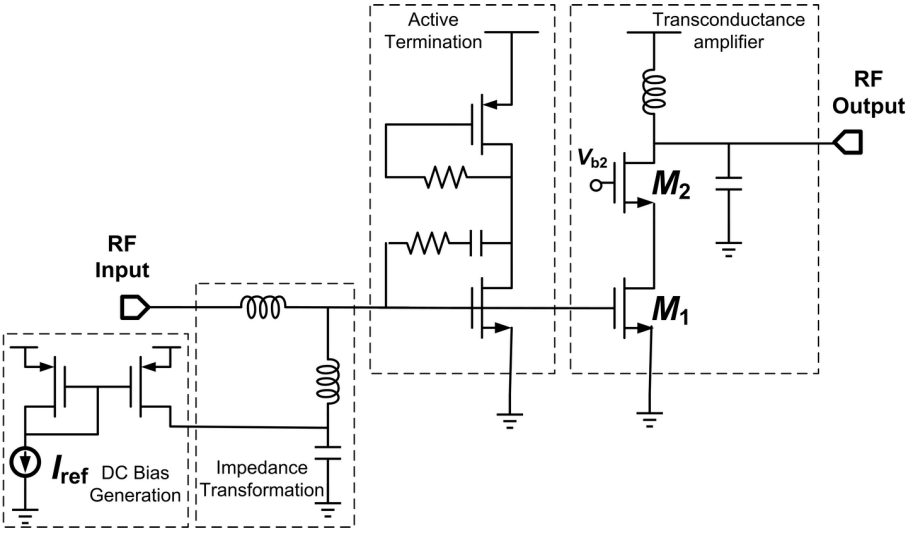


Fig. 3. Simplified low-noise-amplifier circuit schematic

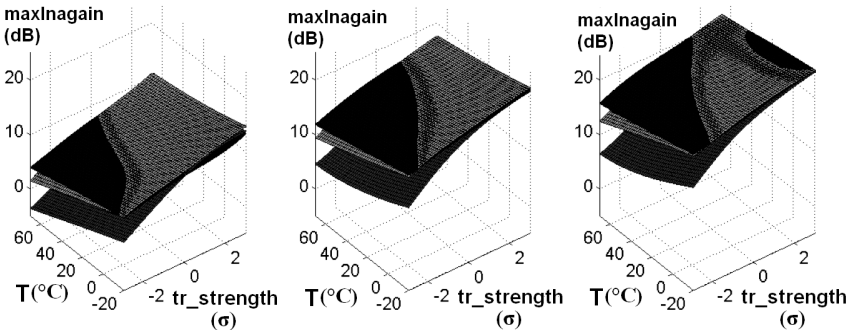
The Kriging performance model *maxlnagain*(*tr_strength*, *T*, *mlna,* I_{ref}) was constructed using the data obtained from transistor-level simulations in HSPICE®. Latin Hypercube sampling with corner points was used as the initial sampling strategy.

The adaptive sampling method LOLA-Voronoi [5] determined the non-linear regions of the true response and sampled those more densely. 5-fold cross validation with root-relative-square-error (RRSE) was used for the model validation. The definition of RRSE is defined as

$$\mathrm{RRSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2}{\sum_{i=1}^{n}(y_i - \tilde{y})^2}} \tag{1}$$

where y_i, $\tilde{y}_i$ and $\tilde{y}$ are the actual, predicted, and mean actual response values.

The constructed model has an RRSE of 4.72%. Fig. 4 shows the plots of the model surfaces used to explore the design space. The results show the effects of the design parameters and the parametric variations. It is seen that both transistor variations and temperature variations can significantly impact performance. It is possible to modulate the design parameters, in order to achieve an optimal gain value under the specific variations.



(a) mlna=0.5 (b)mlna=1.25 (c)mlna=2

Fig. 4. Plot of the model surfaces. (a)—(c) are for different mlna values. The three slices in each plot are for three I_{ref} values. Black is for 50 μA, light grey is for 125μA, and dark grey is for 200μA.

3 Scalable and Variation-Sensitive IO Macromodel

Good macromodels of input/output circuits are essential for fast timing, signal-integrity, and power-integrity analysis in high-speed digital systems. The most popular approach to IO modelling is to use the traditional table-based input-output buffer information specification (IBIS) [9]. IBIS models are simple, portable, IP-protected, and fast in simulation. However, they are unable to simulate continuous PVT variations and unsuitable for statistical analysis. We propose a new type of macromodel, called the surrogate IBIS model, to solve the problem [10]. In the new method, an equivalent circuit structure is used to capture the static and dynamic circuit behaviors, while surrogate modeling is used to approximate each element over a range of Process-Voltage-Temperature (PVT) parameters, so that the macromodel is able to dynamically adapt to the PVT variations in analysis.

3.1 Proposed Macromodel Structure

Fig. 5 shows the proposed macromodel structure that is composed of physically-based equivalent model elements [10]. I_{pu} and I_{pd} represent the nonlinear output current. Time-variant coefficients K_{pu} and K_{pd} determine the partial turn-on of the pull-up/down networks during switching transitions. C_{power} and C_{gnd} represent the nonlinear parasitic capacitance between the output and the supply rails. Surrogate models of these model elements are constructed, to capture the effects of supply voltage, terminal voltages, semiconductor process, and temperature.

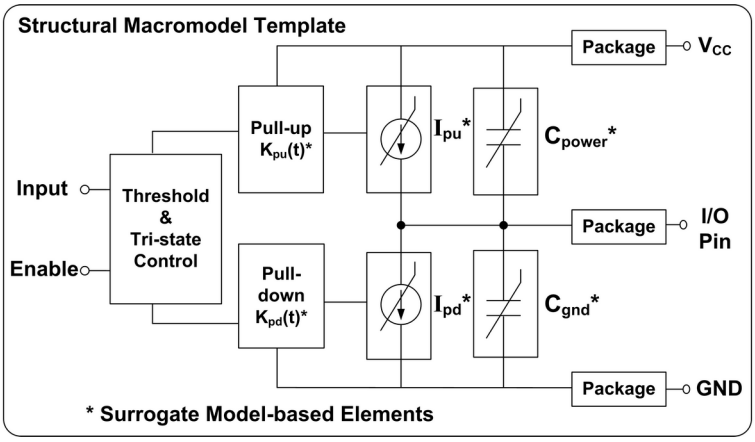


Fig. 5. Structural IO buffer macromodel template with surrogate model elements

3.2 Macromodel Construction

The automatic modeling process described in Section 2 was used to construct surrogate models for the model elements in Fig. 5. The method is demonstrated with the single-ended output buffer circuit shown in Fig. 6. The circuit is designed in 180 nm CMOS process with a 3.3 V normal supply voltage. The threshold voltage variations ΔV_{th} in the MOS transistors are considered as the main process variations and they are assumed to be within ±20% of the nominal value V_{th0}. The parameter $P = \Delta V_{th}/V_{th0}$ is used to describe the threshold voltage variation. The supply voltage Vs is assumed to fluctuate within ±30% of the nominal supply (3.3 V) and temperature (T) is set in the range of 0 to 100°C. In the modeling process, those PVT-related parameters are sampled adaptively in their ranges.

Here modeling data was extracted from transistor-level SPICE circuit simulations. Fig. 7 (a) shows the circuit test-bench to extract the pull-up output current I_{pu} (V_S, V_{pu}, T, ΔV_{th}). The parameter V_{pu} is defined as the voltage difference between the power supply rail and the output, and it ranges from $-V_{CC}$ to $+2V_{CC}$ covering the maximum reflection case [11]. Transient simulations were performed and the simulation time was long enough (in this case it was 1 ms with 1 ns step size) to record a stable output current I_{pu}.

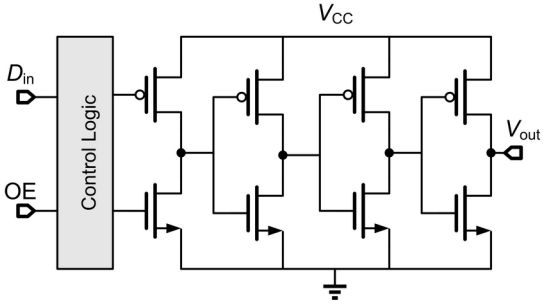
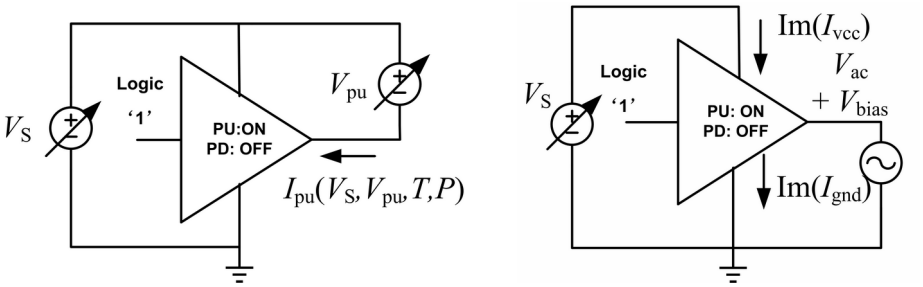


Fig. 6. Simplified schematic of the driver circuit



(a) I_{pu} (b) Capacitance

Fig. 7. Test-benches for extracting model elements: (a) pull-up current I_{pu} (a) output capacitance C_{gnd} and C_{power}

The data was used to fit rational function models in the form:

$$f(X) = \frac{P(X)}{Q(X)} \tag{2}$$

where P and Q are polynomial functions in $X=\{x_1,x_2,\ldots,x_n\}$and Q is non zero. P and Q have no common factor of positive degree.

Similarly, the pull-down current model I_{pd} was extracted by turning on the pull-down network and turning off the pull-up network. I_{pd} was extracted as a model function of PVT variations and V_{pd}, where V_{pd} is defined as the voltage difference between the output and the ground.

The test setup for extracting the output parasitic capacitance is shown in Fig. 7(b). An AC signal is attached to the output ports and the imaginary currents in the power and the ground ports are measured. The capacitances C_{power} and C_{gnd} were derived using

$$C_{power} = \frac{\Im(I_{VCC})}{2\pi f V_{AC}}, \qquad C_{gnd} = \frac{-\Im(I_{gnd})}{2\pi f V_{AC}} \tag{3}$$

where $\Im(I_{VCC})$ and $\Im(I_{gnd})$ are the imaginary parts of the measured currents, f is the frequency of the AC source, and V_{AC} is the AC voltage amplitude. The time-variant transition coefficients K_{pu} and K_{pd} were obtained according to the 2EQ/2UK

algorithm [12]. Fig. 8(a) shows the test to obtain the switching output voltage waveforms. A simplified circuit to illustrate the 2EQ/2UK algorithm is shown in Fig. 8(b). The switching output voltage waveforms wfm_1 and wfm_2 were obtained with different terminal voltage V_{term}, and the unknown coefficients K_{pu} and K_{pd} are derived using the equations

$$\begin{aligned} K_{pu}(t)I_{pu}(V_{wfm_1}(t)) - K_{pd}(t)I_{pd}(V_{wfm_1}(t)) - I_{out} = 0 \\ K_{pu}(t)I_{pu}(V_{wfm_2}(t)) - K_{pd}(t)I_{pd}(V_{wfm_2}(t)) - I_{out} = 0 \end{aligned} \tag{4}$$

where $I_{out} = (V_{out} - V_{term}) / R_{load}$. I_{pu} and I_{pd} are the output current models.

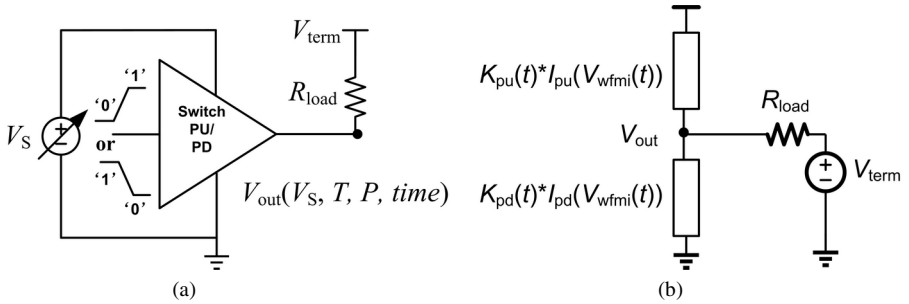


Fig. 8. (a) Test-benches for extracting model elements output capacitance C_{gnd} and C_{power} (b) illustration of 2EQ/2UK algorithm.

To implement the new model, we modified the Verilog-A behavioral version of the IBIS model [13] and applied the surrogate model expressions for the model elements. The surrogate models were implemented in the form of analog functions.

3.3 Test Results

In this section the surrogate IBIS model is compared to the reference provided by the transistor-level simulation, and to the traditional IBIS model extracted from SPICE using the S2IBIS3 v1.0 tool [14].

The test setup is shown in Fig. 9 where the driver is connected to a 0.75-m long lossy transmission line (RLGC model) with a loading resistor. The characteristic impedance of the transmission line is 50 Ω. The loading resistor is 75 Ω. Two test cases were examined. The results are shown in Fig. 10.

1. Case 1, used a 250 MHz square wave as a test input signal. The input data has the pattern "01010" with a 0.1-ns rise/fall time and 2-ns bit-period. The supply voltage varied from 2.8 to 3.8 V.
2. Case 2, used a data pattern with a 1024 bit long pseudorandom bit sequence (PRBS) with 2-ns bit time. The power supply voltage was constant.

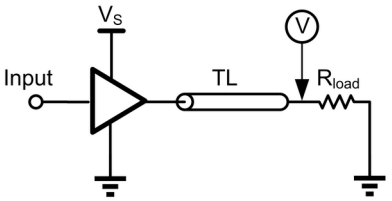
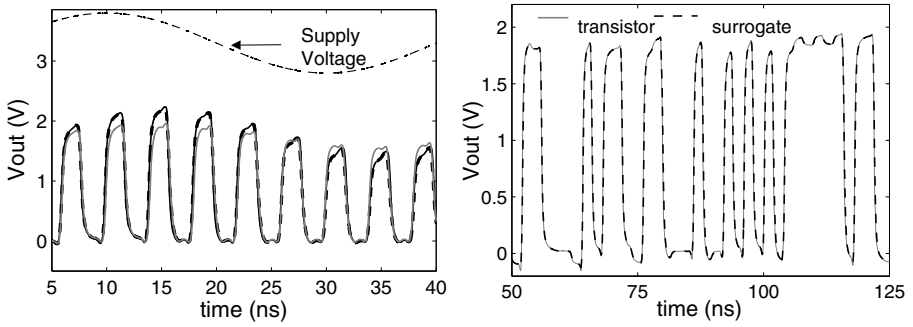


Fig. 9. Test setup for model validation



(a) Case 1 (b) Case 2

Fig. 10. Output voltage at the far end of the transmission line, (a) Case 1, black solid line—transistor-model, grey solid line—traditional IBIS, black dash line—proposed surrogate IBIS. Black dash-dot line—supply voltage. (b) Case 2, grey solid line—transistor, black dashed line—macromodel.

The accuracy of the macromodels is quantified by computing the timing error and the maximum relative voltage error. The timing error is defined as the time difference between the reference and the macromodel voltage responses measured for crossing half of the output voltage swing. The maximum relative voltage error is defined as the maximum error between the reference and macromodel voltage responses divided by the voltage swing.

The results show that in Case 1 when there are large variations of the supply voltage, the surrogate IBIS model has much better accuracy both of the timing error and of the relative voltage error than the traditional IBIS model. The maximum timing error of the surrogate-IBIS model is 79 ps, and the maximum relative voltage error is 6.77%. The surrogate IBIS model achieves the improved accuracy by capturing the complex output capacitance characteristics, the effects of the supply voltage, and gate modulation effects on the output current [15]. In Case 2, the result shows that the surrogate-IBIS achieves good accuracy. In this case, the maximum timing error is 70 ps (3.5% of the bit-time) and the maximum relative voltage error is 6.45%. We also analyze the eye-diagram of the output in Case 2. The eye-width (W) was measured when the eye-height (H) was equal to 1 V. The results under different PVT conditions show that the eye-width differences within 0.04 ns (2% of the bit-time).

In summary, the proposed surrogate-IBIS macromodel achieves good accuracy in the analysis. The macromodels obtained show good accuracy in capturing the effects of reflections and variations, and their scalability makes flexible design analysis possible.

4 Surrogate-Based Device Modeling

Scaling of device sizes induced high variability of transistor parameters. There are two major reasons for this. Firstly, quantum mechanics-based phenomena such as the drain induced barrier lowering (DIBL) or gate tunnelling which are negligible in long-channel devices become more significant. Additional physics-based effects increased the dependence of many circuit design quantities including the drain current, I_{ds}, and device transconductance, g_m, on the transistor process parameters such as the oxide thickness, t_{ox}. Furthermore, the tolerance of semiconductor manufacturing components did not scale down as the transistor sizes shrink [16]. As a consequence, the amount of uncertainty in the design quantities remained constant while device sizes become smaller leading to higher percentages of variability with respect to the nominal values of the transistor process parameters. The experimental data revealed that the traditional process corner analysis might not reflect the real distribution of the critical transistor parameters such as the threshold voltage V_{th} [17] while the Monte Carlo analysis become more computationally intensive with increasing number of variability factors.

The response surface of design quantities which become more complex with the presence of extreme process variations can be accurately captured by surrogate modelling. Surrogate modelling aims to express the output quantity in terms of a few input parameters by evaluating a limited number of samples. These samples are used by the basis functions which establish the response surface of the desired output. Coefficients of the basis functions should be optimized to minimize the modelling error. This approach has been applied to the problem of I_{ds} modelling in order to assess the effects of variability in analogue circuit building blocks, in particular, the differential amplifiers [18]. In this section, the modeling of g_m of n-channel transistors will be discussed.

The transconductance g_m is an important quantity for analog circuits, particularly in determining the AC performance of amplifiers, mixers, and voltage controlled oscillators. The modeling here is based on 65 nm device technology (IBM 10SF design kit) and uses six process parameters (t_{ox}, intrinsic threshold voltage $V_{th,0}$, intrinsic drain-source resistance $R_{ds,0}$, intrinsic mobility μ_0, channel length variation ΔL_{eff}, and channel doping N_{ch}) as input to the model in addition to the terminal voltages of the transistor (gate-source voltage V_{gs}, drain-source voltage V_{ds} , and bulk-source voltage V_{bs}) and the temperature T. The choice of these process parameters is based on their physical origin which ensures a weak correlation between each parameter. BSIM model I_{ds} equations are analytically differentiated to yield g_m [19]:

$$g_m = \partial I_{ds} / \partial V_{gs} . \tag{5}$$

The g_m expression is validated by extensive SPICE circuit simulations over the process corners and at temperature extremes so that it can be used to evaluate the

samples, each a function of the ten parameters described above. Although an analytic equation for g_m is used in this work, the modelling methodology is general and can employ simulations or measurement results given that they have the same input and output parameters.

Kriging basis functions are used to construct the surrogate model with the necessary coefficients being optimized using the MATLAB toolbox Design and Analysis of Computer Experiments (DACE) [20]. The device width is assumed to be 10 µm. The finalized model is tested for accuracy using the root relative square error (RRSE) metric where RRSE can be given by Equation (1).

The g_m model is constructed using a total number of 2560 input samples, and tested with 6400 samples other than the input samples. The resulting model yields an RRSE of 3.96% indicating to a high level of accuracy.

The model can be used to observe the changes in g_m with respect to its input parameters. Examples of this are provided in Figure 8. The graphs provide critical insight to the designer about the fundamental relations and trade-offs between the chosen process parameters, terminal voltages, and temperature. Higher g_m values are obtained with smaller $V_{th,0}$, L_{eff}, and t_{ox}, as well as larger μ_0. This information becomes especially vital when variability of the circuit performance that depends on g_m must be considered. In the example of an RF cascode low-noise amplifier , voltage gain A_v, input and output return ratios, S_{11} and S_{22}, as well as the optimum noise impedance, Z_{opt}, are complex functions of the g_m value of the common source transistor [21]. Any

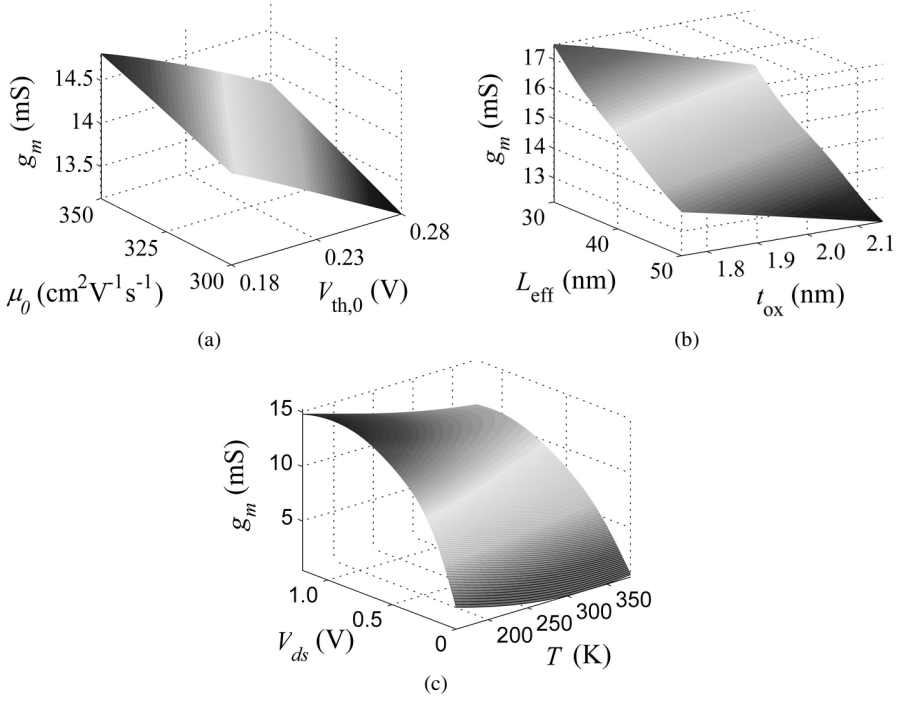


Fig. 11. 3D graphs showing the trade-offs between the different inputs on the modeled g_m

variability of the process parameters of this transistor may push the design outside of the specification range. In this case, information presented in Fig.11 (a)—(c) can be used to change the matching network of the amplifier such that it can yield the desired design metrics in all cases of process variability.

Finally, it should be noted that surrogate model-based device modeling is not limited to one single design quantity. Response surface models of other important design metrics can also be developed by using the methodology described here. As an example, consider the bandwidth of a single-stage amplifier. The bandwidth is both a function of process parameters used in g_m modeling and a function of the junction capacitances of the transistor. However, these junction capacitances depend also on some process parameters. The exact relationships can be quantified by analytical expressions as given in the device model equations [19]. Once the additionally required parameters are determined, then the surrogate modeling process can be applied as in g_m modeling.

5 Surrogate Model-Based Circuit Design

5.1 Yield-Aware Circuit Optimization

As IC technologies scale down to 65 nm and beyond, it is more challenging to create reliable and robust designs in the presence of large process (P) and environmental variations (e.g. supply voltage (V), temperature (T)) [22]. Without considering PVT fluctuations, the optimal circuit design would possibly minimize the cost functions by pushing many performance constraints to their boundaries, and result in a design that is very sensitive to process variations. Therefore, we need to not only search for the optimal case at the nominal conditions, but also carefully consider the circuit robustness in the presence of variations. However, the fulfillment of all these requirements introduces more complications in circuit design.

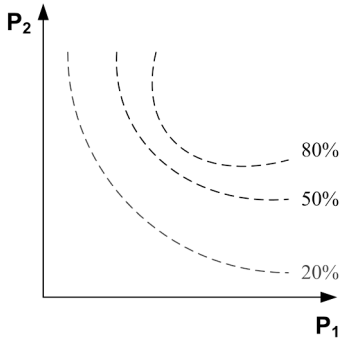


Fig. 12. Illustration of Pareto fronts with different yield levels

Yield is defined as the number of dies per wafer that meet all predefined performance metrics. Monte Carlo analysis of a circuit is an important technique used for yield estimation. However, this method requires a large number of sampling points to achieve sufficient accuracy and therefore it is very time-consuming. One solution to

reduce computational cost is to use the performance surrogate models proposed in Section 2. As performance models are constructed as a function of selected design parameters and parametric variations, they can be used instead of using circuit-level simulations. Therefore, yield estimation can be achieved without large computational cost.

One application of a variation-aware performance model is to obtain the yield-aware Pareto fronts [23] which is best trade-offs of the overall circuit performance and yield. In this application, in addition to searching for the general Pareto-optimal designs, performance yield at those design points is evaluated by using the variation-aware performance model. As a result, the yield-aware Pareto fronts can be generated. An illustration is shown in Fig. 12. P_1 and P_2 are the performance parameters to trade-off, and the curves are the Pareto fronts with different yield levels. The yield-aware Pareto fronts of sub-blocks could be further used in yield-aware system design.

5.2 Surrogate-Based Circuit Optimization

Simulation-based circuit optimization is a very good application of surrogate modeling, as the process requires a great number of iterative evaluations of objective functions. In an optimization process, surrogate models are used to guide the search instead of achieving the global accuracy.

In the surrogate-based optimization process, generally there are two types of simulation models, a low-fidelity and a high-fidelity model. In our circuit design problems, the transistor-level circuit simulation is used as a high-fidelity model while the built surrogate model is used as the low-fidelity model. The general surrogate-based optimization process is shown in Fig. 13 [24].

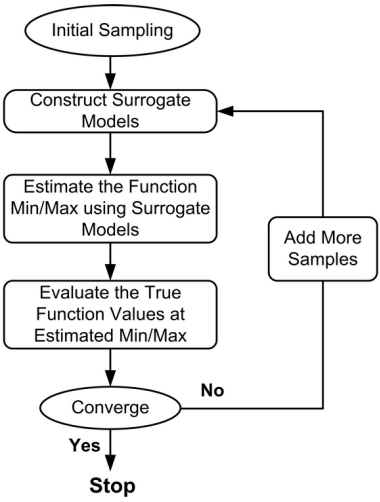


Fig. 13. General surrogate-based optimization flow

The Gaussian-based Kriging model can be used as an approximation method since this model is able to provide estimation of the uncertainty in the prediction. Adaptive sampling methods (e.g. [3]) can be used to balance exploration (improving the general accuracy of the surrogate model) and exploitation (improving the accuracy of the surrogate model in the local optimum area) during optimization. An alternative method, space mapping [25], maps the input/output space of a low-fidelity model to the input/output space of the high-fidelity model. These methods can significantly improve the optimization efficiency when physically-based and computational efficient low-fidelity models are available.

6 Summary

This work presents the applications of surrogate modeling in variation-aware circuit macromodeling and design analysis. Surrogate modeling can facilitate the design exploration and optimization with variation-aware performance models. Also, surrogate modeling can be used to enhance the accuracy and scalability of IO macromodels. Moreover, the surrogate model-based method is able to generate device models with critical variability parameters. The surrogate-based method greatly reduces the complexities and costs of variation-aware macromodeling and circuit design.

Acknowledgements. This material is based up on the work supported by the Self-HEALing mixed signal Integrated Circuits (HEALICs) program of the Department of Defense Advanced Research Projects Agency (DARPA) and AFRL under contract number: FA8650-09-C-7925. Approved for Public Release. Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The authors T. Zhu, Dr. P. D. Franzon, and Dr. M. B. Steer would like to thank Dr. T. Dhaene of Ghent University, Belgium, for providing the SUrrogate MOdeling (SUMO) Toolbox and for helpful discussion.

References

1. Gorissen, D., Turck, F.D., Dhaene, T.: Evolutionary Model Type Selection for Global Surrogate Modeling. Journal of Machine Learning Research 10(1), 2039–2078 (2009)
2. Eldred, M.S., Dunlavy, D.M.: Formulations for Surrogate-based Optimization with Data Fit, Multifidelity, and Reduced-order models. In: 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA-2006-7117, Protsmouth, Virginia (2006)
3. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Moldeling: A Practical Guide. John Wiley & Sons (2008)
4. Kleijnen, J.: Design and Analysis of Simulation Experiments. Springer (2008)
5. Crombecq, K., Couckuyt, I., Gorissen, D., Dhaene, T.: Space-Filling Sequential Design Strategies for Adaptive Surrogate Modelling. In: 1st International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, Paper 50, Civil-Comp Press, Stirlingshire (2009)
6. Meckesheimer, M., Booker, A.J., Barton, R., Simpson, T.: Computationally Inexpensive Metamodel Assessment Strategies. AIAA Journal 40(10), 2053–2060 (2002)

7. Gorissen, D., Crombecq, K., Couckuyt, I., Dhaene, T., Demeester, P.: A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design. Journal of Machine Learning Research 11, 2051–2055 (2010)
8. Jeon, S., Wang, Y., Wang, H., Bohn, F., Natarajan, A., Babakhani, A., Hajimiri, A.: A Scalable 6-to-18 GHz Concurrent Dual-Band Quad-Beam Phased-Array Receiver in CMOS. IEEE Journal of Solid-State Circuits 43(12), 2660–2673 (2008)
9. IO Buffer Information Specification, http://www.eigroup.org/ibis/ibis.htm
10. Zhu, T., Steer, M.B., Franzon, P.D.: Accurate and Scalable IO Buffer Macromodel Based on Surrogate Modeling. IEEE Transactions on Components, Packaging and Manufacturing Technology 1(8), 1240–1249 (2011)
11. IBIS Modeling Cookbook, http://www.vhdl.org/pub/ibis/cookbook/cookbook-v4.pdf.
12. Muranyi, A.: Accuracy of IBIS models with reactive loads, http://www.eda.org/pub/ibis/summits/feb06/muranyi2.pdf
13. LaBonte, M., Muranyi, A.: IBIS Advanced Technology Modeling Task Group Work-achievement: Verilog-A element library HSPICE test, http://www.vhdl.org/pub/ibis/macromodel_wip/archive-date.html
14. Varma, A., Glaser, A., Lipa, S., Steer, M.B., Franzon, P.D.: The Development of A Macro-modeling Tool to Develop IBIS Models. In: 12th Tropical Meeting on Electrical Performance Electronic Packaging, pp. 277–280. Princeton, New Jersey (2003)
15. Varma, A.K., Steer, M.B., Franzon, P.D.: Improving Behavioral IO buffer modeling based on IBIS. IEEE Transactions on Advanced Packaging 31(4), 711–721 (2008)
16. Orshansky, M., Nassif, S.R., Boning, D.: Design for Manufacturability and Statistical Design: A Constructive Approach. Springer, New York (2008)
17. Saha, S.K.: Modeling process variability in scaled CMOS technology. IEEE Design and Test of Computers 27(2), 8–16 (2010)
18. Yelten, M.B., Franzon, P.D., Steer, M.B.: Surrogate Model-based Analysis of Analog Circuits—Part I: Variability Analysis. IEEE Transactions on Device and Material Reliability 11(3), 458–465 (2011)
19. Morshed, T.H., Yang, W., Dunga, M.V., Xi, X., He, J., Liu, W., Yu, K., Cao, M., Jin, X., Ou, J.J., Chan, M., Niknejad, A.M., Hu, C.: BSIM4.6.4 MOSFET model- User's manual (April 2009), http://www-device.eecs.berkeley.edu/~bsim3/BSIM4/BSIM470/BSIM470_Manual.pdf
20. Lophaven, S.N., Nielsen, H.B., Sondergaard, J.: A MATLAB Kriging toolbox 2.0 (August 2002), http://www2.imm.dtu.dk/?hbn/dace/dace.pdf
21. Yelten, M.B., Gard, K.G.: A Novel Design Methodology for Tunable of Low Noise Amplifiers. In: Wireless and Microwave Conference (WAMICON 2009), Florida, USA, pp. 1–5 (2009)
22. Semiconductor Industry Association, International Technology Roadmap for Semiconductors (ITRS)
23. Tiwary, S.K., Tiwary, P.K., Rutenbar, R.A.: Generation of Yield-aware Pareto Surfaces for Hierarchical Circuit Design Space Exploration. In: 43rd ACM/IEEE Design Automation Conference, San Francisco, CA, U.S.A. (2006)
24. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P.K.: Surrogate-based Analysis and Optimization. Progress in Aerospace Sciences 41, 1–28 (2005)
25. Koziel, S., Cheng, Q.S., Bandler, J.W.: Space mapping. IEEE Microwave Magazine 9(6), 105–122 (2008)

Analysis and Optimization of Bevel Gear Cutting Processes by Means of Manufacturing Simulation

Christian Brecher[1], Fritz Klocke[2], Markus Brumm[1], and Ario Hardjosuwito[1]

[1] Chair of Machine Tools, Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Steinbachstr. 19, D-52074 Aachen, Germany
[2] Chair of Manufacturing Technology, Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Steinbachstr. 19, D-52074 Aachen, Germany
{C.Brecher,M.Brumm,A.Hardjosuwito}@wzl.rwth-aachen.de, F.Klocke@wzl.rwth-aachen.de

Abstract. When torque and speed are transmitted from a non-parallel orthogonal axle to another axle bevel gears are often used. They are applied e.g. in drive trains of automotive, ships and helicopters. The manufacturing of this gear type is often difficult due to unpredictable tool wear during the bevel gear cutting process. This leads e.g. to unexpected production stops for tool changes which results in undesired additional manufacturing costs. Currently it is not possible to analyze the bevel gear cutting process sufficiently, because of its complexity. Thus, cutting process design happens iteratively in order to find the optimum process parameters regarding high productivity and less tool wear. This issue leads to the demand for exact knowledge of the tool wear behavior. Due to this a manufacturing simulation for bevel gear cutting has been developed. During the simulation a modeling of the tool, workpiece and kinematics is performed as well as a geometrical penetration calculation. With this simulation approach different characteristic values for a detailed analysis are calculated. Within this paper the analysis and optimization of bevel gear cutting processes regarding tool wear are presented. Here, the calculated results from simulation have been compared to tool wear from experimental cutting trials.

Keywords: Gear Cutting Simulation, Manufacturing Simulation, Bevel Gears, Tool Wear Analysis.

1 Introduction

In general the transmission of torque or speed from one axle to a non-parallel orthogonal axle is realized by bevel gears. Bevel gears are used in many applications like in marine, in rear axles of automotive, in helicopters and industrial drives. The standard manufacturing of bevel gears is performed in a complex Computerized Numerical Control (CNC) cutting process. Due to unpredictable tool wear and sudden failure of the cutting tools, unexpected production stops for tool changes occur and lead to a loss of productivity, low process stability and finally to additional manufacturing costs. Thus, the productivity of the machining process depends significantly on the

N. Pina et al. (Eds.): *Simulation & Modeling Methodologies, Technologies & Appl.*, AISC 197, pp. 271–284.
DOI: 10.1007/978-3-642-34336-0_18

tool wear and tool life, see [1]. Currently it is not possible to analyze the bevel gear cutting process sufficiently regarding tool wear and tool life, because of its complexity. Hence, the design of the cutting process happens iteratively in order to find the optimal process parameters. So an exact knowledge about the tool wear and tool life behavior is necessary to optimize the cutting process. These challenges and issues are presented in figure 1. In order to provide a simulation tool for analyzing and optimizing the bevel gear cutting process a manufacturing simulation has been developed at WZL, as presented by [2-3].

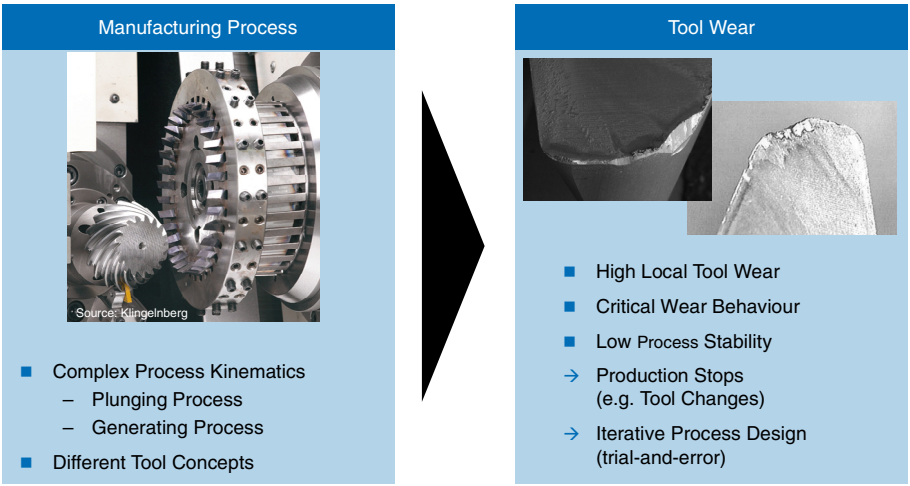


Fig. 1. Challenges in bevel gear cutting

2 Modeling and Simulation Method

Within the manufacturing simulation a geometrical penetration calculation is conducted as described by [2-4]. Before starting the calculation the kinematics , workpiece and tool have to be modeled as described below.

2.1 Process Kinematics

All axis movements, like in the real cutting process, can be considered in the simulation. In figure 2 the axis movements and positions of the bevel gear cutting machine are presented. More precisely, the variables and the resulting movements are: mccp (the machine center to cross axis point), α (cradle angle), β (modified roll by workpiece rotation), γ (angular Motion), φ (radial motion), ε (horizontal Motion), η (vertical motion), χ (helical motion), ω (tool rotation). All axis movements can be described as a series expansion, which is truncated after the sixth order, regarding the mean tool angle ω_m and the mean cradle angle α_m.

For example the axis movement for the depth position χ of the cutting tool , so called helical motion, is calculated by formula 1. The variables a_χ to u_χ represent the

coefficients. E.g. the coefficient a_χ is a time invariant value therefore a positioning of the cutter, whereas u_χ is linearly time-dependent:

$$\chi(\alpha,\omega) = a_\chi + b_\chi\cdot(\alpha-\alpha_m) + c_\chi\cdot(\alpha-\alpha_m)^2 + d_\chi\cdot(\alpha-\alpha_m)^3 + e_\chi\cdot(\alpha-\alpha_m)^4 + f_\chi\cdot(\alpha-\alpha_m)^5 + g_\chi\cdot(\alpha-\alpha_m)^6 + p_\chi\cdot(\omega-\omega_m) + q_\chi\cdot(\omega-\omega_m)^2 + r_\chi\cdot(\omega-\omega_m)^3 + s_\chi\cdot(\omega-\omega_m)^4 + t_\chi\cdot(\omega-\omega_m)^5 + u_\chi\cdot(\omega-\omega_m)^6 \quad (1)$$

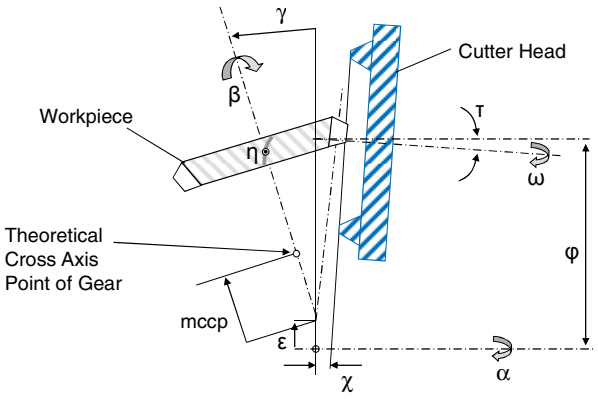


Fig. 2. Scheme and axis movements of a bevel gear cutting machine

2.2 Modeling of Workpiece and Tool

In the simulation the workpiece and the tool envelope are modeled as 3D clouds of scattered points. With these points a mesh of triangles is generated for the workpiece and the tool. The modeling of the workpiece can be described in three steps, see figure 3. At first the cross section of the gear flank is defined by four points. With these points the gear width b, the toe and the heel of the bevel gear are defined.

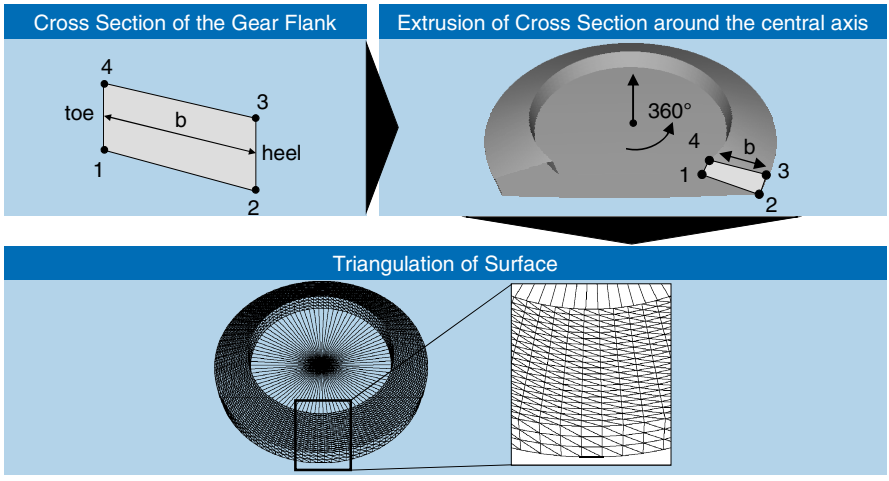


Fig. 3. Modeling and triangulation of workpiece

The heel is the face with the largest diameter respectively the largest distance to the central axis. The toe represents the face with the smallest diameter. In order to get a solid 3D body the cross section is revolved around the central axis by 360°. Finally this extruded body is getting triangulated as described by [3].

The tool envelope is modeled by an extrusion of the tool profile depending on the process kinematics, see figure 4. Like the workpiece, the tool is meshed by triangulation as well.

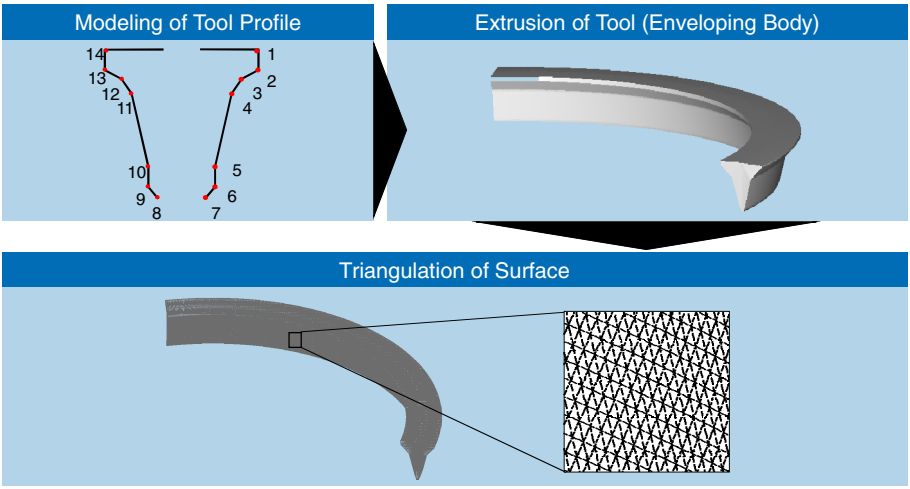


Fig. 4. Modeling and triangulation of tool

The data for workpiece, tool and process kinematics can be imported in the software from an ASCII file in the neutral data format. This data format is a common one in gear industry and developed by [5].

2.3 Simulation Method

When the modeling is finished a geometrical penetration calculation is conducted within the manufacturing simulation. During this calculation the 3D bodies of workpiece and tool envelope penetrate each other in compliance with the kinematics. The penetration is realized by ray-tracing as described by [6]. The calculated penetration volume can be interpreted as the undeformed chip geometry resulting from the cutting process, see figure 5. With this undeformed geometry different characteristic values, like the chip thickness h_{cu}, can be calculated. It has to be mentioned that the penetration calculation is a geometrical calculation, i.e. that no plastic deformations and thermal effects are considered.

In order to accelerate the penetration calculation bounding-boxes and Binary Space Partitioning (BSP) trees are used as depicted from [6]. An additional approach to accelerate the simulation is the calculation of the penetration by General-purpose Computing on Graphics Processing Units (GPGPU) as published by [7] where OpenCL, CUDA, PGI Accelerator are used in combination with graphics processing units (GPU) in order to increase the performance.

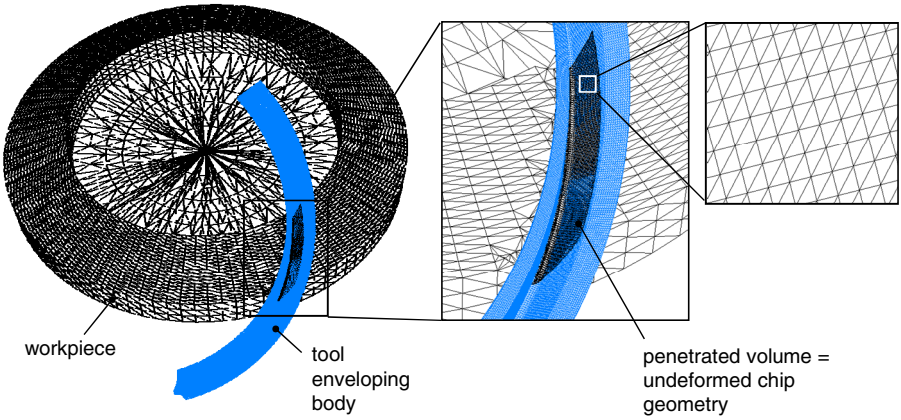


Fig. 5. 3D penetration calculation of triangulated surfaces

3 Calculated Characteristic Values for Tool Wear Analysis

3.1 Current Characteristic Values for Process Analysis

Chip thickness h_{cu} and the working length l_e are characteristic values, besides the cutting forces, for description and analyzing the bevel gear cutting process according to [3-4], see figure 6. The chip thickness represents the thickness of the undeformed chip at a certain point on the cutting edge. The working length represents the cutting path i.e. the contact length between the tool and the workpiece during the cutting.

Investigations from [3-4] and [8] show that also the characteristic values like the working rake angle γ_e and relief angle α_e have a significant influence on the tool wear. Especially the flank wear of the tool is not only influenced by the chip thickness h_{cu} and the working length l_e, but also by the geometry of the cutting edge. The cutter geometry inter alia is determined by the rake and relief angle of the cutter. These angles are defined in the German Standard DIN 6580 and DIN 6581 [9-10].

As a conclusion of the state of the art investigations it can be stated that there is partly the possibility to analyze the tool wear by means of the current characteristic values. But there is no sufficient approach yet for a prediction of tool wear in bevel gear cutting.

In industrial application there are different tool concepts. One of the concepts is the alternating half profile blades. They are separated into outside and inside blades, see figure 7. Both blades are defined as a blade group. One blade group cuts the contour of one gap including gear root and flanks. The outside blade cuts the concave flank and the inside blade cuts the convex flank of the gear gap. From this typical two-flank chips are cut. The removed material on the flanks respectively the chip thickness $h_{cu,flank}$ depends inter alia on the chip thickness on the tip $h_{cu,tip}$ and the pressure angle δ_{OB} or δ_{IB} of the tools:

$$h_{cu,flank} = h_{cu,tip} \cdot \sin(\delta) \tag{2}$$

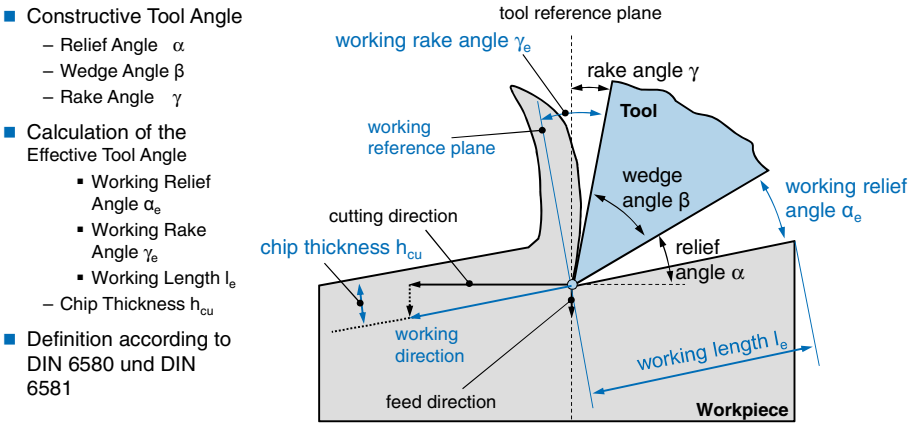


Fig. 6. Calculable characteristic values

Besides the concept of alternating blades there is the concept of full profile blades, where one blade group consists of only one full profile blade. This type of blades has a theoretical rake angle of $\gamma = 0°$. The full profile blades cut both flanks (inside and outside) at the same time. Thus, typical three-flank chips are cut, as shown in figure 7, below right. The advantage of this concept is the increased productivity due to the possible increase of the number of blade groups in the cutter head.

The chip thickness h_{cu} is the most common characteristic value for the analysis of the cutting process according to [3]. A higher pressure angle δ results in a higher chip thickness on the flank. The chip thickness itself depends on the feed velocity per blade group.

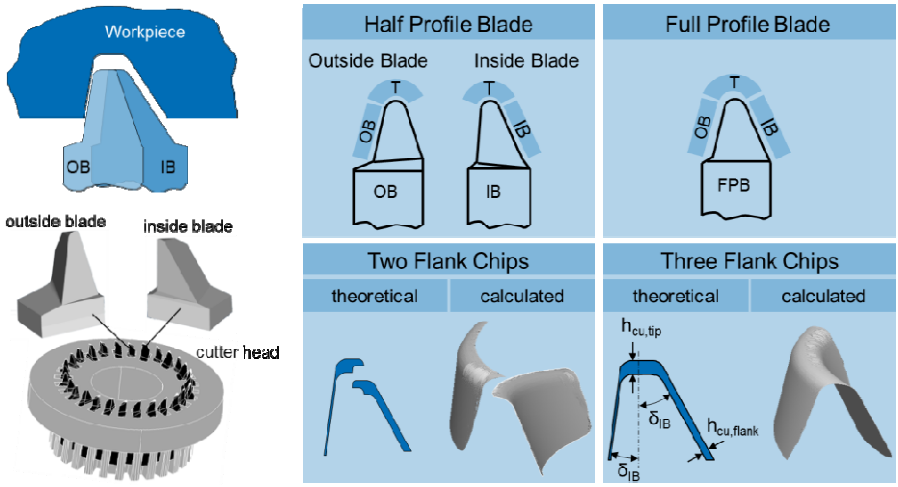


Fig. 7. Tool concepts and cutter separation of blades

3.2 New Approaches of Characteristic Values for Tool Wear Prediction

Within the investigations of [8] regarding tool wear in bevel gear cutting, it turned out that especially the corner radius is critical regarding tool wear like chipping. This is caused by the multi-flank chip-formation at the corner radius. Here the material of the chip is compressed and squeezed, see figure 8. Thus, a simple analysis of the chip thickness in this area of the tool is not sufficient.

From the figure it becomes clear that the chip thickness h_{cu} is varying along the cutting edge l_s. Especially in the transition area of the corner radius between the flank (IB) and the tip (T) a gradient of the chip thickness $\Delta h_{cu}/\Delta l_s$ is visible.

The spatial compression of the chip is determined by the pressure angle of the tool, as [11] presented. In order to consider the compression and squeezing of the chip in the geometrical penetration calculation the gradient of the chip thickness can be used. The gradient represents the varying chip thickness along the cutting edge l_s. At the tool flank the gradient is zero due to the invariant chip thickness. This is plausible, because of the not existing compression of the chip material referred to the rake face respectively the working reference plane, see [10].

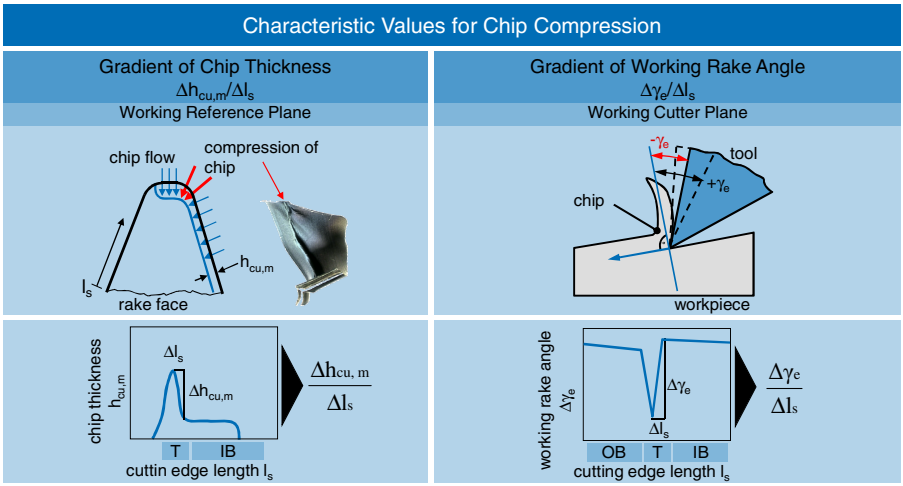


Fig. 8. Description of mechanical and thermal load due to the chip compression by geometrical characteristics

A higher compression of the chip results in a higher thermal and mechanical load at the cutting edge and the risk of tool wear. Due to the spatial chip formation and compression a consideration of only the working reference plane is not sufficient. Even the working cutter plane in which the chip flows orthogonal to the rake face has to be considered according to DIN 6581 [10]. Both planes take the working direction of the cutter into account. For the description of the chip compression in the working cutter plane the gradient of working rake angle $\Delta\gamma_e/\Delta l_s$ can be used, see figure 8. This characteristic value represents the varying rake angle along the cutting edge and thus the varying chip formation and compression. A rapidly changing gradient in a small

area of the cutting edge corresponds with a changing chip formation and a varying load during the cutting. This varying load has negative influence on the tool wear behavior. Hence, the gradient should have a minimum value.

Besides the aspect of chip compression and the so caused tool load further characteristic values for analyzing the cutting process can be used. The entire working length l_e describes the contact length, which the cutter is in contact with the workpiece under consideration of the working direction, see figure 9. So the working length is a geometrical approach for the description of the thermal and mechanical load on the cutting edge. A higher working length results in a higher temperature respectively friction and thus in higher loads on the cutting edge.

Additionally, the working relief angle α_e can be used for the analysis of the tool load. The working relief angle influences the thermal stress on the cutting edge. Thus the gradient $\Delta\alpha_e/\Delta l_s$ is a useful characteristic value for the alternating thermal stress along the cutting edge. The higher the gradient is the higher will be the alternating thermal stresses on the relief face of the cutter. Hence, a minimum value of the gradient is desirable. In order to integrate the presented characteristic values in only one value K_G is introduced:

$$K_G = l_e \cdot \frac{\Delta h_{cu}}{\Delta l_s} \cdot \frac{\Delta \alpha_e}{\Delta l_s} \cdot \frac{\Delta \gamma_e}{\Delta l_s} \tag{3}$$

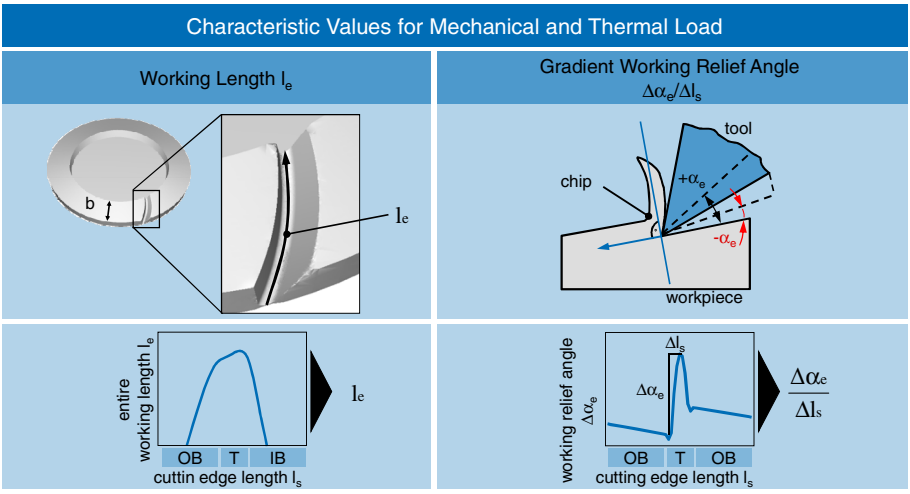


Fig. 9. Description of mechanical and thermal load by geometrical characteristic values

A high K_G value results in a higher tool load and thus in higher tool wear. With the new characteristic value K_G a first qualitative comparison of different processes and so of the tool wear is possible. From this an optimization of the process is feasible and a first approach for a qualitative tool wear prediction model is realized.

In the future the characteristic value with the tool wear model has to be modified by weighting of the coefficients in order to enable a quantitative evaluation and prediction of the tool wear.

4 Tool Wear Analysis

In general pinion and ring gear are manufactured by using cutter heads which are equipped with stick-type blades. Here the face-milling process, as described by [5] is used for manufacturing. The plunging process is mostly used for the manufacturing of the ring gear whereas the generating process is used for pinion manufacturing. Following different examples regarding tool wear of stick type blades in industrial application will be presented. Here the tool wear behavior in plunging and generating process with different tool concepts is compared to the new characteristic value K_G. In this report the focus is set on the discontinuous face milling process with carbide tools in dry cutting.

4.1 Tool Wear Analysis of Plunging Process

In figure 10 the tool wear of two full profile blades is presented. They are used for plunging process 1 and 2 where the ring gear manufacturing is realized. In process 1 a cutting velocity of v_c = 200 m/min and a ramp with a feed of f_{BG} = 0.15 – 0.06 mm per blade group was used. The cutter head with an outer diameter D_a = 231 mm was equipped with 7 carbide tools. The characteristic value K_G was displayed in the diagram over the unrolled cutting edge length l_s which is separated into the outside blade (OB), the tip area (T) and the inside blade (IB). The maximum tool wear occurs at the corner radius of the outside blade (OB) to the tip area (T). In the simulation the characteristic value K_G has its maximum at the same tool area. Additional tool wear occurs at the corner radius of the inside blade (IB) and the tip area (T). In this area the tool wear is less than at the other corner radius. The same tendencies are calculated in the simulation. Thus, the good correlation of the calculated value K_G and the real tool wear is evident. On the one hand the maximum tool wear can be located by K_G, on the other hand the lower tool wear at the inside blade can also be calculated.

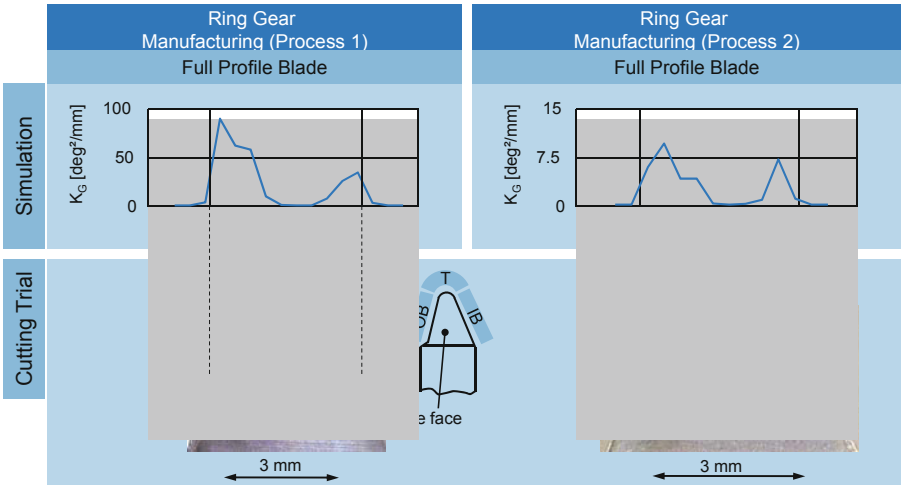


Fig. 10. Tool wear analysis of full profile blades

In process 2, see figure 10 right, a cutting velocity of v_c = 150 m/min and a feed ramp of f_{BG} = 0.16 – 0.10 mm per blade group was chosen. The cutter head with an outer diameter D_a = 165 mm was equipped with 14 carbide tools. Here the correlation between the tool wear from cutting trial and the simulation is also good. The maximum tool wear occurred at the corner radius of the outside blade (OB) and the tip (T). Even the tool wear in the corner radius of the inside blade (IB) can be determined by the simulation. In this example the tool wear is similar in both corner radii compared to process 1, where the amount of tool wear is very uneven. This tool wear behavior is predictable with the simulation by means of the characteristic value K_G.

In addition to full profile blades there is the concept of alternating half profile blades. In order to show the good correlation between the cutting trial results and the simulation results the tool wear and the characteristic value K_G are presented for this process, too,see figure 11.

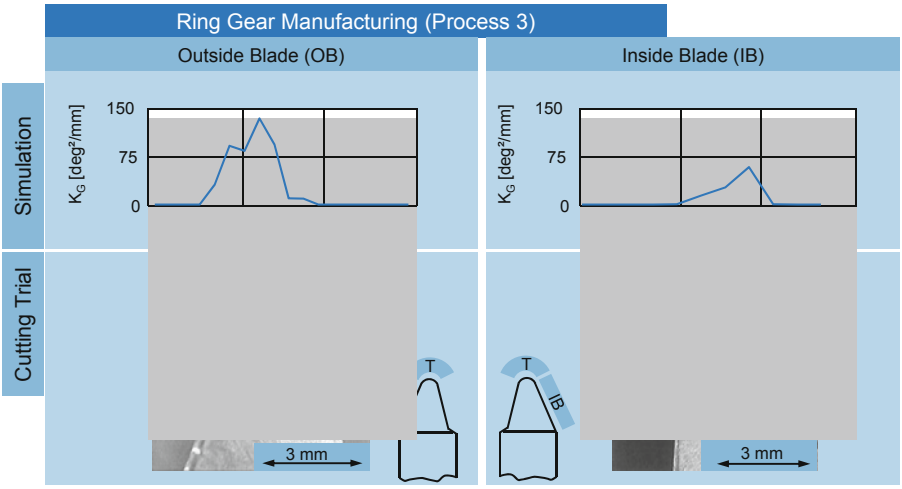


Fig. 11. Tool wear analysis of half profile blades

Here (process 3) the focus is not only the localization of the maximum tool wear at the cutting edge but also the identification of the most critical blade regarding tool wear. The maximum value for K_G was calculated at the corner radius of the outside blade (OB). This correlates well with the occurred tool wear from the cutting trials. Even the lower tool wear of the inside blade (IB) was calculated correctly.

4.2 Tool Wear Analysis of Generating Process

Generally pinions are manufactured by a generating process. The tool and the workpiece movement are coupled depending on the process kinematics. In process 4 a feed ramp of v_w = 8.72 - 12.4 °/s and a cutting velocity of v_c = 230 m/min have been used. The outer diameter of the cutter head has been D_a = 268 mm and it was equipped with 32 carbide full profile blades. In this process a generating from heel to toe is

conducted. In figure 12 the occurred tool wear on the full profile blade is shown. It is visible that the maximum tool wear is located at the corner radius of the outside blade (OB). Here a chipping is observed. The tool wear at the inside blade is about 50% of the maximum wear. Former investigations contain the tool wear analysis of this process, but did not show sufficient results. Here the tool wear characteristic value K_G is used for the first time to identify the tool with the highest amount of tool wear. A comparison of the real tool wear in generating process with the calculated characteristic value K_G is presented in figure 12. The calculation results correlate well with the tool wear from the cutting trials. In addition not only the location of the tool wear correlates well with the characteristic value K_G but also the amount of tool wear.

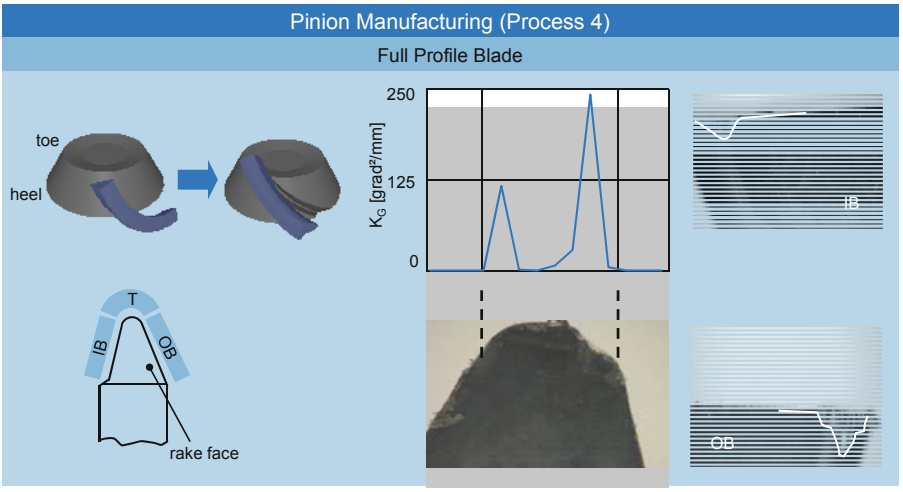


Fig. 12. Tool wear analysis of pinion manufacturing

The presented results show that now it is possible to analyze the occurring tool wear with only one characteristic value. Thus, it is not necessary anymore to analyze the cutting process regarding tool wear by applying and analyzing many different characteristic values, like the chip thickness or the working tool angles.

4.3 Identification of Optimal Tool Concept

Within this report different tool concepts have been presented. Process 1 (full profile blades) and 3 (half profile blades) differ in the tool concept. The manufactured geometry of the ring gear and the productivity of the process is identical. This means e.g. that the cutting and feed velocity is the same and the number of blade groups of the full profile concept is reduced to 50%. This reduction is possible, because the number of active cutting edges is the same for 14 half profile blades and 7 full profile blades.

Now it is interesting to know which concept is the best for the presented application. A comparison of process 1 (full profile blades) and process 3 (half profile blades) by means of the characteristic value K_G is done, see figure 13.

The comparison of the calculated characteristic value K_G for the two processes shows that the maximum tool wear appears at the outside blade (OB) of the half profile blade concept. The value of K_G at the outside blade has approximately the double magnitude of the value of the full profile blade. The tool life of the full profile blade was to L = 500 workpieces whereas the tool life of the half profile blades was to L = 400 workpieces. Thus, there is a good correlation between the characteristic value K_G and the tool wear but there is also a good correlation between K_G and the tool life of the different tool concepts.

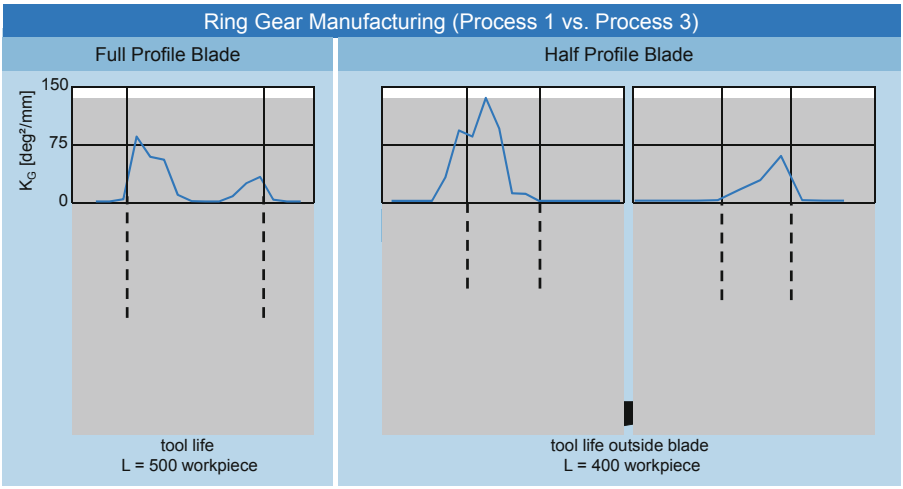


Fig. 13. Comparison of different tool concepts

It can be stated that the manufacturing simulation including the calculation of the new characteristic value K_G allows for the first time analyzing the bevel gear cutting process regarding the expected tool wear. Now a qualitative analysis and prediction of tool life is possible.

5 Conclusions

Within this report the manufacturing simulation for bevel gear cutting is presented. At first the modeling of the workpiece and the tool is realized. Under consideration of the process kinematics the simulation can be conducted. Within the manufacturing simulation a 3D penetration calculation of workpiece and tool is carried out. From the penetrated volume the undeformed chip geometry can be calculated. With information from this penetrated volume different characteristic value like the chip thickness can be derived. With these values a first analysis of the cutting process regarding tool loads and wear is possible. Unfortunately, there is often no correlation between these singular values and the expected tool wear.

Currently a new characteristic value for the tool wear analysis is developed and implemented in the manufacturing simulation. This new value includes the gradient of

different calculated characteristic values over the cutting edge like the gradient of the chip thickness $\Delta h_{cu}/\Delta l_s$. This gradient, for example, represents the compression and squeezing of the chip over the cutting edge. Thus, this value can be used for the analysis of the tool load at the cutting edge.

The comparison of the calculated new characteristic value and the tool wear from cutting trials show good correlations. The localization of tool wear as well as a qualitative comparison of different processes regarding the expected tool life is possible. E.g. the tool life behavior of full profile blades and half profile blades correlates well with the simulation results.

In the future the manufacturing simulation has to be enhanced in order to calculate the expected tool life for bevel gear cutting. This, for instance, can be applied for increasing the productivity of the cutting process and for optimizing the process design regarding tool changes. Thus, the development of a tool life model for the bevel gear cutting process has to be realized. In a first step analogy trials in single flank cutting on a lathe will be performed, see figure 14.

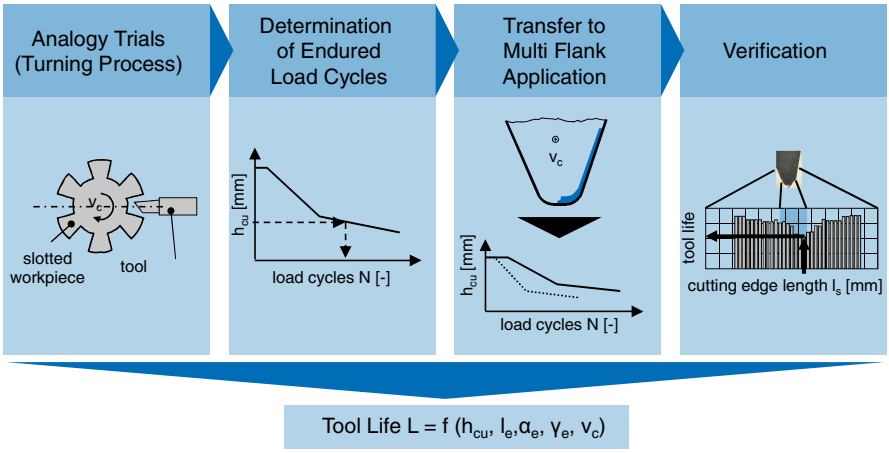


Fig. 14. Methodology for prediction of tool life in bevel gear cutting

Here the influence of chip thickness h_{cu}, working length l_e, working relief angle α_e, working rake angle γ_e, cutting velocity v_c and tool coating will be investigated. Afterwards a determination of the endured load cycles over cutting edge length will be conducted in step 2. With the load cycles a hands-on visualization will be realized. A transfer to multi flank application will be done in step 3. Here the presented characteristic value K_G will be used to consider the additional load at the corner radius of the tool due to multi flank chips. Finally industrial cutting trials will be performed in order to optimize and verify the tool life model. Further the tool life model will be implemented in the manufacturing simulation.

Acknowledgements. The authors would like to thank the German Research Foundation (DFG) for supporting the presented works, which are part of the research project "Methodik zur Vorhersage der Werkzeugstandzeit bei Fertigungsprozessen mit Mehrflankenspänen am Beispiel des Kegelradfräsens".

References

1. Chavoshi, S.: Tool flank wear prediction in CNC turning of 7075 AL alloy SiC composite. Production Engineering – Research and Development 5(1), 37–47 (2011)
2. Brecher, C., Klocke, F., Gorgels, C., Hardjosuwito, A.: Simulation Based Tool Wear Analysis in Bevel Gear Cutting. In: International Conference on Gears, pp. 1381–1384. VDI, Düsseldorf (2010)
3. Rütjes, U.: Entwicklung eines Simulationssystems zur Analyse des Kegelradfräsens. Dissertation RWTH Aachen University (2010)
4. Klocke, F., Brecher, C., Gorgels, C., Hardjosuwito, A.: Modellierung und Simulationen zum Kegelradfräsen – Analyse der Werkzeugbelastung. In: KT2009 – 7. Gemeinsames Kolloquium Konstruktionstechnik, pp. 187–194. Bayreuth (2009)
5. Klingelnberg, J.: Kegelräder – Grundlagen und Anwend-ungen. Springer, Heidelberg (2008)
6. Akenine-Möller, T., Haines, E.: Real-time Rendering, 2nd edn. AK Peters, Massachusetts (2002)
7. Wienke, S., Plotnikov, D., an Mey, D., Bischof, C., Hardjosuwito, A., Gorgels, C., Brecher, C.: Simulation of bevel gear cutting with GPGPUs—performance and productivity. In: International Supercomputing Conference, Special Issue Paper. Springer, Berlin
8. Klein, A.: Spiral Bevel and Hypoid Gear Tooth Cutting with Coated Carbide Tools. Dissertation RWTH Aachen University (2007)
9. N.N.: Bewegungen und Geometrie des Zerspanvorgangs. German Standard DIN 6580 (1985)
10. N.N.: Bezugssysteme und Winkel am Schneidteil des Werkzeuges. German Standard DIN 6581 (1985)
11. Klocke, F., Gorgels, C., Herzhoff, S., Hardjosuwito, A.: Simulation of Bevel Gear Cutting. In: 3rd WZL Gear Conference, Boulder, USA (2010)

Author Index