SIMUL 2014

The Sixth International Conference on Advances in System Simulation

ISBN: 978-1-61208-371-1

October 12 - 16, 2014

Nice, France

SIMUL 2014 Editors

Amr Arisha, College of Business, DIT, Ireland

Georgiy Bobashev, RTI International -Research Triangle Park, USA

SIMUL 2014

Forward

The Sixth International Conference on Advances in System Simulation (SIMUL 2014), held between October 12 - 16, 2014 in Nice, France, continued a series of events focusing on advances in simulation techniques and systems providing new simulation capabilities.

While different simulation events are already scheduled for years, SIMUL 2014 identified specific needs for ontology of models, mechanisms, and methodologies in order to make the selection of an appropriate tool easier. With the advent of Web Services and WEB 3.0 social simulation and human-in simulations, new challenging situations arise along with more classical process simulations and distributed and parallel simulations. The event was also intended as an update on simulation tools considering these new simulation flavors.

The conference had the following tracks:

- Simulation tools and platforms
- Transport simulation
- Simulation models
- Human-in simulation
- Building simulation
- Model-based system prediction
- Practical applications on process simulations
- Simulation methodologies

Similar to the previous edition, this event continued to be very competitive in its selection process and very well perceived by the international community interested in system simulation. As such, it attracted excellent contributions and active participation from all over the world. We were very pleased to receive a large amount of top quality contributions.

We take here the opportunity to warmly thank all the members of the SIMUL 2014 technical program committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to SIMUL 2014. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the SIMUL 2014 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope SIMUL 2014 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in system simulation research. We also hope that Nice, France provided a pleasant environment during the conference and everyone saved some time to enjoy the charm of the city.

SIMUL 2014 Chairs

SIMUL Advisory Chairs

Edward Williams, PMC-Dearborn, USA
Paul Fishwick, University of Florida-Gainesville, USA
Christoph Reinhart, Harvard University - Cambridge, USA
Amr Arisha, College of Business, DIT, Ireland

SIMUL 2014 Research Liaison Chairs

Tae-Eog Lee, KAIST, Korea
Marko Jaakola, VTT Technical Research Centre of Finland, Finland

SIMUL 2014 Industry Liaison Chairs

Diglio A. Simoni, RTI International – RTP, USA
Shengnan Wu, American Airlines, USA
Ann Dunkin, Palo Alto Unified School District, USA
Tejas R. Gandhi, Virtua Health-Marlton, USA

SIMUL 2014 Special Area Chairs

Model-based system prediction
Georgiy Bobashev, RTI International -Research Triangle Park, USA
Aida Omerovic, SINTEF & University of Oslo, Norway

Process simulation
Ian Flood, University of Florida, USA
Gregor Papa, Jozef Stefan Institute - Ljubljana, Slovenia

SIMUL 2014 Publicity Chairs

Nuno Melao, Catholic University of Portugal - Viseu, Portugal

SIMUL 2014

Committee

SIMUL Advisory Chairs

Edward Williams, PMC-Dearborn, USA
Paul Fishwick, University of Florida-Gainesville, USA
Christoph Reinhart, Harvard University - Cambridge, USA
Amr Arisha, College of Business, DIT, Ireland

SIMUL 2014 Research Liaison Chairs

Tae-Eog Lee, KAIST, Korea
Marko Jaakola, VTT Technical Research Centre of Finland, Finland

SIMUL 2014 Industry Liaison Chairs

Diglio A. Simoni, RTI International – RTP, USA
Shengnan Wu, American Airlines, USA
Ann Dunkin, Palo Alto Unified School District, USA
Tejas R. Gandhi, Virtua Health-Marlton, USA

SIMUL 2014 Special Area Chairs

Model-based system prediction
Georgiy Bobashev, RTI International -Research Triangle Park, USA
Aida Omerovic, SINTEF & University of Oslo, Norway

Process simulation
Ian Flood, University of Florida, USA
Gregor Papa, Jozef Stefan Institute - Ljubljana, Slovenia

SIMUL 2014 Publicity Chairs

Nuno Melao, Catholic University of Portugal - Viseu, Portugal

SIMUL 2014 Technical Program Committee

Erika Abraham, RWTH Aachen University, Germany
Dimosthenis Anagnostopoulos, Harokopio University of Athens, Greece
Chrissanthi Angeli, Technological Institute of Piraeus - Athens, Greece

Amr Arisha, College of Business - DIT, Ireland
Joseph Barjis, Delft University of Technology, Netherlands
Marek Bauer, Politechnika Krakowska, Poland
Ateet Bhalla, Oriental Institute of Science and Technology, India
Keith Bisset, Virginia Tech, USA
Georgiy Bobashev, RTI International -Research Triangle Park, USA
Christophe Bourdin, Université d'Aix-Marseille, France
Jan F. Broenink, University of Twente, Netherlands
Dilay Celebi, Istanbul Technical University, Turkey
E Jack Chen, BASF Corporation, USA
Soolyeon Cho, North Carolina State University - Raleigh, USA
Franco Cicirelli, Università della Calabria, Italy
Homero H. Contreras, Universidad Popular Autónoma del Estado de Puebla (UPAEP), Mexico
Kendra Cooper, University of Texas at Dallas / University of Calgary, USA / Canada
Duilio Curcio, University of Calabria - Rende (CS), Italy
Andrea D'Ambrogio, University of Roma TorVergata, Italy
Yuya Dan, Matsuyama University, Japan
Saber Darmoul, King Saud University, Saudi Arabia
Jacinto Dávila, Universidad de Los Andes, Venezuela
Luis Antonio de Santa-Eulalia, Université de Sherbrooke, Canada
Luís de Sousa, Public Research Centre Henri Tudor - Luxembourg, Luxembourg
Gabriella Dellino, IMT Institute for Advanced Studies Lucca, Italy
Tom Dhaene, Ghent University - IBBT, Belgium
Jan Dijkstra, Eindhoven University of Technology, Netherlands
Atakan Dogan, Anadolu University, Turkey
Ann Dunkin, Palo Alto Unified School District, USA
Khaled S. El-Kilany, Arab Academy for Science - Alexandria, Egypt
Sabeur Elkosantini, Higher Institute of Computer Science of Mahdia - University of Monatir, Tunisia
Zhou Fang, VR and Immersive Simulation Center - Renault DE-TD, France
Paul Fishwick, The University of Texas at Dallas, USA
Ian Flood, University of Florida, USA
Martin Fraenzle, Carl von Ossietzky Universität Oldenburg, Germany
Terrill L. Frantz, Peking University HSBC Business School, Shenzhen Campus, Guangdong, China
José Manuel Galán, Universidad de Burgos, Spain
Tejas Gandhi, Medical Center of Central Georgia, USA
Genady Grabarnik, St. John's University, USA
Christoph Grimm, TU Kaiserslautern, Germany
Amine Hamri, LSIS Marseille, France
Zhi Han, MathWorks, Inc., USA
Xiaolin Hu, Georgia State University, USA
Michael Hübner, Karlsruhe Institute of Technology, Germany
Mauro Iacono, Seconda Università degli Studi di Napoli, Italy
Segismundo S. Izquierdo, Universidad de Valladolid, Spain

Marko Jaakola, VTT Technical Research Centre of Finland, Finland
Emilio Jiménez Macías, University of La Rioja, Spain
Christina Kluever, University of Duisburg-Essen, Germany
Natallia Kokash, Centrum Wiskunde & Informatica (CWI), Netherlands
Timo Lainema, Turku School of Economics, Finland
SangHyun Lee, University of Michigan, USA
Fedor Lehocki, National Centre of Telemedicine Services / Slovak University of Technology in Bratislava, Slovakia
Jennie Lioris, CERMICS, France
Francesco Longo, University of Calabria, Italy
Jose Machado, Universidade do Minho, Portugal
Ricardo Marcelín-Jiménez, Universidad Autónoma Metropolitana, Mexico
João Pedro Jorge Marques, University of Porto, Portugal
Goreti Marreiros, Engineering Institute - Polytechnic of Porto, Portugal
Stefano Marrone, Seconda Università di Napoli, Italy
Don McNickle, University of Canterbury - Christchurch, New Zealand
Nuno Melao, Catholic University of Portugal - Viseu, Portugal
Jürgen Melzner, Bauhaus-University Weimar, Germany
Marco Mevius, HTWG Konstanz, Germany
Lars Mönch, University of Hagen, Germany
Muaz Niazi, Bahria University, Pakistan
Libero Nigro, Università della Calabria, Italy
Mara Nikolaidou, Harokopio University of Athens, Greece
Aida Omerovic, SINTEF ICT, Norway
Gregor Papa, Jozef Stefan Institute - Ljubljana, Slovenia
Laurent Perochon, VetaGro Sup, France
Claudine Picaronny, LSV ENS Cachan, France
Henri Pierreval, IFMA-LIMOS, France
François Pinet, Irstea, France
Marta Pla-Castells, Universitat de València, Spain
Katalin Popovici, MathWorks Inc., USA
Francesco Quaglia, Sapienza Universita' di Roma, Italy
Urvashi Rathod, Symbiosis Centre for Information Technology, India
Cláudia Ribeiro, INESC-ID Lisbon, Portugal
José Luis Risco Martín, Universidad Complutense de Madrid, Spain
Agostinho Rosa, Technical University of Lisbon, Portugal
Rosaldo J. F. Rossetti, University of Porto, Portugal
Hendrik Rothe, Helmut Schmidt Universitat, Germany
Manuel Filipe Santos, University of Minho, Portugal
Jean-Francois Santucci, University of Corsica, France
Guodong Shao, National Institute of Standards and Technology - Gaithersburg, USA
Larisa Shwartz, IBM T. J. Watson Research Center - Hawthorne, USA
Jeffrey S. Smith, Auburn University, USA
Eric Solano, RTI International, USA

Copyright Information

Table of Contents

Model of Running Time Disturbances for Buses Using Designated Lanes on Approaches to Junctions Equipped with Traffic Signals

Marek Bauer
Department of Transportation Systems
Cracow University of Technology
Cracow, Poland
e-mail: mbauer@pk.edu.pl

***Abstract*—The efficiency of designated bus lanes is very differentiated. This is an effect of the influence of many disturbing factors and the lack of determination and consequence in privileging buses. Problems are mostly seen on the approaches to junctions equipped with traffic signals. In this paper, the factors influencing onto bus running conditions on the designated lanes were classified, as well as the model of bus traffic disturbances at approaches to junctions with traffic signals. The partial elements of the model (e.g., queue clearing time, stopping time in bus lane at approach to junction) were estimated with regression models using. Also, the methods of reducing losses of bus travel time at approaches to junctions were presented. Finally, the micro simulation of bus running time at approaches to junctions with traffic signals in different traffic conditions were executed. Was proved that the reduction of queues in bus lanes may be achieved through shortening the section of the bus lane which is open for vehicles taking a right turn or through the separation of additional, external traffic lane for turning right vehicles, off the bus lane.**

Keywords-bus lane; signalized junction; running time.

I. INTRODUCTION

Designated bus lanes belong to a group of efficient measures designed to privilege public transport vehicles. Agrawal, Goldman and Hannaford [1] described a large set of advantages for the adoption of designated bus lanes. Bus travel conditions are significantly better in dedicated lanes than in lanes used by other vehicles. Dedicated lanes enable attaining higher travel speeds, favor improved punctuality and regular bus runs [2]. Thus, they boost the attractiveness of the public transport, which translates into an increased number of passengers travelling by buses using dedicated traffic lanes. They help in better bus routing (Molecki, [3]) and strongly increase the capacity of the whole public transport systems, as described by Fernandez and Planzer [4]. However, even upon dedicating a traffic lane, the preservation of consistent, beneficial travel conditions for buses is impossible in many cases. Schedule speeds in the aforementioned segments are significantly higher than speeds which would be attained if bus lanes were absent and buses travelled in open-access lanes among other vehicles. It would be expected that average schedule speeds, even in a section so greatly encumbered with public transport vehicle traffic, should at least amount to 20 [km/h]. Therefore, we should ponder upon the causes – specifically, what are the factors disturbing bus traffic in dedicated lanes and how can they be prevented. A simulation approach can be very helpful in solving such problems. Micro simulation models describing bus movement on designated bus lanes were presented in [5], [6] and [7].

In Section 2, we present the factors influencing buses running on the sections with designated bus lanes. In Section 3, we present the model of bus traffic disturbances at approaches to junctions with traffic signals. Sections 4 and 5 explain the results of estimations of queue clearing time and queue motion time. Section 6 presents the methods of reducing bus time losses at approaches to signalized junctions. In Section 7, an example of possible simulation results is presented. Finally, Section 8 presents the conclusion of the paper.

II. DISTURBANCES IN SECTIONS EQUIPPED WITH BUS LANES

Running time of buses was considered in many publications, e.g., by Diab and El-Geneidy [8]. Bus traffic disturbances result from a complex influence of many factors, which more or less affect bus travel times, also in sections with dedicated bus lanes. Many of the factors are of random nature and these factors are greatly responsible for travel disturbances. These are:

- Factors related to traffic in dedicated bus lanes – this is a group of factors with the highest influence on bus travel conditions within sections, junctions and bus stops: the size, the variability and the structure of traffic volumes of buses and other vehicles (taxis and other vehicles authorized to use the lane), the size, the variability and the structure of traffic volumes of vehicles using the lane at the approach to a junction in order to take a right turn.
- Behavioral factors encompassing behaviors of all road users including bus drivers (driving skills, discipline, psychophysical characteristics), caused by the size of the current advancement or delay in reference to the time table – motivating towards a slower or a faster bus driving.

ISBN: 978-1-61208-371-1

- Environmental factors, among which are: season, daytime, day type, atmospheric conditions.

Additionally, disturbances on sections with bus lanes may result from factors of a directly deterministic nature, which themselves are not sources of disturbance; on the contrary, they represent solutions, whose goal was to aid the operation of bus transport. The influence of deterministic factors on bus travel disturbances results, most often, from either insufficient range and consistency in implementing particular infrastructural solutions or their low resistance to intensified activity of random factors. Disturbances may also result from a traffic organization which is insufficiently promoting public transport vehicles and from incorrect planning of services. Deterministic factors may be generally classified as follows:

- Factors related to the infrastructure of designated bus lanes, such as: the width and the length of the section, the scope and the manner of separation from open-access lanes (use of markings or, additionally, physical separation e.g., green lane, concrete or plastic separators).
- Factors related to other elements of the infrastructure of streets with designated bus lanes: the number of traffic lanes in the street section, the amount of junctions with and without traffic signals, the number and location of designated pedestrian crossings.
- Factors related to the infrastructure of junctions and bus stops located within bus lanes, such as: the type of the traffic organization at junctions, the manner of conduct of conflict-point maneuvers, the number of traffic lanes (especially at the approach), the location of the stop (near side or far side the junction area), the occurrence of stop lines and possibly the distance between the line and the bus-stop shelter, the position of the stopping position in relation to the face of the stop, the amount of stands and their location, the width of the side platform, the location of neighboring pedestrian crossings.
- Factors related to the traffic organization, especially at approaches to junctions: the manner of the traffic organization at the junction, including the type of control, the scope and the form of assigning priorities in traffic signals, the quality of signaling, the dedicated traffic lane availability monitoring.
- Factors related to the transit organization, including: the frequency of bus runs and other public transport vehicles, the level of viability of time tables, the scale and the scope of dispatch control; the frequency of inspections checking the quality of transit services, the form of ticket distribution (an on-board ticket machine, tickets sold by the driver).
- Rolling stock factors – the technical characteristics of the fleet influencing the attainability and the upkeep of the desirable speed (the ability to accelerate and brake) and the efficiency of passenger alighting and boarding processes (the capacity of the vehicle, the bus floor height at entrances and other parts of the bus, the number of doors and their width, the number of entrance steps, the sitting and standing area structure and arrangement), failure frequency.

Some of the influences (e.g., drivers' psychophysical and motoric characteristics) are almost unmeasurable in real conditions, while other influences (e.g., the number of passengers giving way to passengers alighting from the bus) are hard to study due to technical issues in executing sufficiently accurate measurements. Rarely do the bus travel disturbances result from a single, easily distinguishable factor. It is most often a cumulated effect of numerous factors, especially in junction areas. The current paper contains the analysis of the issue of time wasted by buses which are stopped in queues at approaches to junctions with traffic signals.

III. MODEL OF BUS TRAFFIC DISTURBANCES AT APPROACHES TO JUNCTIONS WITH TRAFFIC SIGNALS

Assigning right curbside bus lanes, which may be also used by vehicles taking a right turn at the nearest junction, is a very popular solution in European cities. The dedicated lane is open for vehicles taking a right turn in a set distance from the stop line, which helps car drivers to avoid a complicated relation at the approach to the junction. Hence, the traffic stream at the approach has a mixed nature [9] and its structure and length greatly defines the conditions for the bus travel through the junction.

The reference point of this analysis is a single stop-to-stop section with only one signalized junction located in the final part of the section. This section can be divided into two units (Figure 1): the longer part of the section between the head of the bus stop and the stopping line at the approach of junction (S_1) and the smaller part of the section – between junction stopping line and the head of the next bus stop (S_2). In this case, both bus stops are located far side signalized junction. This approach enables a comparison of time lost in a single intersection in relation to undisturbed running time of the section.

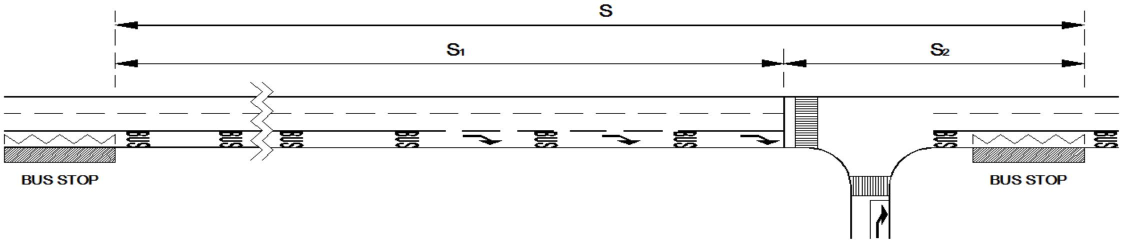


Figure 1. Description of model: stop-to-stop section with designated bus lane and one signalized junction in final part of this section.

A. Undisturbed bus riding the bus lane through the signalized juntion

If vehicles taking a right turn appear at the green signal and are able to leave the bus lane smoothly (the situation at the exit of the junction allows it), then the blockage of buses is occasional. This situation is presented in Figure 2; traffic volume on the bus lane is small, and even in case of significant parallel stream of pedestrians, the bus is able to cross the junction with desired speed.

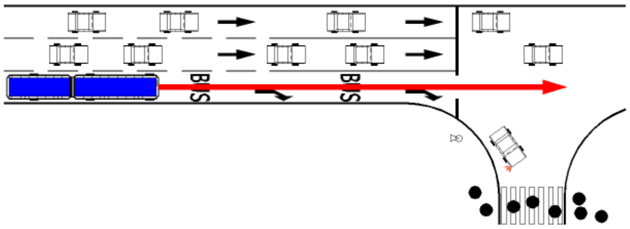


Figure 2. Small traffic volume on the bus lane at the approach to signalized junction, without significant influence onto bus movement.

This is the most convenient situation for passengers of a bus. In case of not extremely long sections, running time consists of acceleration time (during departure from stop), time of running with desired speed and deceleration time before next stop. Acceleration and deceleration times can be estimated with acceleration and deceleration rates - most often, they are values in the range from 0.6 to 1.1 [m/s^2]. The running time of the main part of the section depends mainly on speed limits. In [10], average bus running time T_r [s] on the section with designated bus lane, in close to free traffic conditions, was described as the function of stop-to-stop section's length L [km]:

$$\overline{T}_r = 0.065 \cdot L + 21 \qquad (1)$$

For such a function, the best fit to the results of measurements carried out on the sections equipped with bus lanes in case of close to free bus traffic conditions, in two Polish cities (Cracow and Warsaw) – has been provided. This model is very useful in meso scale transportation planning, but it is not possible to evaluate bus traffic conditions at the approaches to the signalized junctions.

Therefore, we decided to carry out a more detailed analysis with the section travel time separation for: the acceleration time during departure from stop, the time of running with desired speed and deceleration time, just before the next stop on the section. This kind of stop-to-stop running time description has been proposed by Vuchic [11]. There were taken into consideration two cases of bus running in the middle part of the stop-to-stop section:

- Running with (desired) constant speed, chosen by bus driver;
- Running with no constant speed, with coasting – which is an effect of economic driving with respect to energy consumption. In this case, motion at a constant deceleration takes place over almost the entire length of the section.

In the first case, the bus running time can be estimated by using the formula:

$$T_r = \frac{S}{v_{\max}} + \frac{v_{\max}}{2}\left(\frac{1}{a} + \frac{1}{b}\right) \qquad (2)$$

where:
T_r – running time of the section [s];
S – length of the stop-to-stop section [m];
$v_{\max}$ – maximum speed on the section [m/s];
a – average acceleration rate[m/s^2];
b – average deceleration rate[m/s^2].

Whereas, in the second case, it is:

$$T_r = v_{\max} \cdot \left(\frac{1}{a} + \frac{1}{b}\right) + v_c \cdot \left(\frac{1}{b} - \frac{1}{c}\right) \qquad (3)$$

where, additionally:
v_c – speed at the end of the coasting interval [m/s];
c – constant deceleration rate until braking must be applied [m/s^2].

Both formulas have been used in stop-to-stop running time calculations for sections with lengths ranging from 350 to 800 [m] – the same range of sections' lengths was used in (1). In case of running with constant speed, the best fitting to the measurements' results was obtained for acceleration rate equals to 0.70 [m/s^2], deceleration rate 0.80 [m/s^2] and maximum speed 54.0 [km/h]. Whilst, in case of stop-to-stop running with coasting, there were obtained – successively: 0.70 [m/s^2], 0.75 [m/s^2] and 59.5 [km/h]. Additionally, constant deceleration rate equals 0.1 [m/s^2]. Results of this comparison are presented in Figure 3.

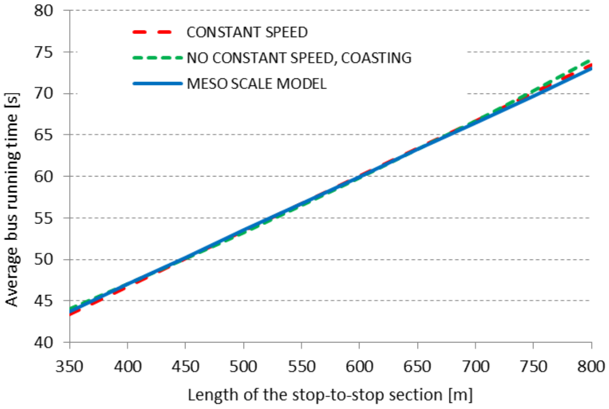


Figure 3. Comparison of stop-to-stop section running times in dependence of the length of the section and the method of running time estimation.

Generally, (1) and (2) could be used in bus running time description. Taking into account adaptation to changes in speed, registered in short intervals, it was decided that the model with constant speed will be used in further analysis.

B. Bus stopping on the first position at the approach of the signalized junction

When the bus has to stop at the stopping line and there is no vehicle in front of it – the amount of time lost depends mainly on the length of the red signal. This situation is presented in Figure 4.

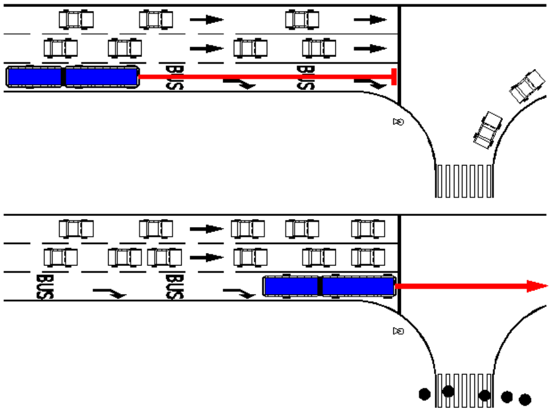


Figure 4. Bus stopping at the approach of the signalized junction, due to the red signal.

In this case, the running time is the sum of the time period from the moment of departure from stop to the moment of stopping at the junction stopping line (t_{S1}), the time of waiting for a green signal (t_w) and the time period from the moment of bus starting from junction stopping line to the moment of stopping the bus at the bus stop (t_{S2}):

$$T_r = t_{S_1} + t_w + t_{S_2} \tag{4}$$

The time period from the moment of departure from stop to the moment of stopping at the junction stopping line can be estimated from (2) for S_1 distance. The waiting time at the red signal usually results from a lack of implementation of traffic signal prioritization at the approach to the junction. It should be determined as an average value based on traffic lights structure. In order to determine the time period from the moment of bus starting from junction stopping line to the moment of stopping at the bus stop, the following formula can be used:

$$t_{S_2} = \sqrt{\frac{2\cdot(\bar{a}+\bar{b})\cdot S_2}{\bar{a}\cdot\bar{b}}} \tag{5}$$

where:
S_2 is the length of the section between junction stopping line and the stop [m].

Most frequently, (5) is used for stop-to-stop running time estimation, when the length of the section is shorter than the critical distance – the running time is composed exclusively of acceleration and deceleration times, because it was assumed that the junction is located at final part of the stop-to-stop section; such an approach is empowered. Finally, the stop-to-stop running time in case of bus stopping on the first position at the approach of the signalized junction one can estimate by using the formula:

$$T_r = \frac{S_1}{v_{\max}} + \frac{v_{\max}}{2}\left(\frac{1}{\bar{a}} + \frac{1}{\bar{b}}\right) + t_w + \sqrt{\frac{2\cdot(\bar{a}+\bar{b})\cdot S_2}{\bar{a}\cdot\bar{b}}} \tag{6}$$

where:
S_1 – length of the section between head of the stop and the junction stopping line[m];
S_2 – length of the section between junction stopping line and the following stop [m].

C. Bus stopping in the queue at the approach of the signalized junction

When traffic volume of right turning vehicles at the junction is significant, buses often stop in the queue. This may result in a significant increase of time lost, especially when the queue is long and/or the parallel pedestrian stream is blocking vehicles turning right. Figure 5 shows this phenomenon.

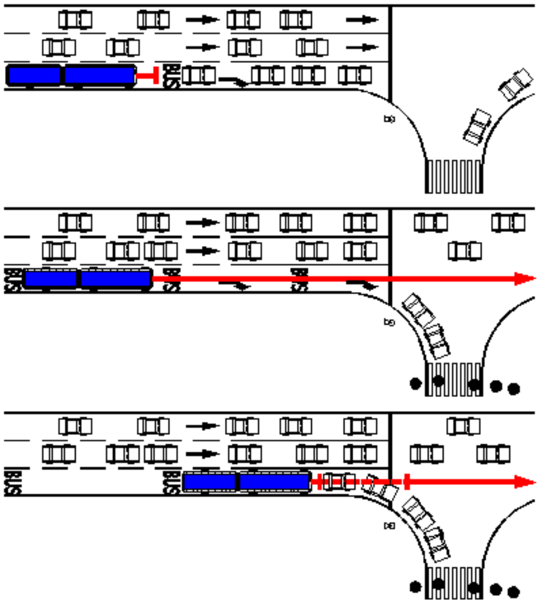


Figure 5. Bus stopping in the queue at the approach of the signalized junction.

If the number of vehicles at the approach is low (usually, no more than 2-3 vehicles) and time lost narrows down to the time needed to stop at the red signal, and the time necessary to clear the queue amounts to several seconds. If, for example, there are two vehicles stopped in front of a bus, after the signal changes to green, they commence a right turn and stop in front of a pedestrian crossing, causing no blockage for the bus, which also resumes the travel.

However, the situation changes significantly, when at the same approach to the junction, the flow of vehicles taking a right turn is longer than the accumulation section before the pedestrian crossing. Then, each consecutive vehicle preceding the bus in the queue will block its motion and force an additional halt. The loss of time resulting from the time necessary to clear the queue will extend. Certainly, if the halt was short, the length of the time lost at the junction would be acceptable. However, if the length of the queue is significant and the traffic situation at the exit of the junction (influenced by a congestion or, more often, by pedestrian movement at the parallel crossing) prevents the queue from clearing smoothly, the bus will lose a significant amount of time. At worst, it will lose the opportunity to cross at the current green signal and the time lost will be extended further due to the necessity to wait for next green light.

When a bus stopping at the approach to the signalized junction takes place in the queue (*K*), two additional time processes have to be introduced. These are:

- Queue clearing time (t_{qc}), defined as the time period from the moment the green signal appears to the moment the bus moves onward from the queue;
- The bus queue motion time (t_{qm}), concerning the queue which formed in front of the bus at the approach to the junction, which is counted from the moment the bus moves onward from the queue to the moment it crosses the stop line at the junction.

In Polish conditions, it is often assumed that every car in the queue equals 6.0 [m] in average. Other kinds of vehicles should be recalculated for cars (e.g., one heavy goods vehicle or one articulated bus in queue – equals 3 cars). If *K*=1 or *K*=2, there is no additional blockage for the bus, and the bus running time can be calculated using the formula:

$$T_r = \frac{S_1 - 6K}{v_{\max}} + \frac{v_{\max}}{2}\left(\frac{1}{\bar{a}} + \frac{1}{\bar{b}}\right) + t_w + t_{qc} + \sqrt{\frac{2\cdot(\bar{a}+\bar{b})\cdot(S_2+6K)}{\bar{a}\cdot\bar{b}}} \quad (7)$$

Whereas, when *K*>2:

$$T_r = \frac{S_1 - 6K}{v_{\max}} + \frac{v_{\max}}{2}\left(\frac{1}{\bar{a}} + \frac{1}{\bar{b}}\right) + t_w + t_{qc} + t_{qm} + \sqrt{\frac{2\cdot(\bar{a}+\bar{b})\cdot S_2}{\bar{a}\cdot\bar{b}}} \quad (8)$$

The queue clearing time and the queue motion time – depend mostly on the amount of vehicles placed in front of the bus and on the traffic situation at the exit caused by the pedestrian traffic at the crossing parallel to the bus lane.

IV. DEPENDENCE BETWEEN QUEUE CLEARING TIME AND LENGTH OF QUEUE

The study on the influence of the length of a queue on the queue clearing time includes the total number of 415 halts at approaches, including 105 instances of queues consisting of 1 to 20 vehicles. The best match was achieved by a simple nonlinear regression model:

$$t_{qc} = \sqrt{18.9 + 2.0\cdot K^2} \quad (9)$$

A graphic representation of the model, matched with the measurement data output, is shown in Figure 6, which also includes the confidence interval (marked with dotted lines), based on a level of confidence of 95%.

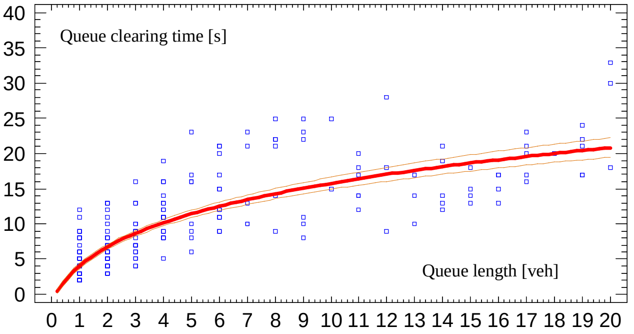


Figure 6. Dependence of queue clearing time from queue length.

The correlation coefficient for this match amounted to 0.93, which translates into a strong relation between the variables.

V. ESTIMATION OF BUS QUEUE MOTION TIME

Moreover, the analysis of the influence of the length of a queue on the queue motion time includes the results of 1237 bus halts at approaches to junctions, including 263 instances of queues consisting of 1 to 20 vehicles. The analysis does not concern instances in which the buses did not manage to cross the junction during a single signal cycle; those instances will be contained in future research. The best match between a regression model and the measurement data output was achieved with a nonlinear regression model:

$$t_{qm} = (2 + 0.856\cdot \ln K)^2 \quad (10)$$

A graphic representation of the model, matched with the measurement data output is shown in Figure 7.

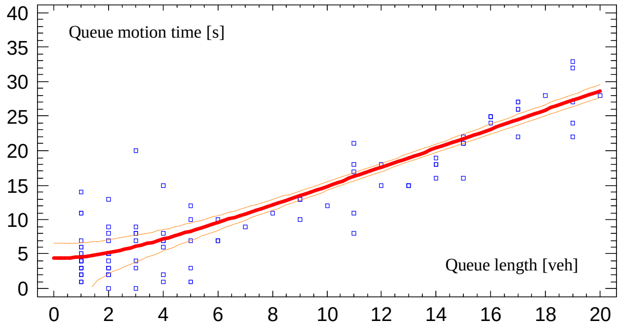


Figure 7. Dependence of queue motion time from queue length.

This time, the correlation coefficient for this fitting amounted to 0.84, which can be considered a strong relation between queue motion time and the length of queue.

VI. METHODS OF REDUCING LOSSES OF BUS TRAVEL TIME AT APPROACHES TO JUNCTIONS

A reduction of queues in bus lanes at approaches to junctions may be achieved through shortening the section of the bus lane which is open for vehicles taking a right turn. If it is too long, the possibility of extending a queue in which a bus may be halted increases. Shortening this section allows to regulate the length of a queue in front of a bus at the approach to a junction – upon the implementation of this solution, vehicles taking a right turn would have to use the open-access lane until the place where the bus lane is open for those taking a turn, which would lead to a decrease in the number of vehicles placed in front of a bus at the approach to a junction. It is a solution which may be recommended for areas of particularly long time lost, where the probability of the queue not being cleared throughout the duration of the green signal – is substantial. Buses often need to wait for the next green traffic signal. Significant reduction of time lost can be also achieved by the activation of a right-turn green arrow at the approach to the junction, which will enable a partial queue clearance during the green signal.

A very efficient solution can be the separation of additional traffic lane for turning right vehicles – outside the bus lane (if it is spatially possible). In that case, the influence of disturbances caused by an excessive traffic of vehicles taking a right turn on the bus traffic is being practically eliminated. The only contact between these vehicles and the bus lane is when they cross it. If the number of vehicles taking a right turn is so great that the queue for the right turn starts in an open-access lane, assigning even two lanes outside the bus lane while simultaneously taking care of a clear signing for the bus lane is worth considering.

VII. EXAMPLE OF BUS RUNNING TIME SIMULATION ON THE SECTION EQUIPPED WITH BUS LANES, WITH ONE SIGNALIZED JUNCTION

Four scenarios of bus operation on single stop-to-stop section at a length 500 [m], equipped with designated bus lane were taken into consideration. On the whole length of this section, only one signalized junction is located. At the approach to this junction, all vehicles turning right use this bus lane. The length from the stop to the stopping line of the junction is 440 [m]. The following scenarios are considered:

- Scenario S1: bus crosses the junction with constant speed, without stopping at approach to the junction;
- Scenario S2: bus stops at the junction stopping line, on the first position (no queue);
- Scenario 3: bus stops in third place in the queue at the approach to the junction (*K*=2);
- Scenario 4: bus stops in sixth place in the queue at the approach to the junction (*K*=5).

Table I presents the values of the time lost per bus, in the considered scenarios. In scenarios 2, 3, and 4, the time of waiting for the green signal is relatively short – it is 20 [s]. Even, if we ignore it, the time lost in scenarios S2, S3 and S4 in comparison to S1 are significant.

TABLE I. SIMULATION RESULTS

Scenario	Bus running time, without t_w [s]	Total bus running time [s]
S1	53.4	53.4
S2	67.4	87.4
S3	73.4	93.4
S4	85.1	105.1

The length of the queue is important as well, its shortening about 3 vehicles (S3 and S4) saves almost 12 [s] per bus. Having considered the number of passengers on a bus, assuming that section is used only by 1000 [pers./h] – the difference between scenarios S1 and S4 in losses of passengers' time during only one hour – amount to 14.36 [h].

VIII. CONCLUSION AND FUTURE WORK

Familiarity with mechanisms of origin of bus traffic disturbances may be helpful in implementation of fully efficient means of bus privileging in traffic. Long queues of vehicles result in far-off positions of buses at the approaches to signalized junctions, and consequently, in time lost for buses. Limitation of the time lost can bring serious benefits for passengers and public transport operators. The results of the presented analysis can be used to improve bus traffic in Polish cities. They may also constitute a word of warning against the implementation of too long sections of shared-access to bus lanes. The methodology can be used also in other countries, it is only necessary to carry out the control tests of the queue clearing time and the queue travel time, taking into account the local conditions of bus traffic. In further studies, the model will be complemented by a priority in traffic lights.

REFERENCES

[1] A. W. Agrawal, T. Goldman, and N. Hannaford, "Shared-Use Bus Priority Lanes on City Streets: Approaches to Access and Enforcement," Journal of Public Transportation, vol. 16, no. 4, 2013, pp. 25-41.

[2] A. Tirachini and D. A. Hensher, "Bus congestion, optimal infrastructure investment and the choice of a fare collection system in dedicated bus corridors," Transportation Research Part B: Methodological, vol. 45, issue 5, 2011, pp. 828–844.

[3] A. Molecki, "The Meaning Of Different Limiters Relations For Defining Public Transport Routes Capacity," Transport Problems, vol. 1, 2007, pp. 95-100.

[4] R. Fernandez and R. Planzer, "On the capacity of bus transit systems," Transport Reviews: A Transnational Trans-disciplinary Journal, vol. 22, issue 3, 2002, pp. 267-293.

[5] X. Chen, L. Yu, L. Zhu, J. Guo, and M. Sun, "Microscopic Traffic Simulation Approach to the Capacity Impact Analysis of Weaving Sections for the Exclusive Bus Lanes on an Urban Expressway," Journal of Transportation Engineering, vol. 136, issue 10, 2010, pp. 895–902.

[6] V. T. Arasan and P. Vedagiri, "Bus Priority on Roads Carrying Heterogeneous Traffic: a Study using Computer Simulation," EJTIR, 8, no. 1, 2008, pp. 45-64.

[7] L. A. Koehler and W. Kraus Jr., "Simultaneous control of traffic lights and bus departure for priority operation," Transportation Research Part C, vol. 18, 2010, pp. 288–298.

[8] E. I. Diab and A. M. El-Geneidy, " Understanding the impacts of a combination of service improvement strategies on bus running time and passenger's perception," Transportation Research Part A 46, 2012, pp. 614–625.

[9] J. Chodur, K. Ostrowski, and M. Tracz, "Impact of Saturation Flow Changes on Performance of Traffic Lanes at Signalised Intersections," Procedia - Social and Behavioral Sciences, vol. 16, 2011, pp. 600–611.

[10] M. Bauer, "Bus Running Time onto Sections with Designated Bus Lanes Modelling," Logistyka 2014, in press.

[11] V. R. Vuchic, "Urban Transit Systems and Technology," John Wiley & Sons, Inc., Hoboken, New Jersey 2007.

Traffic-light Cycle Coordinated by Microsimulation: a Solution to the Traffic Congestion in Palermo

Giuseppe Salvo and Luigi Sanfilippo
Technology and Economy of Transport
University of Palermo
Palermo
{giuseppe.salvo, luigi.sanfilippo}@unipa.it

Abstract—**This article aims to highlight the results of an analysis of the traffic management carried out by the Department of Civil Engineering of the University of Palermo, through traffic microsimulation model and mathematical calculations. In this paper, we demonstrate the usefulness of a coordinated traffic light cycle in an area of Palermo with high traffic flow. To achieve this, we used microsimulation for planning traffic lights, we calculated mathematically the phases in traffic lights and we made changes in the geometry of the intersection. Although this approach is not new, we would like to emphasize the importance of the use of microsimulation models in urban planning of medium-size cities, such as Palermo, characterized by a high rate of traffic congestion.**

Keywords- microsimulation model; traffic lights.

I. Introduction

Mobility is one of the most important elements of a modern society [1] and traffic congestion has been causing many critical problems [2].

Palermo, as many cities and towns in the world, is subject to high traffic flows and their corresponding consequences in terms of air pollution, congestion, and decreased levels of safety.

A proper planning of traffic-light cycle could reduce this kind of problems, improving vehicles flow. Some of the advantages of intersection traffic lights include: homogeneous traffic flows, orderly movement of vehicular currents, reduction of the frequency of accidents [3][4][5][6].

The traffic lights planning cycle is a task often complicated by the number of traffic lights installed in the urban road network: when the nearby traffic lights operate independently, vehicles have a bumpy ride because of the ongoing stops and starts, resulting in degraded performance and increased pollution. To synchronize and coordinate systems means to link them together so as to achieve a constant rate in turning green [7]. It is therefore essential to have a proper synchronization of intersections, whose programming is most often preceded by simulations carried out by the use of microsimulation models. These are a source of immediate and continuous information on the traffic flow [8][9][10][11].

Thanks to the development of innovative software, models capable of representing the dynamic of flow conditions, which actually occurs on road infrastructure, have been created.

Depending on the level of aggregation, there are three main types of models: macroscopic models, mesoscopic models, and microscopic models.

In macroscopic models, traffic flow is defined by the rules of conduct that are a function of the interaction of vehicles with each other and with the infrastructure, and the main variables that are taken into consideration are flows, velocity and density [13]. In this type of model, the current traffic as a whole is analyzed, and not the individual vehicles in the network; therefore, these models are called "continuous". Mesoscopic models differ from macroscopic models by the fact that they consider the current traffic by dividing it into groups of vehicles, which can be, for example, the platoons that move in response to a traffic light stop. Microscopic models, or microsimulation models, differ from the two first mentioned as the smallest unit analyzed is represented by the single vehicle, which moves on the road network and interacts with the other vehicles and with the infrastructure, according to a series of parameters that depend on instantaneous velocity, acceleration, drivers behavior, and mutual distances between the vehicles themselves [13]; these models are then "discrete".

In this paper, we present a microsimulation application related to the area between piazza Don Bosco and piazza Leoni in Palermo (Figure 11), classified in the urban traffic plan as interdistrict roads because of the connection between the city center with Mondello.

Using microsimulation models, two scenarios are analyzed:

- The current scenario (with uncoordinated traffic lights cycles);
- The project scenario (with coordinated traffic light cycles).

In the following section, we describe the current scenario by providing an overview of the intersection arrangement. We also describe the scheme used for the construction of the microsimulation model: matrix origin/destination and the other variables useful for the analysis (e.g., travel times, traffic signal cycle times, etc.).

In the third section, the project scenario is described. In this scenario, differently from the present scenario, the traffic light synchronization is implemented, and we hypothesize a change at the intersection of piazza Leoni. In addition, we provide useful calculations for analysis scenario and in comparison to the present scenario (e.g., saturation flux, load index, equivalent courses, cycle traffic lights, etc.).

In the last section, we show the study results, indicating in particular the benefits in terms of travel time.

II. THE CURRENT SCENARIO

Palermo is the capital of Sicily. With 678,412 inhabitants, this city is one of the most congested in Europe, with critical areas such as intersections, where there are frequent traffic jams. Following an increase in demand for transport, due to the high number of vehicles on the infrastructure, there is a consequent deterioration of the circulation. Therefore, a reorganization of the road network is essential. The reasons for the increase in transport demand are several, such as the increase in the number of cars and the deficiencies is public transport. This leads to overloaded road networks. It is, therefore, necessary to take action and perform analysis to understand what interventions can improve the transport.

To control vehicle movement in intersections, Palermo, as any other city, uses traffic lights in order to minimize interference between moving vehicles.

The area under consideration in this paper includes two intersections in the city of Palermo, piazza Don Bosco and piazza Leoni (Figure 12). This area is congested due to the presence of sport centers in Viale del Fante (such as the stadium, the swimming pool and the race course) and the connection with Mondello; no less important is the presence of numerous lorries that cross the two intersections from via Imperatore Federico to reach highway A29 Palermo-Mazara del Vallo.

The two studied intersections are located at a distance of about 300 meters and a coordinated traffic lights system is currently absent: the two traffic light cycles are uncoordinated for duration and stage numbers.

The traffic light cycle of piazza Don Bosco has three phases articulated (Figures 1, 2 and 3) in 142 seconds (Table I), while the current traffic-light cycle in piazza Leoni has two phases (Figures 4 and 5) lasting a total of 86 seconds (Table II).

TABLE I TIME OF TRAFFIC LIGHTS CYCLE – PIAZZA DON BOSCO

Time of traffic lights cycle 142 (sec)								
Step1			Step 2			Step 3		
Green 1	Yellow 1		Green 2	Yellow 2		Green 3	Yellow 3	
63	3	2	35	3	2	29	3	2
G				Y	R			
G	Y	R						
G	Y	R				G		
R			G	Y	R			
R			G				Y	R
R						G	Y	R

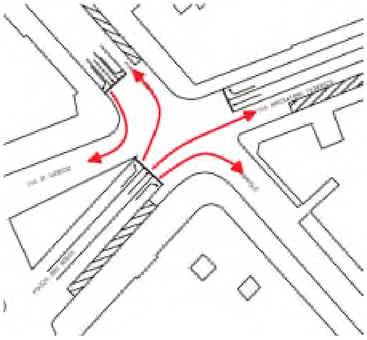


Figure 1. Step 1 in current scenario – Piazza Don Bosco

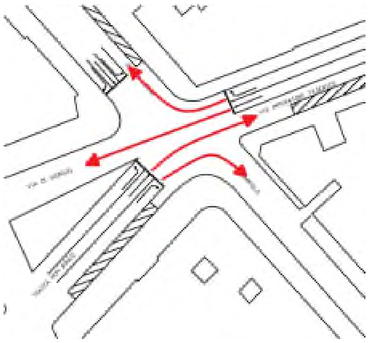


Figure 2. Step 2 in current scenario – Piazza Don Bosco

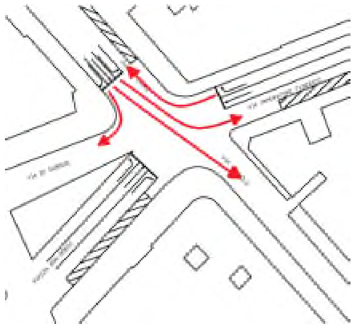


Figure 3. Step 3 in current scenario – Piazza Don Bosco

TABLE II TIME OF TRAFFIC LIGHTS CYCLE – INTERSECTION PIAZZA LEONI

Time of traffic lights cycle 86 (sec)					
Step 1			Step 2		
Green 1	Yellow 1		Green 2	Yellow 2	
41	4	2	33	4	2
G	Y	R			
R			G	Y	R
G					

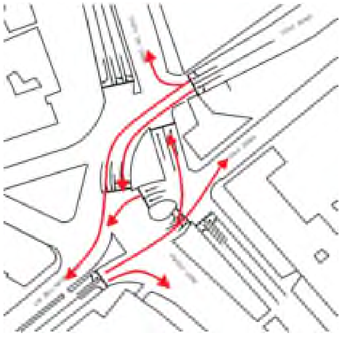


Figure 4. Step 1 in current scenario – Piazza Leoni

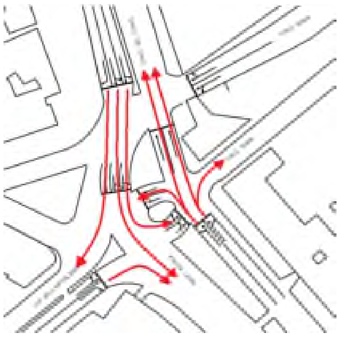


Figure 5. Step 2 in current scenario – Piazza Leoni

From the survey of the traffic flow in each intersection, which was done in December 2013 at 8:00/9:00, 13:00/14:00, 18:00/19:00, we were able to obtain the origin-destination matrix of the whole network for each class of traffic (cars, buses, motorcycles and heavy vehicles). In addition to the vehicles, we have also considered pedestrian flows.

The comparison between the measurements in different bands showed that the traffic flows do not undergo significant changes during the day and it is therefore justified to use a fixed-cycle.

For the analysis through microsimulation models, we consider the time slot 8:00/ 9:00 am.

The maximum traveling speed was estimated through a survey with GPS in a car, carried out over a period of the day with low traffic flows, with data logging interval of 5 seconds. The GPS survey was carried out along 24 routes, connecting each source to each destination. We used a GPS with accuracy of 5 meters and data storage interval equal to 1 second.

The collected data were processed with the aid of the QuantumGIS [14]. Knowing the time of storage of each point we obtain, by the difference, the travel time.

The travel times calculated thanks to the GPS survey were compared to the minimum, average and maximum time values obtained from the simulations, verifying the condition (1).

$$t_{min} \leqslant t_{GPS} \leqslant t_{max} \qquad (1)$$

III. PROJECT SCENARIO

The synchronization of the plant was carried out considering as a priority the traffic coming from the access of viale del Fante and that along the stretch between the two junctions.

We decided to set up a three cycle phases in piazza Leoni (Figures 6, 7 and 8), so we removed two traffic lights currently present in the central island of the same intersection. We retained only the traffic lights that control the left-turn vehicles from viale del Fante, allowing thus in one step the left turn for users coming from viale Diana and via dell'Artigliere and preventing access from piazza Leoni.

The origin-destination matrix of the project scenario was then modified with respect to the real scenario only with respect to the vehicle coming from the intersection of Piazza Don Bosco with destination via dell'Artigliere.

In the phase plane of piazza Don Bosco (Figures 9, 10 and 11), we considered a different sequence of the three phases, giving the green light the same time to the current traffic with similar index in both intersections.

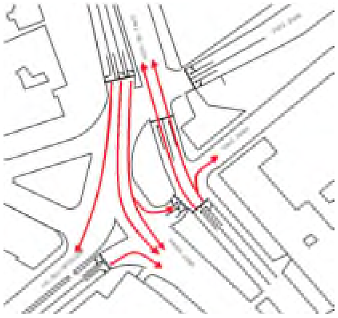


Figure 6. Step 1 in project scenario – Piazza Leoni

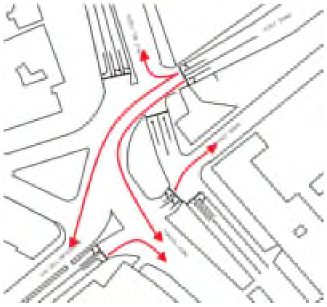


Figure 7. Step 2 in project scenario – Piazza Leoni

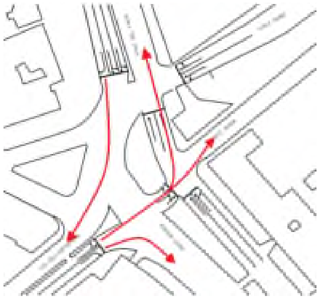


Figure 8. Step 3 in project scenario – Piazza Leoni

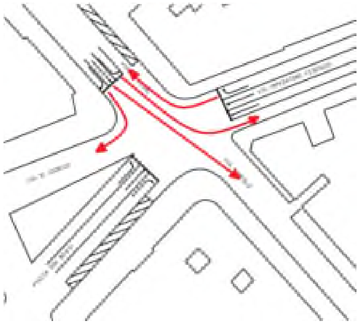


Figure 9. Step 1 in project scenario – Piazza Don Bosco

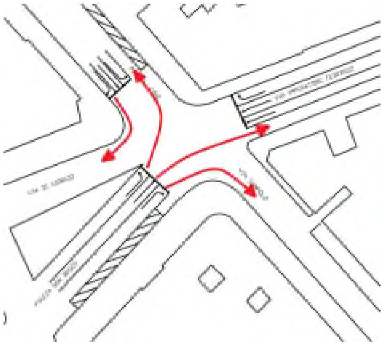


Figure 10. Step 2 in project scenario – Piazza Don Bosco

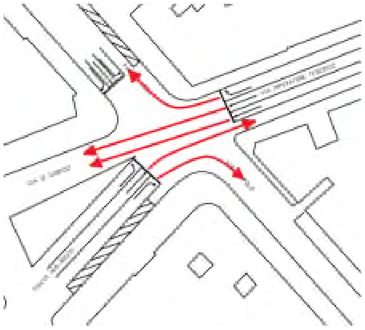


Figure 11. Step 3 in project scenario – Piazza Don Bosco

It was also useful to add an additional lane to the crossing of via Imperatore Federico, whereby the load index of the lane group coming from access lanes in question has decreased. This change was possible because via Di Giorgio is large enough to accommodate two lanes, allowing for an increased capacity of the arc in question.

Regarding the traffic lights of piazza Leoni, we adopted, as a first step, the crossing from viale del Fante, synchronized with the first phase of piazza Don Bosco, which involves crossing and turning from the access piazza Leoni in order to get the green wave.

For the calculation of the traffic-light cycle, we identified the groups of critical lanes and the respective load indices, and we proceeded to the calculation of the minimum cycle.

Load indices, defined as the ratio between the flow rate and the saturation flow, were calculated after determining these two quantities.

The saturation flow was calculated using the relation proposed from HCM (Highway Capacity Manual) [12] as in (2).

$$FS = FS_0 * N * f_b * f_{tp} * f_i * f_p * f_B * f_a * f_u * f_D * f_S * f_{PD} * f_{PS} \qquad (2)$$

where:

- FS_0 is saturation flux under optimal conditions;
- N is the lanes number;
- $f_b$is coefficient of the lane width;
- f_{tp} is coefficient of heavy vehicles;
- f_i is slope coefficient;
- f_p is social activities coefficient;
- f_B is a coefficient that take into account the presence of bus stops;
- f_a is area coefficient;
- f_u is lane use coefficient;
- f_D is right turns coefficient;
- f_S is left turns coefficient;
- f_{PD}e f_{PS} are pedestrian coefficients.

The course equivalents have been deduced from the values of different traffic flows for the four categories of vehicles, by multiplying the coefficients relating to the type in question. The coefficients are:

- 1 for passenger cars;
- 1.75 for heavy vehicles;
- 2.25 for buses;
- 2.5 tram;
- 0.33 for motorcycles;
- 0.2 for bicycles.

This way, we obtained all the load indices for each group of lanes and, for the purposes of calculating the traffic-light cycle, we considered the highest values for each of the three phases, having determined the plane of the stages shown in the previous figures.

The cycle (3) then applies:

$$C_{min} = \frac{\sum_{i=1}^{n}(P_i+TR_i)}{1-(\sum_{i=1}^{n}\gamma_i)} = 133\ sec \qquad (3)$$

where:

- *i* is traffic-light phase;
- *P* is total time waster;
- *TR* is actual red light duration;
- γ_i is equal load capacity index, with the highest value in the i-th stage.

We calculated TR (4) for each of the three phases based on the length of the trajectory of the restrictive vehicular current:

$$TR_i = t_{pr} + \frac{S_f+D+d}{v_v} - G = 3\ sec \qquad (4)$$

where:

- t_{pr} is perception and reaction time;
- S_f is braking distance;
- D is trajectory length;
- d is average length of vehicles;
- v_v is speed vehicles.

The time yellow G is assumed to be equal to 4 seconds, as in urban areas. The braking distance was calculated considering the arrival rate imposed by the limits of the law.

We obtained a minimum cycle of 133 seconds. We decided not to calculate the optimal cycle because it would lead to even higher values. Then, deciding to adopt a three-phase cycle of 133 seconds, we determined green times (5) effective for each phase:

$$VE_i = \left(\frac{Q_i}{FS_i}\right) * C \qquad (5)$$

where:

- Q is the incoming flow;
- C is the entire cycle length

The time thus obtained are:

- VE_1 = 40 seconds;
- VE_2 = 42 seconds;
- VE_3 = 30 seconds.

The times for green (6) and red (7) were found using the following two relations:

$$V_i = VE_i + P_i - G_i \qquad (6)$$

$$R_i = C - (V_i + G_i) \qquad (7)$$

Table III shows the schematic of the traffic-light cycle of the project. As can be seen from the table, some maneuvers are permitted in stages, not presenting points of conflict with other simultaneous operations. Specifically, it is allowed to turn right on Viale del Fante in the first and in the third phase, as well as the right turn from via Imperatore Federico. The crossing and turning right from the access of piazza Don Bosco have been retained in the second and third stages.

TABLE III TIME OF TRAFFIC LIGHTS CYCLE – GLOBAL AREA

Time of traffic lights cycle 133 (sec)								
Step 1			*Step 2*			*Step 3*		
Green 1	Yellow 1		Green 2	Yellow 2		Green 3	Yellow 3	
40	4	3	42	4	3	30	4	3
G				Y	R			
G	Y	R						
G	Y	R				G		
R			G	Y	R			
R			G				Y	R
R						G	Y	R
G								

Finally, we determined the offset time between the cycles of the two intersections, so that the vehicles of the current priority can continue beyond the first traffic lights avoiding further stops. Knowing that the distance between the stop lines of the current priority is equal to 350 meters, advising the maintenance of a constant speed equal to 40 km/h, we calculated an offset of 32 seconds, in this way we favored the current flowing in the path from viale del Fante at the intersection of piazza Don Bosco.

We completed the changes to the project network via Cube Dynasim and we carried out simulations by collecting data from Data Collector always placed in the same sections, in order to make a comparison between the current situation and that of the project. The data were compared for the minimum, average, and maximum travel time, with particular attention to the average time relative to both scenarios.

IV. CONCLUSIONS

The changes made to the current scenario were geometric and logical, having changed the origin-destination matrix, the distribution of the lanes, especially, the traffic-light cycle. These variations have been applied on the microsimulation models, in order to verify an improvement in the conditions of the current outflow with propriety and without having an excessive deterioration of the secondary flows.

The geometric changes have therefore provided the addition of a lane of accumulation for the crossing from via Imperatore Federico, whose length has been set equal to 80 meters, with consequent removal of the area for stopping; the elimination of left turn onto via dell'Artigliere for vehicles coming from piazza Leoni.

When comparing the simulated times with the current scenario, there was a reduction of throughput times for all destinations with origin viale del Fante, via Imperatore Federico, piazza Don Bosco, which are the paths along the streets of interdistrict.

Variation Journey Times = (Time of Scenario Current / Time of Project Scenario) -1 *(8)*

In particular, the paths via Imperatore Federico, via del Fante, and via Sampolo (Table IV) we obtained, as in (8). We also obtained a reduction in average travel times by 20% and 10% to viale del Fante and via Di Giorgio, and via Artigliere and via Diana by just over 7 %. This confirms that coordination reduces the average travel times and, as you can see, the largest decreases for access viale del Fante take place for couples origin-destination connecting the two intersections.

For paths with origin piazza Don Bosco, there were reductions in the average travel time varying from 8 to 38% , with the major reduction happening for maneuvers and left turn crossing into the intersection itself.

The greatest reduction was for the crossing of via Imperatore Federico, where the addition of an extra lane produced a reduction in the average travel time by 55 %, from 134 seconds to just 60 seconds.

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-371-1

In order to demonstrate the utility of using a traffic-light cycle coordinated by microsimulation to reduce the problem of congestion in Palermo, we studied an alternative scenario than the current one. In particular, we have changed the distribution of the lanes and we have determined a traffic-light cycle common to both intersections.

We have temporally spaced so that the traffic-light cycle vehicles of current priority, surpassing the first stop line and maintaining a constant predetermined speed, can continue along the route without making further stops.

By performing the simulation scenario of the project we have observed an improvement in terms of average travel times along the routes identified as a priority, which leads us to confirm the validity of the instrument microsimulation which supports the design of alternative solutions.

V. REFERENCES

[1] R. Jiang and Q. S. Wu, The traffic flow controlled by the traffic lights in the speed gradient continuum model, Physica A: Statistical Mechanics and its Applications, vol. 355, issue 2, 2005, pp. 551-564.

[2] W. Wen, “A dynamic and automatic traffic light control expert system for solving the road congestion problem”, Expert Systems with Applications, vol. 34, 2008, pp. 2370-2381.

[3] J. Garcıa-Nieto, E. Alba, and A. Carolina Olivera, “Swarm intelligence for traffic light scheduling: Application to real urban Areas”, Engineering Applications of Artificial Intelligence 25, 2012, pp. 274–283.

[4] J. McCrea, and S. Moutari, “A hybrid macroscopic-based model for traffic flow in road networks”, European Journal of Operational Research, vol. 207, 2010, pp. 676–684.

[5] J. Sànchez, M. Galàn, and E. Rubio, “Applying a traffic lights evolutionary optimization technique to a real case: Las Ramblas area in Santa Cruz de Tenerife”, IEEE Trans. Evol. Comput., vol. 12, 2008, pp. 25–40.

[6] J. C. Spall and D. C. Chin., “Traffic-responsive signal timing for system-wide traffic control”, Transportation Research, Part C: Emerging Technol. 5 (3–4), 1997, pp. 153–163.

[7] C. Guerra and L. Mussone, “Linee evolutive della regolazione semaforica nel controllo del traffico urbano”, Trasporti e trazione, vol. 1, 1995.

[8] M. Dotoli, M. P. Fanti, and C. Meloni, “A signal timing plan formulation for urban traffic control”, Control Engineering Practice, vol. 14, 2006, pp. 1297--1311

[9] C. Karakuzu and O. Demirci, “Fuzzy logic based smart traffic light simulator design and hardware implementation”, Appl. Soft Comput. vol. 10, 2010, pp. 66–73.

[10] K. N. Hewage and J.Y. Ruwanpura, “Optimization of traffic signal light timing using simulation”, WSC ’04: Proceedings of the 36th Conference on Winter Simulation, Winter Simulation Conference, 2004, pp. 1428–1436.

[11] G. Lim, J. J. Kang, and Y. Hong,, The optimization of traffic signal light using artificial intelligence. In FUZZ-IEEE, 2001, pp. 1279–1282.

[12] HCM2000, Highway Capacity Manual 2000 by Transportation Research Board, 2000.

[13] V. Astarita , D. C. Festa, V. P. Giofré, G. Guido, F. Saccomanno, A. Vitale, "The use of microsimulation as a tool for the evaluation of traffic safety performances", Atti del convegno SIDT 2011 – International Conference, Venezia, 6/10/2011, 2011.

[14] QGIS Development Team, 2011. QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org (last accessed June 2014)

TABLE IV VARIATION JOURNEY TIMES

Time	t_{med}		t_{med}		t_{med}		t_{med}		t_{med}		t_{med}	
	t_{max}	t_{min}	t_{max}	t_{min}	t_{max}	t_{min}	t_{max}	t_{min}	t_{max}	t_{min}	t_{max}	t_{min}
O/D	*Via dell’Artigliere*		*Viale Diana*		*Via Di Giorgio*		*Via del Fante*		*Via Imperatore Federico*		*Via Sampolo*	
Via dell’Artigliere	/		113,64%		31,48%		32,00%		-15,28%		-20,25%	
			-10,61%	0,00%	17,65%	17,24%	-32,96%	-40,74%	-24,64%	28,13%	-30,80%	11,43%
Viale Diana	22,64%		/		-4,39%		17,31%		29,15%		17,07%	
	-4,00%	44,57%			-20,38%	20,77%	2,71%	55,13%	12,76%	76,00%	-6,08%	30,06%
Piazza Don Bosco	/		-12,38%		/		-8,33%		-38,24%		-29,03%	
			-18,60%	16,13%			-21,29%	23,33%	-23,17%	75,00%	-28,41%	166,67%
Via del Fante	-7,45%		-6,71%		-9,88%		/		-20,79%		-19,62%	
	8,14%	-56,00%	1,57%	27,06%	-14,53%	30,00%			-16,78%	-52,38%	-26,78%	-51,00%
Via Imperatore Federico	/		-15,22%		-55,22%		-7,77%		/		/	
			-13,08%	0,00%	-48,93%	-72,09%	-34,60%	-2,33%				

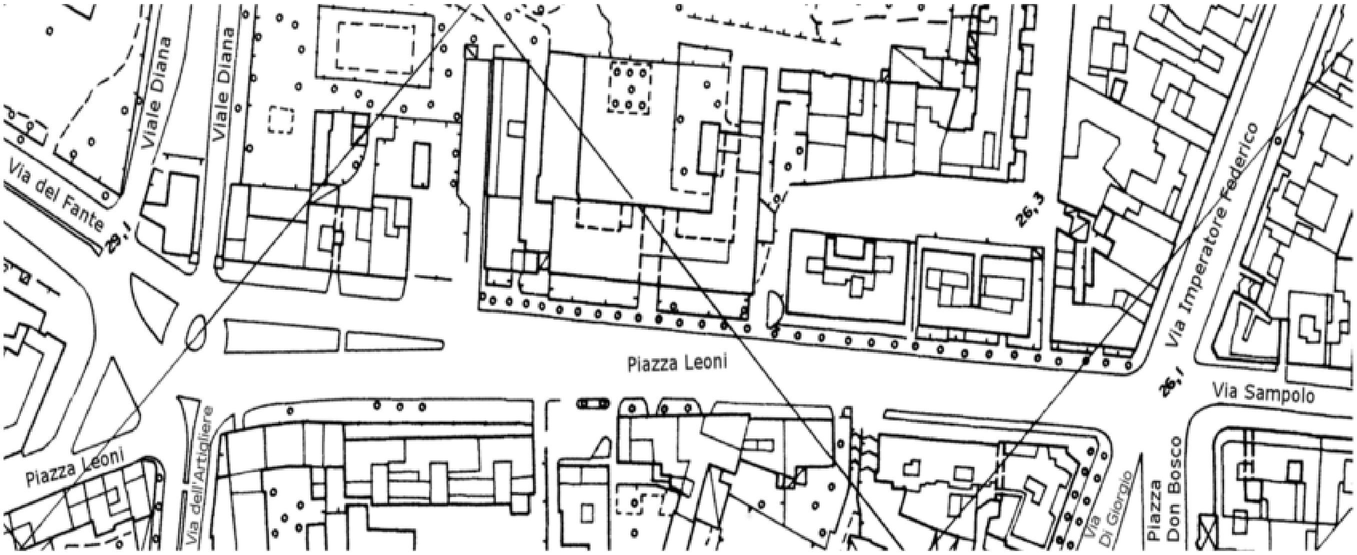


Figure 12. Map of the study area

Towards a Hybrid Real/Virtual Simulation of Autonomous Vehicles for Critical Scenarios

Franck Gechter, Baudouin Dafflon, Pablo Gruer and Abderrafiaa Koukam

IRTES-SeT, UTBM F-90010 Belfort Cedex
Email: firstname.lastname@utbm.fr

***Abstract*—Developing control and perception algorithms for autonomous vehicle is a time-consuming activity if one performs it directly on vehicles hardware level. Moreover, some test cases are hard to reproduce. For this reason, many laboratories and companies are generally using simulation tools. The goal of these tools is to benefit from a testing environment as close as possible to reality and able to reproduce specific testing cases. The main problem with standard simulation tools is that they may not accurately represent real conditions. In order to increase the quality of the simulations, hardware is generally introduced in the loop.**

The goal of this paper is to present a ”work-in-progress” adaptation of IRTES/SeT-Lab simulation tool named VIVUS so as to be able to introduce hybrid simulation which consists in both introducing hardware in the simulation loop and/or software simulation in the hardware experimental loop.

Keywords–hybrid simulation; autonomous vehicle algorithms; sensors simulation; augmented reality

I. Introduction

Developing control and perception algorithms for autonomous vehicle time-consuming activity if one performs it directly on vehicles hardware level. This time cost is linked to hardware issues, vehicle availability, etc. Moreover, some test cases are hard to reproduce (dealing with moving obstacles for instance) or forbidden so as to preserve vehicle integrity (testing collision avoidance algorithm for instance). For this reason, many laboratories and companies are generally using simulation tools. The goal of these tools is to benefit from a testing environment as close as possible to reality in terms of perception and control and able to reproduce specific testing cases. The main problem of standard simulation tools is their distance with real conditions since they generally simplify vehicle/sensors physical models and road topology. Virtual cameras are generally reduced to a simple pinhole model without distortion simulation and vehicle models do not take into account any dynamical characteristics and/or tyre-road contact excepted in specific automotive industry tools such as Calas [1] for instance. Some other tools such as Pro-Sivic [2] are focused on the quality of the virtual sensors having introduced hydro-meteor (rain drops) and other kind of perturbations (fog, etc.) in the sensors simulations. Many laboratories simulators are also exiting focusing each on one specific aspect such as mechanical link between elements, electrical conception [3], platoon control [4] or ergonomic considerations [5]. Despite the quality of these simulations on both physical and perceptual points of view, one needs to increase the quality of the simulations by introducing hardware in the loop. This introduction allows to improve simulation quality while keeping some interesting parts of simulation tools. This hardware in the loop simulation is not really new and is widely used in car manufacturer bench but also for Unmanned Aerial Vehicles (UAV) [6] or Unmanned Underwater Vehicles (UUV) [7]. Generally, the car bench is used to test vehicle dynamical behaviour under stressed situations or for training purposes. Nevertheless, car manufacturers still widely use experiment campaigns for the tuning of vehicle components such as brake systems, driving assistance, etc.

The goal of this paper is to present a ”work-in-progress” adaptation of IRTES/SeT-Lab simulation tool named VIVUS (Virtual Intelligent Vehicle Urban Simulator) [8], [9] so as to be able to introduce hybrid simulation. Hybrid simulation consists in both introducing hardware in the simulation loop and/or software simulation in the hardware experimental loop. If the first kind of simulation is well known in literature, the second aspect is more scarcely represented. This approach allows for instance, to simulate a virtual sensors detecting a moving obstacle and sending data to a real vehicle that will behave following its embedded algorithms (cf. Figure 1).

The paper is structured as follow: Section II presents the global architecture of the VIVUS simulation tool and sketches up the available vehicle modes. Section III gives a short overview of the simulated elements. Section IV introduces the hybrid simulation principle and gives some use case for each situation. Finally, section V concludes this paper while giving some future applications of this work.

II. VIVUS Architecture

This part presents the simulator architecture. After an overview of global structure, this section will focus on specific interaction between external interfaces and VIVUS.

A. Global structure

VIVUS is a 3D based simulator, which supports a 3D render and physical simulation. These two components are ensured by a third party application called Unity3D. The choice of Unity3D was made to limit development time and to enable cross platform applications.

The architecture is based on client-server communication processes. In/out communication can be seen as an external Application Programming Interface (API) where the simulator

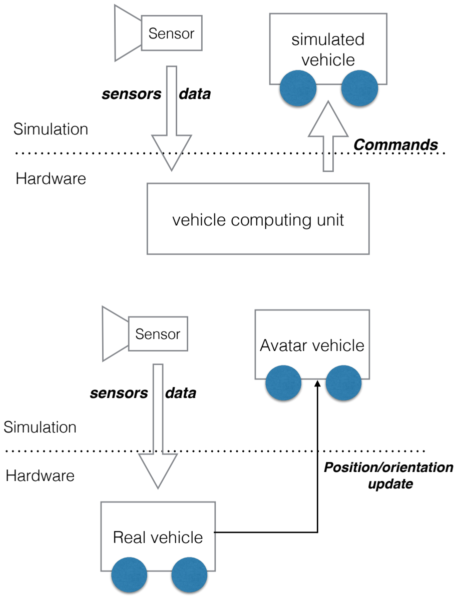


Figure 1: Hardware in the simulation loop (top) / Simulation in the hardware loop (bottom)

offers a list of services aimed at using simulated items or vehicles (cf. Figure 2).

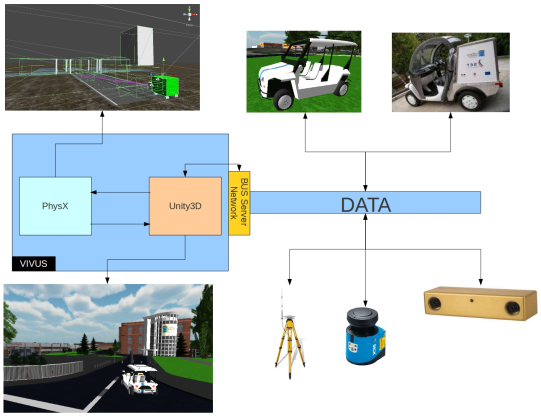


Figure 2: Vivus architecture

The simulator structure is made to allow as much flexibility and adaptability as possible. Each vehicle is considered to be autonomous in terms of perception and behaviour. The structure allows then to have heterogeneous team of vehicles each having a specific equipment and behaviour. At each simulation step, the kernel executes a list of instructions that can be summarized by: (1) *update item perception* ; (2) *update item setpoint* ; (3) *update physics*. The steps *update item perception* and *update item setpoint* are in/out accessors. These can be used to send data from the simulator to an external program or from an external program to the simulator. *Update physics* has got two behaviours depending on the vehicle mode chosen (see section II-B).

B. *Vehicle modes*

As for motion, two modes are allowed by the *update physics* function. The first is a classic motion based on Newtonian laws. The second mode, called Avatar mode, is used for only sensors simulation.

1) Classic motion: Classic motion follows Newton law of motion, based on $\sum F = m.\gamma$ where m is mass, γ is acceleration and solved by physics engine. In this mode, vehicle (or pedestrian) is controlled by an acceleration vector $(acceleration, steeringangle)$ and is limited by its physical capacity.

2) Avatar mode: Avatar mode corresponds to a scan application. Car position and orientation are set, vehicle is then teleported without taking into account physical laws, dynamics and internal parameters. However, sensors are normally simulated linked to their position on the vehicle.

III. Simulated elements

The built-in simulation kernel of VIVUS proposes a set of customisable vehicles and sensors.

A. *Vehicles*

The physical behaviour of the vehicles can be customised following classical parameters such as maximum acceleration, maximal speed, mass, maximum steering angle, etc. In addition, some vehicles are available directly without setting them. VIVUS offers two simulated vehicles inspired by the real experimental platforms developed by SeT laboratory and which are named SeT-Car. This car model has been designed using PhysX engine requirements. The Model used is based on a composition of PhysX elementary objects (cf. Figure **??**). The SeT-Car is then considered as a rectangular chassis with four engine/wheel components. This choice can be considered to be realistic, the chassis being made as a rectangular and undeformable shape.

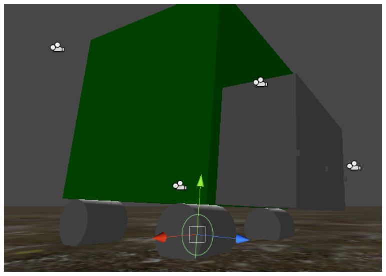


Figure 3: PhysX elementary objects for a SeT-Car

B. *Sensors*

In addition to vehicle models, VIVUS offers a collection of sensors usually found in experimental autonomous vehicles. These sensors are placed relatively to the vehicle without material link since the system does not take into account attachment points. Each sensor can be tuned using a 3D position, a 3D orientation, a frequency rate and its intrinsic parameters. The available sensors are :

- LMS: A Laser Measurement Sensor (Laser Range Finder) able to provide information as a set of distances retrieved from a laser scanning. Intrinsic parameters are aperture and resolution. The default provided laser range finder sends a data set of 181 distances with a resolution of 1 degree and a scanning angle of 180 degrees. (This corresponds to a Sick LMS 200)
- IBEO: Ibeo is a special laser range finder able to convert brightness values from a laser scan. A software driver has been developed to find the position of the three rear beacons laboratory vehicles.
- GPS: Our Global Positionning System receivers are providing NMEA frames. Each frame contains specific dats according to their definition. Intrinsic parameters are: data format (WGS84, Lambert2e, etc.) and origin point.
- Camera: Camera is a sensor able to send over network several video frames. Resolution, framerate and focal distance can be set.
- Odometry: Odometry is measuring the wheel movement. It computes the wheel rotation velocity in Round Per Minute.
- Proprioceptive sensor: This sensor is a customized one, able to provide all intern states of a vehicle (velocity, commands, position, etc). It also simulates an inertial measurement unit.

Each sensor sends a frame over the network.

C. Perturbations and metrics

1) Perturbation: Some perturbations are available in VIVUS. Perturbations represent events that may occur during the movement of vehicles in the environment. They deteriorate sensors and car behaviours. Perturbation can be summarized by :

- Command Perturbations: Command perturbations represents disturbances occuring on command sent to vehicles. They simulates the modifications such as communication loss, magnetic disturbance or interpretation errors, which can happen commonly during experiments. In VIVUS, a command is represented by a 2D vector holding the desired speed (in km/h) and the desired steering angle (in degrees).
- Motor Perturbation: Motor perturbation simulates the possible engine failures. Because the vehicles are electrically powered, a battery discharge is even possible. This discharging produced a disturbance into electrical engine (loss of power), which disturbs vehicle regular behaviour.
- Wheel Perturbation: Wheel perturbation affect a selected wheel of a vehicle. To affect more than one, multiple perturbation must be defined. This perturbation simulates the grip affected by the weather (rain, snow, ice), or by a tire problem.
- Sensor Perturbation: Sensor perturbation is designed to emulate alteration or loss of the vehicle perception. This perturbation is specific to each sensor.

2) Metrics: In many cases, when a simulation is done, you have to make a conclusion from what have happened. Metrics have been also defined to record some parameters during simulation time. They are useful to exploit post-simulation results.

- Inter-vehicles Metrics: Especially designed for platoon algorithm development, these metrics records different distances representing gap between two vehicles.
- Command Integrity Metric :The command metric allow to record all differences between commands sent to a given vehicle and the effective order. These metrics respects the VIVUS command representation (Velocity/Steer Angle).
- Localisation Metric: This metric record the successive positions of a selected vehicle.
- Physical Integrity Metric: Physical integrity is the closest distance before a collision. Time to collision is also evaluated using the current velocity vector.

IV. Introducing hybrid simulation

As explained in the introduction, hybrid simulation consists in mixing together real (hardware) elements and simulated ones. The VIVUS architecture allows to make real and virtual entities communicate together thanks to a network communication based middle-ware. Then, one can define two categories of applications: Hardware in the simulation loop and simulation in the hardware loop. Of course, these two strategies are not mutually exclusive and can be mixed up for specific purposes.

A. Hardware in the simulation loop

Hardware in the simulation is now a wide-spread activity when one want to simulate precisely the behaviour of one specific hardware part of a system. This is used generally for testing components (effectors/actuators, central processing unit, embedded software, etc), testing sensors integration, measuring the time response of a hardware process from sensors data to actuators, ergonomy and design validation, etc. This kind of experiments is used by almost all car manufacturers including the intensive use of car benches.

As for the middle-ware part that allows virtual/real element to communicate together, RTmaps [10] has now become a standard even if some other solutions such as Effibox [11] start to be used in academic applications.

Concerning VIVUS, this has been applied to validate the model of the simulated sensors as compared to real ones. This comparison has been done dealing with sensors data output structure and sensor operation. For instance, the way the laser range finder works has been reproduced in simulation using ray tracing and introducing specular reflexion on objects depending on their texture. Then we compared the obtained result using same kind of detectable objects on both sides.

B. Simulations in the hardware loop

References on this kind of experiments are scarcely represented in literature. The goal is to be able to test real vehicle behaviour in critical cases, such as testing an obstacle

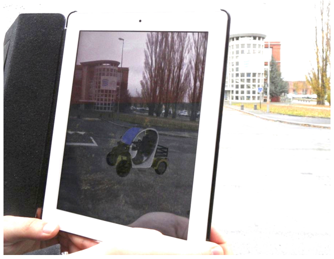


Figure 4: Augmented reality viewer for VIVUS

avoidance algorithm, for instance. The main interest of these kind of configurations is to be able to reproduce precisely the scenarios in terms of moving obstacles trajectories and of perturbations on sensors level (data transmission loss, false data transmissions, optical perturbation due to light exposure, etc.).

Now, let us take an example of application of the use of simulation in the hardware loop. Since a couple of years, IRTES-SeT laboratory is developing driving assistance and automatic control algorithms for autonomous vehicles. Among these, we use, for instance, simulated sensors and real vehicles to test the performances of multi-agent based obstacle avoidance algorithms [12]. To that way, we simulate sensors and moving obstacles in the virtual world. Sensors data is transmitted to real vehicle on which the obstacle avoidance algorithm is running. The real vehicle behaviour is tracked with a cm-precise Real Time Kinematic differential GPS. The real position of the vehicle is then transmitted to VIVUS so as to update correctly virtual perception. In this situation, the avatar mode of VIVUS is used in order to have a representative of the real vehicle into the virtual world. This avatar is required to be able to tackle with geometrical issues linked to real vehicle dynamics, to sensors' positions and to obstacle dynamics. In this example, the use of simulated perception is an important progress by contrast to the classical experimental protocol that are using boxes or to represent static obstacles. Moreover, it allows to perform detailed experiments in terms of combination of sensors and algorithms comparisons.

C. Virtual and real vehicles working together: an interesting side effect

One interesting side effect of the possibility to merge real/virtual vehicles and real/virtual sensors in a same simulation is to be able test multi-vehicle navigation algorithms. For instance, hybrid real/virtual vehicle platoon have been successfully tested. In the tests performed, the leader vehicle of the train is a real vehicle with real sensors. The first follower is a virtual vehicle with virtual sensors that are able to detect the avatar of the train leader. The second follower is a real vehicle with virtual sensors. Moreover, we also develop a augmented reality viewer, which can observe both real and virtual vehicle on a real playground. Figure 4 shows a view of a virtual vehicle over the real playground.

V. Conculsion and futur work

The Simulation/Hardware features introduced into VIVUS enable to perform more precise autonomous vehicle algorithms evaluation due to the introduction of hardware element into the simulation loop. Even if ”hardware in the loop” simulations are widely spread, the fact of using simulated, near real, elements such as sensors into the hardware loop is particularly interesting. Indeed, it allows to test real vehicle behaviours under critical condition following dangerous/forbidden scenarios. For the moment, the coupling between real vehicle and simulated elements is linked to the possibility to measure precisely the real absolute position of the vehicle in its real environment. This measurement requires, for the moment, expensive materials such as differential GPS. From now, we are focusing on this part, trying to figure out if cheaper sensors can be used to solve this issue while maintaining an acceptable precision. To this end, some experiments with an inertial sensor and accelerometers have been done. As soon as the correct precision is reached, we will perform experiments with real vehicles and software sensors such as obstacle avoidance in emergency situations, mixed real/virtual vehicles platoon control, etc We also plan to integrate this tool for tuning our autonomous vehicle algorithms. Eventually, one can say that this feature allows to integrate an intermediate step between classical simulations and experiments.

Acknowledgement

This works was done with the support of the French ANR (National Research Agency) through the ANR-VTT SafePlatoon 4 project (ANR-10-VPTT-011). The authors would like to thank Jean-Michel Contet, Maxime Gueriau and Etienne François for their development.

References

[1] http://www.thales safare.com, [last accessed October 2014].

[2] http://www.civitec.com, [last accessed October 2014].

[3] G. Tsampardoukas and A. Mouzakitis, “Deployment of full vehicle simulator for electrical control system validation,” in Control (CONTROL), 2012 UKACC International Conference on, Sept 2012, pp. 551–556.

[4] K. Kim, D. il Cho, K. S. Huh, and K. Yi, “Real-time heterogeneous multi-vehicle simulator for platoon control,” in SICE 2003 Annual Conference, vol. 1, Aug 2003, pp. 617–621.

[5] M. Kallmann, P. Lemoine, D. Thalmann, F. Cordier, N. Magnenat-Thalmann, C. Ruspa, and S. Quattrocolo, “Immersive vehicle simulators for prototyping, training and ergonomics,” in Computer Graphics International, 2003. Proceedings, July 2003, pp. 90–95.

[6] V. Chandhrasekaran and E. Choi, “Fault tolerance system for uav using hardware in the loop simulation,” in New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on, May 2010, pp. 293–300.

[7] B. Davis, P. Patron, and D. Lane, “An augmented reality architecture for the creation of hardware-in-the-loop hybrid simulation test scenarios for unmanned underwater vehicles,” in OCEANS 2007, Sept 2007, pp. 1–6.

[8] F. Gechter, J.-m. Contet, O. Lamotte, S. Galland, and A. Koukam, “Virtual intelligent vehicle urban simulator: Application to vehicle platoon evaluation,” Simulation Modelling Practice and Theory (SIMPAT), vol. 24, 2012, pp. 103–114.

[9] O. Lamotte, S. Galland, J.-m. Contet, and F. Gechter, “Submicroscopic and physics simulation of autonomous and intelligent vehicles in virtual reality,” in 2nd International Conference on Advances in System Simulation (SIMUL10). Nice, France: IEEE CPS, 2010.

[10] http://www.intempora.com, [last accessed October 2014].

[11] http://effistore.effidence.com, [last accessed October 2014].

[12] B. Dafflon, F. Gechter, P. Gruer, and A. Koukam, “Vehicle platoon and obstacle avoidance: a reactive agent approach,” IET Intelligent Transport Systems, no. 3, Sep. 2013, pp. 257–264(7).

Modelling Drivers' Route Choice Behaviour through Possibility Theory Using Driving Simulator

Mario Marinelli, Mauro Dell'Orco
D.I.C.A.T.E.Ch.
Technical University of Bari
Bari, Italy
mario.marinelli@poliba.it, mauro.dellorco@poliba.it.

***Abstract*— This paper presents a modelling approach based on the Possibility Theory to reproduce drivers' choice behaviour under Advanced Traveller Information Systems (ATIS). The Possibility Theory is introduced to model uncertainty embedded in human perception of information through a fuzzy data fusion technique. Drivers' choice models are often developed and calibrated by using, among other, Stated Preferences (SP) surveys. An experiment is presented, aimed at setting up an SP-tool based on driving simulator developed at the Technical University of Bari. The obtained results are analysed in order to compare the outcomes of the proposed model with preferences stated in the experiment.**

Keywords- modelling; route choice behaviour; possibility theory; data fusion; stated preference; driving simulator

I. INTRODUCTION

The study of travellers' behaviour in Advanced Travellers Information Systems (ATIS) contexts is a crucial task in order to properly simulate phenomena like compliance with information, route choices in presence of information, etc. Several researchers have dealt with conceptual models of drivers' behaviour under information provision. Basic idea in these models is that each driver updates his/her knowledge of costs of alternatives using provided information. Then, he/she compares the updated costs of alternatives and chooses, among them, the best one from his point of view. However, since both knowledge of alternatives and information are rarely perfect, uncertainty affects single person's decision. Handling uncertainty is therefore an important issue for these models. Approaches followed by different scientists to face this issue can be arranged into two main groups, according to how uncertainty has been modelled. In the first group, the approach followed generally uses randomness to represent uncertainty. For this kind of models, unavailability of full numerical data could limit the model reliability; in fact, these models are generally unable to handle non-numerical values of parameters. On the contrary, models included in the second group can easily model un-certainty through verbal, incomplete or imprecise data using the concepts of Fuzzy Logic. In fact, the fundamental concepts of Fuzzy Sets Theory, linguistic variables, approximate reasoning, and computing with words introduced by Zadeh have more understanding for uncertainty, imprecision, and linguistically articulated observations. These concepts support "the brain's crucial ability to manipulate perceptions-perceptions of distance, size, weight, colour, speed, time, direction, force, number, truth, likelihood, and other characteristics of physical and mental objects. A basic difference between perceptions and measurements is that, in general, measurements are crisp whereas perceptions are "fuzzy" [1]. First, Teodorović and Kikuchi [2] proposed a route choice model based on Fuzzy Set Theory. In order to estimate models of the travellers' behaviour, observation of reactions is needed. The most adopted approach for collecting data is the Stated Preferences (SP) one. Two main types of tools for SP in ATIS contexts are the most popular: driving-simulators (DSs) and travel-simulators (TSs). Both methods are computer-based. DSs are characterised by a greater realism, provided that the respondents are asked to drive in order to implement their travel choices, as it happens in the real world. In TSs, travel choices are entered after having received a description of travel alternatives and of associated characteristics, without any driving. In most cases, data have been collected by using TSs, as for instance in [3][4][5][6][7]; while only a limited number of studies have been carried out by adopting DSs [8][9][10][11][12]. In this work, an experiment has been carried out at the Technical University of Bari. Other than the assessment of the internal consistency of the experiment allows for validating the simulation environments, thus showing that more trials and experiments worth to be implemented. The network reproduced in the virtual experiment refer to real one in Bari (Italy). Moreover, the respondents recruited for the experiments were travellers familiar with the networks. The network has been proposed to respondents in the simulations in a double configuration, with and without ATIS. In turns, the configuration without ATIS was presented to respondents with some variants, reproducing different congestion levels and travel times, accordingly with their statistical distribution in the real world. The paper is structured as follows. In the next section, the possibility theory and the data fusion analytical formulation are presented. In section 3, the experiment designed to acquire information about drivers' choice behaviour through a driving simulator is described. In section 4, results of the proposed model are carried out and, in the last section, conclusions are reported.

II. MODELLING PERCEPTION AND DATA FUSION

An informative system, like the ATIS, may provide information to users before they begin the trip (pre-trip information) or while they are moving (en-route information). In the first case, static choice models are involved; in the second case, dynamic ones. In both cases, travellers combine information with their previous experience to obtain a prediction about the cost of each path and to choose the best one.

To incorporate information on system conditions in the choice process, we assume that drivers:

- have some experience about the attributes of the transportation system;
- use information to update his experience;
- choose an alternative according to his updated experience.

Since the drivers' knowledge about the transportation system could be imprecise or approximate, it can be expressed in the same way we used for perceived information. So, both drivers' knowledge and information can be expressed in terms of Possibility, like in Figure 1.

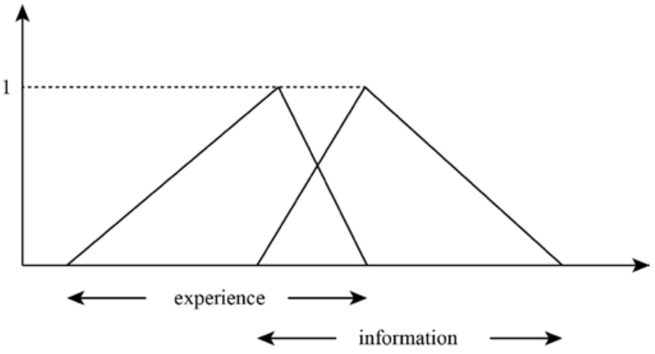


Figure 1. Possibility distributions of experience and information

To update knowledge of the system, drivers aggregate data coming both from their experience and from current information. However, aggregation could not be always meaningful, since data coming from different sources can be far from each other, and thus not compatible. Therefore, a suitable aggregation function should include also a measure of compatibility.

To measure compatibility, Yager and Kelman [13] proposed the relationship $R{:}X^2 \rightarrow I = [0,1]$ such that:

- $\forall x \in X$, $R(x,x) = 1$
- $\forall (x,y) \in X^2$, $R(x,y) = R(y,x)$
- For a given x, R(x,y) is a convex fuzzy set.

A suitable expression for a compatibility function R defined in X^2 is :

$$R(x_1, x_2) = \begin{cases} 0 & \text{if } |x_1 - x_2| > k \\ 1 - \frac{1}{k}|x_1 - x_2| & \text{if } |x_1 - x_2| \leq k \end{cases} \quad (1)$$

where k must be carefully selected case by case, to obtain a proper compatibility measure. Extension to X^n of R is also possible, through the relation:

$$\mathbf{R}(x_1, \ldots x_n) = \min_{i,j=1,\ldots n} R(x_i, x_j) \quad (2)$$

In this paper, we have used for data fusion the *Ordered Weighted Average* (OWA) operator and the compatibility function R, defined in (1).

Given a set $A = \{a_1, a_2, \ldots a_n\}$ and a fusion function F, an OWA operator is a weighting vector $W = [w_1, \ldots w_n]$ such that:

- $w_i \in [0,1]$;
- $\Sigma_i \, w_i = 1$;
- $F(a_1, a_2, \ldots a_n) = \Sigma_i \, b_j \, w_j$

in which b_j is the j-th largest element of A. By adjusting the weighting vector, we can represent different drivers' attitudes: when W favours the smaller valued arguments in the aggregation process it reflects an *aggressive* driver, otherwise it reflects a *cautious* driver.

O'Hagan [14] suggested a method to calculate the weights w_i (i = 1,…n) through the following simple mathematical programming problem:

$$\text{Maximize} \quad -\sum_{i=1}^{n} w_i \ln w_i$$

$$\text{subject to} \quad \begin{cases} \sum_{i=1}^{n} w_i \quad h_n(i) = \beta \\ \sum_{i} w_i = 1 \\ w_i \geq 0 \quad \square i \end{cases} \quad (3)$$

where $h_n(i) = \frac{n - i}{n - 1}$, and $\beta \in [0,1]$ is a coefficient representing, in our case, drivers' cautiousness. Note that, if fusion involves only two sets, then $h_2(1) = 1$, $h_2(2) = 0$. Thus, from the constraints of previous program (Eq. 3):

$$w_1 = \beta, \quad (4)$$

$$w_2 = 1 - \beta. \quad (5)$$

The basic hypothesis we have made in this work to set up a value of β, is that drivers' cautiousness is a function of uncertainty related to perceived information. Let us explain this last concept through an example. Assume that the shorter one of two alternative paths is temporarily closed by barriers. In this case, information that path is closed is not uncertain, that is U(I) = 0, and drivers must choose the longer path. This means that the OWA operator should favour the largest value, that is $w_1 = 1$, and consequently $\beta = 1$ from (4). Conversely, if instead of barriers there is an informative system giving very vague information about the condition of the path, uncertainty U(I) is very large, and drivers should prefer to rely on their own experience. In this case, the OWA operator favours the smallest value, that is w_2 approaches 1 and thus, from (5), β approaches 0. From this example, it appears that the parameter β can be interpreted also as drivers' compliance with information. In fact, $\beta = 1$ means that the driver is totally compliant with information, $\beta = 0$ means the opposite.

Experimental studies have been carried out in last years by some researchers to find out a value of drivers' compliance. Different values, ranging from 0.2 to 0.7, have been found, mainly due to the fact that β is affected by the level of uncertainty imbedded in information. In this study we have assumed that:

- drivers' compliance with information decreases with increasing of uncertainty. This means that the relative elasticity of compliance with respect to uncertainty is negative. In analytical terms: $\frac{d\beta / \beta}{dU(I)/U(I)} < 0$;
- the increase of compliance with additional information is greater in case of ignorance than in case of complete knowledge. That is, the relative elasticity is a function of uncertainty itself.

On the basis of these hypotheses, the following linear relationship between relative elasticity uncertainty level has been carried out:

$$\frac{d\beta / \beta}{dU(I)/U(I)} = -\gamma \cdot Y(I) \qquad (6)$$

and hence:

$$\beta = \frac{1}{e^{\gamma U(I)}} \qquad (7)$$

where γ is a parameter to be calibrated, which takes into account individuals' attributes like age and gender. When n different sources provide information I_i (i=1,…n) with uncertainty $U(I_i)$, compliance rate is calculated as

$$\beta = \min_{i=1,\dots n} \frac{1}{e^{\gamma U(I_i)}} \qquad (8)$$

Now, to incorporate the compatibility concept in the fusion function, we follow the method suggested in [13]. Therefore, let

- A_i (i= 1,…n) be a collection of fuzzy sets. Recall that fuzzy sets and Possibility distributions can be represented in the same way;
- $B = F(A_i)$ be the result of aggregation;
- $A_{i\alpha}=[\, l_{i\alpha}, r_{i\alpha}]$ be the α-cut associated with A_i;
- $l^*_\alpha = \max_i[l_{i\alpha}]$ be the largest lower bound of any α-cut;
- $r^*_\alpha = \min_i[r_{i\alpha}]$ be the smallest upper bound of any α-cut;
- $U^*_\alpha = \inf\{x \mid R(l^*_\alpha,x) \geq \alpha\}$ be the smallest value compatible with l^*_α at level α;
- $V^*_\alpha = \sup\{x \mid R(r^*_\alpha,x) \geq \alpha\}$ be the largest value compatible with r^*_α, at level α.

Provided that $U^*_\alpha \leq r^*_\alpha$ and $V^*_\alpha \geq l^*_\alpha$, the α-cut of B can be calculated as:

$$B_\alpha = [F(d_{1\alpha},\dots d_{n\alpha}), F(e_{1\alpha},\dots e_{n\alpha})] \qquad (9)$$

where:

$$d_{i\alpha} = \begin{cases} l_{i\alpha} & \text{if } l_{i\alpha} \geq U^*_\alpha \\ U^*_\alpha & \text{otherwise} \end{cases} \qquad e_{i\alpha} = \begin{cases} r_{i\alpha} & \text{if } r_{i\alpha} \geq V^*_\alpha \\ V^*_\alpha & \text{otherwise} \end{cases} \qquad (10)$$

The information fusion model incorporates important aspects, such as:

- dynamic nature of information integration. The perceived cost of an alternative is influenced by the user's previous experience and memory;
- accuracy of the informative system. The more accurate information is, the more important is the effect on the drivers' perception;
- non-linear relationship between information and perception.

The parameter β itself is function of information, so that the updated cost is a non-linear function of information.

Possibility is a useful concept in representing decision-maker's uncertainty about the attributes of individual alternatives, but cannot be used directly by analysts; for this reason, a conversion to Probability values on the basis of a justifiable principle is needed.

To pass from Possibility to Probability we use the probabilistic normalization ($\sum_i p_i = 1$), along with the Principle of Uncertainty Invariance, systematized by Klir and Wang [15]. This principle specifies that uncertainty in a given situation should be the same, whatever is the mathematical framework used to describe that situation.

Under the requirement of normalization and uncertainty equivalence, we should use a transformation having two free coefficients. Thus, according to Geer and Klir [16], we use the log-interval scale transformations having the form:

$$\Pi_i = \beta \cdot (p_i)\ \alpha \qquad (11)$$

where Π_i is Possibility and p_i Probability of the i-th alternative; α and β are positive constants.

From (11) we obtain: $p_i = (\Pi_i/\beta)^{1/\alpha}$ and, applying the probabilistic normalization, $\beta = \left(\sum_i \Pi_i^{1/\alpha}\right)^\alpha$ whence, setting ε = 1/α:

$$p_i = \Pi_i^{\varepsilon} \Big/ \left(\sum_i \Pi_i^{\varepsilon}\right) \qquad (12)$$

To calculate ε, we use the Principle of Uncertainty Invariance. Given an ordered Possibility distribution $\{\Pi_1, \Pi_2,\dots \Pi_i, \Pi_{i+1},\dots \Pi_n\}$ for which is always the case that $\Pi_i \geq \Pi_{i+1}$, the possibilistic counterpart of the probabilistic uncertainty, called U-Uncertainty, is given by the following function:

$$U = \sum_{i=1}^{n} \left(\Pi_i - \Pi_{i+1}\right) \ \log_2 i \qquad (13)$$

According to the Principle of Uncertainty Invariance, information I and uncertainty U must have the same value:

$$-\sum_{i=1}^{n} p_i \ \log_2 p_i = -\sum_{i=1}^{n} \frac{\Pi_i^\varepsilon}{\sum_{j=1}^{n} \Pi_j^\varepsilon} \ \log_2 \frac{\Pi_i^\varepsilon}{\sum_{j=1}^{n} \Pi_j^\varepsilon} = \sum_{i=1}^{n} (\Pi_i - \Pi_{i+1}) \ \log_2 i \qquad (14)$$

where $\Pi_{n+1} = 0$ by definition. Numerical solution of (14) is always possible, except when $\Pi_i = K\ \forall i$, $K \in [0, 1]$. However, in this case, from (12) we can easily obtain $p_i = 1/n\ \forall i$.

III. Design of the Experiment

In order to carry out the SP experiments, a PC-based driving simulator of Technical University of Bari has been adopted (Figure 2).

The UC-win/Road driving simulator software was used. This software is developed by FORUM8, a Japanese company. UC-win/Road is plugin-based, allowing to extend software functionalities by using the UC-win/Road SDK Framework that allows for Delphi code. In our case, a plugin was created for data acquisition during driver's simulation, allowing to record for successive analyses (and in CSV format) data related to speed, position, steering, etc. In particular, we have employed data related to position in post-processing in order to observe route choices made by respondents. The simulation system works on a single computer provided with NVidia Graphic Card (1Gb of graphic memory) and a Quad-Core CPU, which guarantees very good real-time rendering and computation performances. The simulation is based on a steering wheel (Logitech™ MOMO Racing Force Feedback Wheel), able to provide force feedback, as well as six programmable buttons (ignition, horn, turn signals, etc.), sequential stick shifters and paddle shifters. A 22" wide-screen monitor was used in order to have a good field of view, also showing internal car cockpit with tachometer and speedometer. Environmental sounds are reproduced to create a more realistic situation.

During the experiment respondents have been asked for choosing a route among three alternatives. The context is configured in such a way that the choice can be assumed as a (possible) switching from a natural reference alternative. As already discussed, respondents were recruited for the experiment ensuring a familiarity with the experimental context. In fact, the simulated networks were part of a real network in Bari (Figure 3). The choice set can be viewed as composed by a main route (route 1) that connects the considered origin-destination pair. Depending on traffic conditions, the traffic could spill-back up to a later diversion node (detour toward route 2) or even up to an earlier diversion node (detour toward route 3). These three different conditions (straight route, later detour, earlier detour) are conventionally classified here as three different levels of congestion (free-flow/low congestion, intermediate congestion, high congestion).

The experiment has been designed in order to have in most of the times (70%) the system in the intermediate congestion pattern, even if extreme (low and high) congestion levels are less frequent. Before starting the simulation, respondents can adapt themselves with the simulator by driving along each alternative route of the choice set, without ATIS and in free-flow traffic conditions. After this possible training, respondents are asked to make 6 successive trials, grouped in 3 driving sessions. At each session, respondents drive twice. The Variable Message Signs (VMSs) representing the ATIS can be active or not in a random way. The activation of the ATIS is a consequence of an accident occurred, that perturbs the standard traffic pattern to an extent that depends on the accident severity.

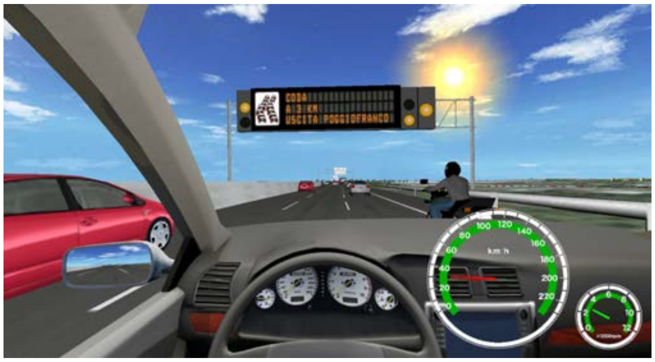


Figure 2. Screenshot of the Driving Simulator

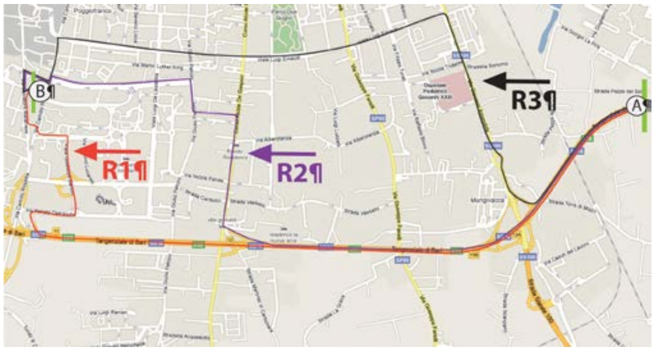


Figure 3. Map of the considered network (Bari, Italy)

TABLE I. LOCATION OF VMSs AND MAIN RAMPS

From	To	Distance (m)
Entrance (A)	1st VMS	400
1st VMS	I Diversion, Exit 13A-Mungivacca	300
I Diversion, Exit 13A-Mungivacca	2nd VMS	700
2nd VMS	3rd VMS	1100
3rd VMS	II Diversion, Exit 12-Carrassi	150
II Diversion, Exit 12-Carrassi	Queue/Accident	900
Queue	Exit 11-Poggiofranco	500

The trials are called “*without information*” when the VMSs are not activated; otherwise trials are called “*with information*”. Moreover, two messages are provided through ATIS: ‘queue’ and ‘accident’, displayed in a random way during all trials. At the first session, respondents can make their choices without information to enforce their perception of the realism of the simulation in terms of consistency with the real network he/she is used to. Then, at a second session, respondents are assisted by information (‘queue’ or ‘accident’). Respondents are provided with information by VMSs located on the main road as reported in Table 1.

The first VMS is 300 meters before the early diversion (Mungivacca, toward route 3), the second is 1250 meters before the late diversion (Carrassi, toward route 2) and the third 150 meters before the same Carrassi diversion node. A queue starts in all cases 900 meters after the later diversion

ISBN: 978-1-61208-371-1

node and 500 meters before the exit-ramp of Poggiofranco. Depending on the simulated congestion level, the queue can spill back more or less. In this scenario VMSs display the presence and the position of the queue, but not queue length or estimated queuing time. At the end of each trial, respondent are asked to answer a questionnaire, where they state, as well as other questions, the chosen route, their perception of the delay due to the provided information (minimum, most expected and maximum time value), if any, and the travel time related to their experience on that route. These data are required to define the triangular fuzzy numbers related to their perceptions in order to apply the data fusion technique previously described.

IV. Results

Recruitment was performed at the Technical University of Bari and 10 respondents were randomly selected. Such a small number of respondents is consistent with the pilot nature of the study.

In trials with information, a respondent perceives a delay as he/she reads the message provided by the VMSs. This delay depends not only on subjective factors, but also by the content of the information itself. Table 2 shows the average time in minutes of perceived delay in terms of triangular fuzzy numbers. We can observe that the average time for the message received 'queue' is lower than that for the message 'accident'. This effect can be imagined because the consciousness of the presence of an accident is definitely more effective in terms of perceptions compared to a queue.

Tables 3 and 4 show the route choice percentages stated at the end of the experiment and compared with the outcomes of the proposed model. Without information (Table 3), it appears that respondents tend to follow the fastest route (R1, Poggiofranco). The changing conditions of the traffic and the presence of queue or incident events does not affect the experience of those who tend to choose the fastest route R1 (87%), in normal traffic, 10% of respondents choose the intermediate route R2 and 3% the slowest route R3.

In the presence of information (Table 4), the choice behaviour varies according to the message displayed on the VMSs. The resulting choice behaviour is consistent with the perceptions of time delays derived from the information provided. A greater perceived delay leads to an increased tendency to abandon the preferred path, turning over other available alternatives. As previously described, the model is characterized by the presence of two parameters *k* and γ (Eq. 1 and 8). These parameters are used to calibrate the compliance with the information system in relation to the provided information. The aim is to find the optimal values of these parameters for each respondent in order to better reproduce the observed behaviour. For sake of simplicity, we have calculated the optimal values of these parameters using Genetic Algorithms (GA) to minimize the root mean square error (RMSE) between the percentages of observed choices and those predicted by the model.

TABLE II. PERCEIVED DELAY TIME (FUZZY VALUES IN MINUTES) IN TRIALS WITH INFORMATION

Respondent	Message 'queue'			Message 'accident'		
	T_{min}	T	T_{max}	T_{min}	T	T_{max}
1	5.5	6.5	8.5	10	15	17
2	10	15	25	10	15	30
3	0	0.5	1	2	3	5
4	4.5	6	10.5	1	2	4
5	2	3	4	5	7	10
6	0	0	0	11.5	15	20
7	2.5	3.5	4	5	8	10
8	1	2	4	3.5	5.5	8
9	3	5	10	7.5	12.5	17.5
10	5	6	10	5	7	11.5
Average	**3.35**	**4.75**	**7.70**	**6.05**	**9.00**	**13.30**

TABLE III. COMPARISON OF ROUTE CHOICE PREFERENCES IN TRIALS WITHOUT INFORMATION

Route	Choice %	
	Observed	Predicted
R1	86.7	90.5
R2	10.0	6.8
R3	3.3	2.7
RMSE	**2.89**	

TABLE IV. COMPARISON OF ROUTE CHOICE PREFERENCES IN TRIALS WITH INFORMATION

Route	Choice %			
	Message 'queue'		Message 'accident'	
	Observed	Predicted	Observed	Predicted
R1	37.50	46.90	0.00	21.65
R2	62.50	47.22	71.43	69.54
R3	0.00	5.88	28.57	8.81
RMSE	**10.90**		**16.96**	

Tables 3 and 4 also show the result of the comparison with observed preferences respectively for trials without and with information. In Table 3, the RMSE value (2.89%) shows the effectiveness of the proposed model to reproduce drivers' choice behaviour in absence of information.

In presence of information, it is important the role of the parameters *k* and γ, whose optimum values have been obtained through GA. Table 4 summarizes the results obtained relating to the ability of the model to reproduce the choice behavior. In the case of 'queue' message, the RMSE value is 10.90%, lower than the case of 'accident' message, 16.96%. Thus, the proposed model is able to reproduce almost correctly the overall behavior and, therefore, the effect of information on users' choices.

V. Conclusions

In this paper, the emphasis was on capturing the reasoning process of drivers making en-route choices in presence of traffic information. The influence of uncertainty in updating the knowledge of attributes of a transportation system, like expected travel time on a path, has been modelled using the concept of compatibility between previous knowledge and current information. The presented model points out the relevant role of the Possibility Theory in calculating uncertainty and thus drivers' compliance level with released information.

A modelling framework, which represents the uncertainties embedded in the perception of travel attributes, has been developed through the Possibility Theory. The model allows the quantitative calculation of users' compliance with information, and thus a realistic updating of expected travel time.

To test the model, an SP experiment has been designed using a Driving Simulation Software for generating a virtual scenario of the city of Bari (Italy). A sample of 10 respondents has been identified to drive using a steering wheel and pedals. The scenario has been proposed in different conditions by combining traffic levels, presence and message type using VMSs. Data acquired through questionnaires have been used to parameterize the proposed model in order to reproduce the resulting driving choice behaviour and perceptions.

The effectiveness of the model has been measured evaluating RMSE values between observed and predicted preferences. The proposed model has resulted to be very effective (2.89% RMSE) in absence of information, where no parameterization is needed. Moreover, the model shows very good abilities in reproducing drivers' preferences under information provision (10.9% RMSE for 'queue'; 16.96% RMSE for 'accident').

In a wider framework, the outcomes of this paper can be used to carry out a road pricing system based on information provision. Therefore, a VMS-based Advanced Traveller Information System can be used as a tool for traffic management.

Future developments concern the extension of this experiment to a greater number of respondents in order to better validate the proposed model. Moreover, an improved driving simulator is intended to be used for a better user experience.

REFERENCES

[1] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility", Fuzzy Sets and Systems 1, 1978, pp. 3-28.

[2] D. Teodorović and S. Kikuchi, "Transportation route choice model using fuzzy inference technique", Proceedings of the First international symposium on uncertainty modeling and analysis, College Park, USA, IEEE Computer Society Press, Dec. 1990, pp. 140–145.

[3] M. E. Ben-Akiva, T. Morikawa, and F. Shiroshi, "Analysis of the reliability of preference ranking data", Journal of Business Research 23, 1991, pp. 253- 68.

[4] H. S. Mahmassani and R. C. Jou, "Transferring insights into commuter behaviour dynamics from laboratory experiments to field surveys", Transportation Research Part A, Policy and Practice 34(4), 1998, pp. 243-260.

[5] E. Avineri and J. Prashker, "The impact of Travel Time Information on Travellers' Learning under Uncertainty", Transportation vol. 33 (4), 2006, pp. 393-408.

[6] C. G. Chorus, E. J. E. Molin, T. A. Arentze, S. P. H. Hoogendoorn, J. P. Timmermans, and B. Van Wee, "Validation of a multimodal travel simulator with travel information provision", Transportation Research Part C: Emerging Technologies, Volume 15, Issue 3, 2007, pp. 191-207.

[7] H. L. Chang and P. C. Chen, "The Impact of Uncertain Travel Information on Drivers' Route Choice Behaviour", In TRB 88th Annual Meeting Compendium of Papers DVD, report n. 09, Jan. 2009.

[8] P. Bonsall and T. Parry, "Using an interactive route choice simulator to investigate drivers' compliance with route guidance advice", Transportation Research Record No. 1306, 1991, pp. 59-68.

[9] P. Bonsall, P. Firmin, M. Anderson, I. Palmer, and P. Balmforth, "Validating the results of route choice simulator" Transportation Research C 5(6), 1997, pp. 371-388.

[10] H. N. Koutsopoulos, T. Lotan, and Q. Yang, "A driving simulator and its application for modelling route choice in the presence of information", Transportation Research Part C: Emerging Technologies, Volume 2, Issue 2, 1994, pp. 91–107.

[11] K. V. Katsikopoulos, Y. Duse-Anthony, D. L. Fisher, and S. A. Duffy, "Risk Attitude Reversals in Drivers' Route Choice when Range of Travel Time Information is Provided", Human Factors and Ergonomics Society, Vol.43, No.3, 2002, pp. 466-473.

[12] H. Tian, S. Gao, D. L. Fisher, and B. Post, "A Mixed-Logit Latent-Class Model of Strategic Route Choice Behavior with Real-time Information", Transportation Research Board 91st Annual Meeting, 2012.

[13] R. R. Yager and A. Kelman, "Fusion of Fuzzy Information With Consideration for Compatibility, Partial Aggregation, and Reinforcement", International Journal of Intelligent Systems 15, 1996, pp. 93 -122.

[14] M. O'Hagan, "Using maximum entropy-ordered weighted averaging to construct a fuzzy neuron", Proceedings 24th Annual IEEE Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 1990, pp. 618-623.

[15] G. J. Klir and Z. Wang, "Fuzzy Measure Theory", Plenum Press, New York, 1992.

[16] J. F. Geer and G. J. Klir, "A mathematical analysis of information-preserving transformations between probabilistic and possibilistic formulations of Uncertainty", Intern. J. of General Systems, Vol. 20, 1992, pp. 143-176.

ISBN: 978-1-61208-371-1

M2ANET Simulation in 3D in NS2

Nasir Mahmood, John DeDourek, Przemyslaw Pochec
Faculty of Computer Science
University of New Brunswick
Fredericton, Canada
e-mail: {nasir.mahmood, dedourek, pochec}@unb.ca

***Abstract*— In this paper, the enhancements to the ns2 source code adding a capability for modeling Mobile Ad Hoc Network (MANETs) and Mobile Medium Ad Hoc Network (M2ANETs) in three dimensions are described. ns2 is an open-source event-driven simulator designed specifically for research in computer communication networks. It allows modeling MANETs in two dimensions which imposes limits on investigating the MANET performance in the real world. Experiments were conducted using the modified ns2 simulator to determine the performance of M2ANETs in 3D, at different node densities and with different movement patterns. The results show that mobile nodes using 802.11 links and running DSR routing protocol can successfully operate as a mobile medium (M2ANET) in a 3D cube with dimensions less than 1500x1500x1500 meters. Simulation experiments also show that deploying of a 3D M2ANET in a multi-story building is not negatively affected by limiting the mobile node movement to horizontal planes corresponding to different floors in the building.**

Keywords-mobile network; simulation; NS2; 3D; MANET; M2ANET; Mobile Medium.

I. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a set of mobile devices that cooperate with each other by exchanging messages and forwarding data [1][2]. Mobile devices are linked together through wireless connections without infrastructure and can change locations and reconfigure network connections. During the lifetime of the network, nodes are free to move around within the network and node mobility plays a very important role in mobile ad hoc network performance. Mobility of mobile nodes significantly affects the performance of a MANET [2].

A Mobile Medium Ad Hoc Network (M2ANET) is a particular configuration of a typical MANET proposed in [3], where mobile nodes are divided into two categories: (i) the forwarding only nodes forming the so-called Mobile Medium, and (ii) the communicating nodes, mobile or otherwise, that send data and use this Mobile Medium for communication. The advantage of this M2ANET model is that the performance of such a network is based on how well the Mobile Medium can carry the messages between the communicating nodes and not based on whether all mobile nodes form a fully connected network. An example of a M2ANET is a cloud of drones released over an area of interest facilitating communication in this area. Recently, a number of projects that match the M2ANET model have been announced; they include Google Loon stratospheric balloons [4] and Facebook high altitude solar powered planes [5] for providing Internet services to remote areas, and the Swarming Micro Air Vehicle Network (SMAVNET) project where remote controlled planes are used for creating an emergency network [6].

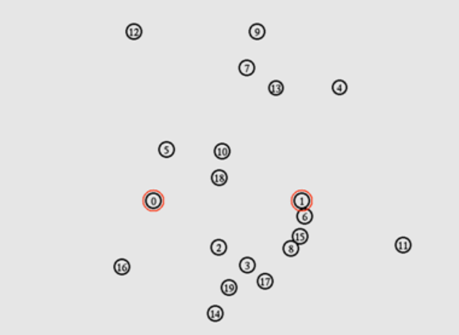


Figure 1. ns2 simulation screen of a MANET in 2D.

The existing simulation environments generally lack 3D capabilities. This paper describes the modifications to the ns2 simulator (Figure 1), version 2.35, adding a 3D mobile network modeling capability. The modified simulator is then used to investigate sample 3D M2ANET scenarios including a number of mobile nodes moving randomly in a 3D volume bounded by a cube and hypothetical scenario of mobile nodes moving on different floors in a multi-story building.

In Section II, the existing 3D network simulation tools are overviewed. Section III describes the modifications made to the ns2 simulator. Section IV describes a case study of wireless network simulation in 3D. Conclusions are presented in Section V.

II. STATE OF THE ART

A MANET is comprised of interconnected mobile nodes, which make use of wireless communication links for multi-hop transmission of data. The mobility plays a paramount role in the operation of a mobile network, however most scenarios referred to in the literature focus on 2D [7]. One of the few tools available for modeling node mobility in 3D is MobiSim [8]. It is a mobility management tool designed to produce movement trace files, Z coordinate included,

Copyright (c) IARIA, 2014.   ISBN: 978-1-61208-371-1

which can then be used for analysis or as an input to a simulator. As it generates only the movement traces, it cannot be used as a data network simulator. OPnet Modeler [9], in particular the Wireless Suite for Defense, includes integrated tools that incorporate the terrain modeling, node mobility and 3D visualization (3DNV), but, as a proprietary package, OPnet is expensive to license and lacks extensibility needed for experimentation with novel network mobility patterns [10]. ns2 is an open source simulator with built-in tools for modeling wired and wireless networks and for traffic visualization [11]. In its current version it does not support wireless node modeling in 3D.

In simulation, a random mobility is often used as a reference case scenario, mostly because of the relative ease of implementing it in a simulator. One of these popular models is the Random Way Point (RWP) model available in ns2 [10]. Nodes are moved in a piecewise linear fashion, with each linear segment pointing to a randomly selected destination and the node moving at a constant, but randomly selected speed.

III. MODIFICATIONS TO NS2

ns2 is an open source simulator consisting of over 300,000 lines of code in a number of libraries. The code uses a number of coding conventions including the representation of a mobile node's location. Mobile node coordinates are designated with variables X_, Y_ and sometimes Z_. As the simulator is designed for modeling 2D environments, if and when Z_ is used it is set zero.

The search of ns2 source files for variables X_, Y_ and Z_ followed by code inspection resulted in the following candidate files for modifications listed in Table I.

TABLE I: MODIFIED SOURCE CODE FILES

File	Methods

Mobilenode.cc	MobileNode( )
	command( )
	bound_position( )
	set destination3d( )
	update_position( )
	log_movement( )

Mobilenode.h	getVelo( )
	destZ( )
	set_destination3d( )
	initialized( )

Topography.cc	Load_cube( )
	command( )

Topography.h	Load_cube( )
	lowerZ( )
	upperZ( )
	Topography( )

In order to move a node in 3D space, a 3D topography needs to be defined. The maxZ of type double was defined and initialized to 0.0 in the Topography constructor where the maxX and the maxY are already initialized to 0.0. There are two getter methods defined for the maxZ in the topography header file. One is lowerZ, which always returns 0.0, and upperZ, which returns the product of maxZ and grid_resolution, where grid_resolution has default value of 1 unless initialized with different value using a TCL script.

A new method named load_cube was defined, which accepts the arguments x, y, z, and res, which are used to initialize maxX, maxY, maxZ, and grid_resolution, respectively. This load_cube method is called from the TCL script written for the purpose of running a 3D simulation experiment.

In the mobilenode header, X_ , Y_ , and Z_ are already defined in the original version of ns2 and they represent the position of a node in the topography, and the Z_ was neither implemented nor used. The dX_ , dY_ , and dZ_ are also already defined and represent a unit vector that specifies the direction of the mobile node movement. This vector is used to update the position of a node. The destX_ , destY_ , and destZ_ represent where the node is going. In order to get a value of the destZ_, a getter method was defined for the destZ_. The initialized method must check whether the topography and trace object are initialized and it also must check whether X_ , Y_ , Z_ are within the topography boundary. The Z_ condition was added in the initialized method to make sure the Z_ coordinate of the node is within the defined topography. In the getVelo method, the velocity factor for X_ and Y_ is already calculated using dX_*speed_ and dY_*speed_ respectively, so the velocity factor for Z_ which is the product of dZ_ and speed_ was added.

The new set_destination3d method is defined in mobilenode header, and takes X, Y, Z destination coordinates of a mobilenode along with the speed.

In the mobilenode class, the MobileNode constructor initialized destZ_ to 0.0. The constructor is also initializing variables like X_ , Y_ , Z_ , speed_ , dX_ ,dY_ , dZ_ , destX_ , and destY_ to 0.0 along with other variables. The bound_position method was also changed: it gets the lower and the upper bound of the X, Y, and Z coordinates of the topography and then checks whether the current position of the node is within the topography. If not the bound_position method will decrease or increase the coordinate value accordingly. In the bound_position method there was no handling of the Z coordinate, so this functionality was added to make it work for 3D. The set_destination3d method makes sure that all the class variables are initialized correctly, checks whether the new destination is within the topography boundary, and then makes a call to the update_position method, which calculates the current position of the node and then updates it to the new position. Then the update_position method makes a call to the log_movement method, which writes the new position in the log file. After the complete execution of the log_movement and update_position methods, the remaining set_destination3d method was executed which saves the value of the previous position of the node. The Z coordinate in the set_destination3d, update_position, and log_movement methods were implemented.

Modifications to the setdest movement file generation utility required modifications to the setdest class file.

The setdest class file does not have MAXZ defined so MAXZ is set to type double and initialized to 0.0 which is then used for the Z coordinate of the topography boundary. In order to set the value which is received as a command line argument for the Z coordinate of a topography another case was added in the switch statement for the Z coordinate that gets the command line argument value which comes after -z and saves it to MAXZ. The setdest utility terminates if the MAXZ value, which is received as a command line argument after –z, is equal to 0.0.

In setdest movement, the speed of the node is variable and does not exceed the max speed which is specified as a command line argument. We simplified the setdest utility and made the speed constant: the MINSPEED is set to MAXSPEED, and the speed is updated to be equal to MINSPEED in the RandomSpeed method. In modifying setdest we chose to keep the speed constant to make node movement modeling simpler and easier to trace and debug.

The RandomPosition method is called for node coordinate initialization. In this method, node X and Y coordinates are equal to uniform()*MAXX and uniform*MAXY respectively and the node Z coordinate is equal to 0.0. To make the RandomPostion method compatible with 3D Z is updated to uniform()*MAXZ.

The RandomDestination method is similarly updated. The RandomDestination method is used to assign a new destination to the node. In this method, node X and Y coordinates are equal to uniform()*MAXX and uniform()* MAXY and node Z coordinate is equal to 0.0 so it is updated to uniform()*MAXZ.

The print statement was updated so it also prints the z coordinate for node movement. Finally, the setdest3d method is used instead of the setdest method. The Z coordinate value of a node for Z coordinate initialization is already written to the output file.

The modifications to the ns2 source code and to the setdest utility were successfully tested with a number of validation tests which included: verifying if the 3D boundary limits are enforced, whether the nodes move correctly along paths defined in 3D, whether packets are received successfully in the 3D environment. As in standard ns2, at the end of a simulation run, the modified version generates a trace file which includes an additional field showing the Z coordinate on each node, like in the example below:

```
r -t 16.005981008 -Hs 0 -Hd 0 -Ni 0 -Nx
832.41  -Ny  842.41  -Nz  852.41  -Ne  -
1.000000 -Nl AGT -Nw  -Ma 13a -Md 0 -Ms
1 -Mt 800 -Is 1.0 -Id 0.0 -It cbr -Il
500 -If 0 -Ii 18 -Iv 32 -Pn cbr -Pi 16 -
Pf 1 -Po 0
```

IV. EXPERIMENTS IN 3D

The enhanced version of ns2 described in this paper was used to model a mobile ad hoc network used as a mobile medium, i.e., M2ANET. Two groups of experiments were conducted: (i) free space experiments with one source and one destination node and the mobile medium nodes moving randomly inside a cube of different dimensions and (ii) multi-story building experiments with two sources and two destinations located at different floors of the building, and the mobile nodes moving either randomly or being limited to the planar motion of different floors.

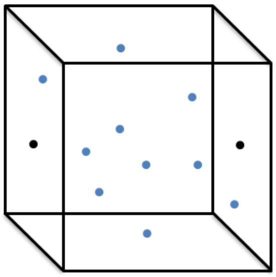


Figure 2. Free space movement in a cube.

The source and destination nodes were placed in the middle on the opposite sides of the cube (Figure 2) and 100 meters from the wall. In case of the two-story building scenario the locations were in the middle of the wall at each floor, with the source sending data to the destination on the same floor. For example, for a 2000x2000x2000 meter cube the source would be at (100, 1000, 1000) and the destination at (1900, 1000, 1000), see Fig. 1 for the general arrangement. In the experiments we used 500 byte packets sent every second (at Constant Bit Rate) over an 802.11 channel. The free space propagation model was used giving each node the communication range of 600 meters. The Dynamic Source Routing (DSR) protocol was used for routing.

Figure 3 shows the summary of simulation results for the first set of experiments in free space bounded by a cube of different dimensions: 1500, 2000, and 2500 meters. In the experiments the packet delivery was measured and averaged over 10 runs for each mobile node density. The first observation is that the performance of the network improves with the increasing number of mobile nodes in the mobile medium. For a 1500 m cube the highest packet delivery occurs at about 20 nodes, for a 2000 m cube at 40 nodes and for a 2500 m cube at 45 nodes. This is because for a larger cube more nodes are needed to build a path for sending packets from the source to the destination. The second observation is that the packet delivery declines when a large number of nodes are used. This is consistent with the DSR routing protocol performance where the route discovery overhead increases with longer paths and frequent disconnections in a larger network [12]. It is interesting to note that the rapid decline occurs at 75 nodes in a 1500 m cube, at 60 nodes in a 2000 m cube and at 50 nodes in a 2500 m cube.

.

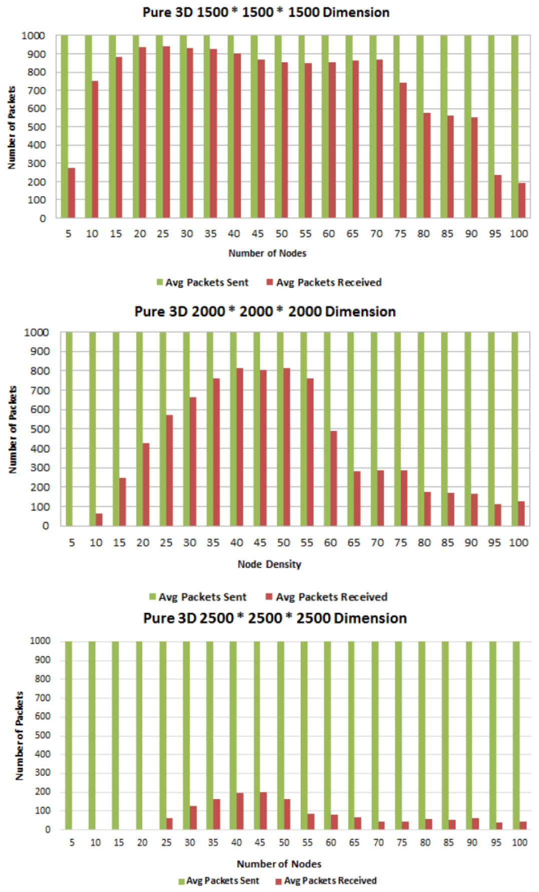


Figure 3. 3D experiments: free space movement in a cube

The two factors impacting the operation of the M2ANET in the experiments, namely improvement in performance due to more nodes available to create the network and decline in performance due routing overhead in a network with a large number of nodes, are opposing each other and have different impact depending on the size of the cube in which the network is deployed. For a smaller 1500 m cube the performance is stable for a range of node densities (20 to 70 nodes) while for a larger 2500 m cube the network never performs adequately because the delivery starts to decline due to DSR overhead even before the number of nodes would reach adequate density for forming a reliable path. This shows a limit on suitability of a DSR based network for creating a mobile medium covering a larger volume.

In the second set of experiments, we set out to demonstrate the use of the 3D network simulation tool in a more practical example of a multi-story building (Figure 4) where the movement of the nodes in not necessarily random in free space bounded by a cube but is limited to a planar movement on each floor, with the nodes on one floor interacting with the nodes on another floor when they are in range.

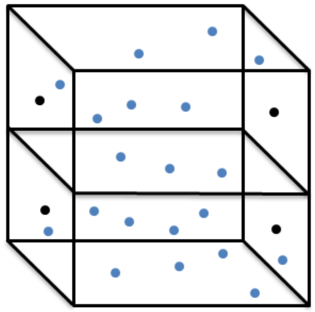


Figure 4. Multi-story building: two sources and two destinations with free space movement.

The experiments were run in a 1500x1500x1500 space with source and destination nodes on two planes: Z = 500 and Z = 1000. A total of 20 mobile nodes was used in each experiment. Three scenarios were tested: (i) random: all 20 nodes moving randomly in a 1500 m cube with no restrictions, (ii) independent: mobile nodes are divided in to two groups of 10 and each group moves randomly but is restricted to a different floor (plane), and (iii) synchronized: nodes are still divided into two groups, one group moves randomly on the upper floor and the other group on the lower floor shadows the movement of the first group with each node staying right below the node on the upper floor.

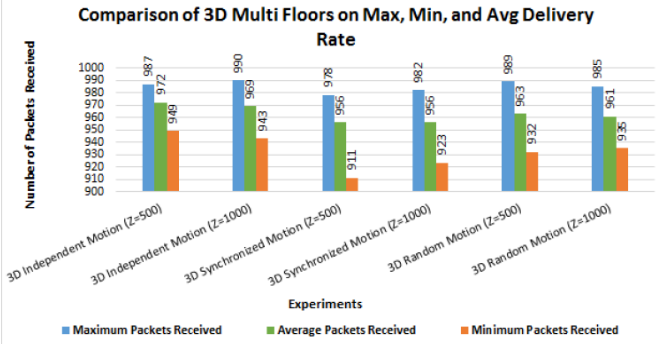


Figure 5. Comparison of delivery rate for three multi-story building scenarios.

The results of packet delivery averaged over 10 simulation runs (Figure 5) show a slight advantage of the scenario when the nodes are restricted to different floors and move independently. This can be explained by noting the fact that when totally random movement is used inside a cube, some nodes move into the corners of the cube which would limit their participation in forming a path form source to destination while restricting the node movements to each floor avoids this occurrence.

V. CONCLUSION AND FUTURE WORK

In this paper, modifications to the ns2 version 2.35 simulator required for adding a new capability for modeling mobile networks in 3D were described. ns2 is an open source simulator and consists of over 300,000 lines of code. The modified ns2 has a capability to model MANET nodes and mobility patterns in 3D. A number of validation tests were performed to test the newly added 3D capability in the ns2 source simulator.

The modified simulator was used to experiment with a mobile medium network (M2ANET) in 3D. The tests showed that the mobile medium formed by mobile nodes running DSR over 802.11 connections with free space propagation range of 600 meters performs well in a 3D scenario covering a 1500x1500x1500 meter cube. In a larger cube, a high performance could not be achieved as a larger network with larger number of nodes required to fill in a larger space suffered from the performance decline due to an increasing overhead of the DSR routing protocol.

Experiments with modeling a multi-story building showed that the mobile medium performance would not decline when the mobile node movement is restricted to moving on each floor rather than randomly in free space. Based on our results, we suggest trying different existing ad hoc routing protocols in the modified simulator in 3D. It may be required to develop new routing schemes especially suited for forwarding in a mobile medium in 3D.

REFERENCES

[1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic (Eds.), Mobile Ad Hoc Networking. New York: Wiley-IEEE Press. 2001.

[2] F. Bei and A. Helmy, A survey of mobility models in wireless Ad hoc Networks, University of California, USA,2004.

[3] J. DeDourek and P. Pochec, “M2ANET: a Mobile Medium Ad Hoc Network”, Wireless Sensor Networks: Theory and Practice, WSN 2011, Paris, France, Feb. 2011, pp. 1-4.

[4] S. Levy, "How Google Will Use High-Flying Balloons to Deliver Internet to the Hinterlands", Wired, June 2013, http://www.wired.com [retrieved: July 2014].

[5] J. Brustein, “Facebook’s Flying Internet Service, Brought to You by Drones”, Bloomberg Businessweek, March 4, 2014.

[6] A. Jimenez Pacheco et al., “Implementation of a Wireless Mesh Network of Ultra Light MAVs with Dynamic Routing”, IEEE GLOBECOM 2012, 3rd International IEEE Workshop on Wireless Networking & Control for Unmanned Autonomous Vehicles 2012, Anaheim, California, USA, 2012, pp. 1591 - 1596.

[7] N. Aschenbruck, E. G. Padilla, and P. Martini, “A survey on mobility models for performance analysis in tactical mobile networks”, Journal of Telecommunications and Information Technology, v2, 2008, pp. 54-61

[8] S. M. Mousavi, H. R. Rabiee, M. Moshref, and A. Dabirmoghaddam, "MobiSim : A Framework for Simulation of Mobility Models in Mobile Ad-Hoc Networks", The 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (IEEE WiMob 2007), New York, USA, October 8-10, 2007, p. 82.

[9] OPnet Modeler, www.opnet.com [retrieved: June 2014]

[10] Ch. R. Rowell, “Modeling Computer Communication Networks in a Realistic 3D Environment”, MCS Thesis, Department of Electrical and Computer Engineering, Airforce Institute of Technology, Wright-Patterson Air Forece Base, 2010, Ohio, USA.

[11] H. Ekram and T. Issariyakul, Introduction to Network Simulator NS2, Springer, 2009.

[12] A. M. Kanthe, D. Simunic, and R. Prasad, “Comparison of aodv and dsr on-demand routing protocols in mobile ad hoc networks”, Emerging Technology Trends in Electronics, Communication and Networking ET2ECN 2012 1st International Conference on, Dec 2012, pp. 1-5.

Pandora: A Versatile Agent-Based Modelling Platform for Social Simulation

Xavier Rubio-Campillo

Barcelona Supercomputing Center
Computer Applications in Science & Engineering
C/Gran Capita 2-4, Nexus I Building, Barcelona
Email: xavier.rubio@bsc.es

Abstract—**The evolving field of social simulation is diversifying the degree of complexity of published works, from simple models to large scale simulations with millions of agents. In this context, available platforms are divided between the ones favoring easy to use interfaces and the development of prototyping, and others created for simulating large scenarios with high computing costs. The later group is truly diverse, based on the way executions are accelerate and the wide range of technologies that they can support. The cost of this increase in performance is a steep learning curve, as the users of these platforms need to have advanced programming skills in order to deal with code complexity. Pandora is a novel open-source framework designed to fill the gap between these approaches. A twin interface in Python and C++ offers the same interface to users developing prototypes as well as projects with High-Performance Computing requirements. The need for parallel programming knowledge is also skipped through the automated generation, during compilation time, of needed code for shared and unshared memory distribution using OpenMP and MPI. A set of different helpers (unit testing, georeferencing support) and analytical tools complement the basic framework in order to facilitate the tasks of development, testing and analyzing any type of Agent-Based Model. Pandora's flexibility is exemplified through different projects that have introduced GPU acceleration, georeferenced data and cloud computing to the framework.**

Keywords*–*Agent-Based Models; Social Simulation; High-Performance Computing; Parallel programming; Multi-Agent Systems

I. Introduction

The disciplines studying human behavior are being increasingly interested on the use of Agent Based Models (ABMs) as virtual laboratories to explore and validate their research hypotheses. This trend has determined the appearance of a growing number of platforms specifically designed to assist projects creating this kind of models. At the same time, the diversification of case studies and requirements has created a bifurcation between platforms created to implement prototypes and simple models and those designed to face advanced requirements, specially regarding the acceleration of the execution.

The first group of platforms tends to emphasize easy to use interfaces, and a quick learning curve that allows for producing models in a short span of time. These features are extremely interesting for a sensible part of the community of social scientists, as it allows to create, deploy and explore simulations with little or non-existent previous programming experience. The powerful capability of creating models without expert programming skills explains the success of platforms such as NetLogo [1], MASON [2] and Repast Simphony [3]. Similar features are provided by Model Driven Engineering platforms, such as INGENIAS [4] and modelling4all [5].

The drawback of this approach is scalability. Graphical tools and dynamic programming languages are not focused on this aspect of software engineering, and they tend to have serious efficiency issues. This is not relevant for simple models, but as the complexity of a model grows the idea of switching towards more efficient programming languages takes importance. This decision has a major impact in the model. Advanced programming skills are required for this task, as languages such as Java or C++ have a steep learning curve. Besides, this decision not always affect the programming language but also the platform chosen to create the ABM. In particular, the Application Programming Interface (API) of the different platforms can be extremely different, even when the programmer is replicating an existing model. Finally, the change of programming language is not enough improvement on efficiency when the models contain a large number of agents with complex behavior; in this case, the better approach is to choose a platform capable of distributing the execution of a simulation.

Parallel programming is one of the most challenging aspects of software development. In particular, the distribution of an ABM is strongly dependent on the nature of the problem to model and the properties of the system (degree of interaction between agents, complexity of behavior, etc.). There is no optimal way to distribute any kind of ABM, because their dynamics are extremely diverse. For this reason there are different strategies that could split an ABM execution amongst different computer nodes, but none of them will be universally optimal. The only possible solution is to provide a set of different techniques that can be applied to similar problems. This idea is the basis of different initiatives like GridABM [6], a framework of template solutions for distributing these type of simulations.

All this diversity of models and software platforms implies that any project involving ABMs must carefully evaluate existing options, considering its primary goals. If a wrong platform is chosen it will be difficult to fix the problem, and this could have a major impact on the success or failure of the project. This also applies when there is a high degree of uncertainty or requirements change over the span of the project, situations that can easily happen in the context of research. The Repast suite [7] is currently the only solution to this issue, as it offers a wide arrange of different tools to use,

ISBN: 978-1-61208-371-1

from graphical modelling to distributed executions. However this is not an optimal solution, as the most advanced tool in terms of computation (Repast-HPC) does not share the same code or language than the rest.

The discussed issues can be transformed in a set of requirements for an all-purpose ABM platform:

- Rapid prototyping in a dynamic programming language
- An alternate interface providing access to efficient version of the same functionality
- The user should be able to analyze their models with a wide array of analytical tools, including spatial analysis, statistics and visualization.
- Parallel execution, not only of different runs but also of a single, CPU-demanding, simulation.
- The switch between sequential and distributed executions should not translate into a re-implementation of the model or in learning the complexities of parallel programming.

This paper presents Pandora, an open-source ABM framework designed to deal with the varied needs of modellers and fill the gap between prototyping and advanced simulations. This innovative platform has a flexible architecture capable of providing the tools needed to create any type of ABM and execute it in any environment in a transparent way. This includes dynamic prototyping, automated parallel execution and analytical tools.

Next Section describes the different tasks involved in the creation of an ABM, that any framework should cover. Next, we describe the general structure of the platform, before focusing on the core concepts of the design fulfilling the requirements. The paper closes with a discussion on the present and future of the platform.

II. Creating an Agent-Based Model

The methodologies used to create agent-based simulations are as diverse as their goals [8]. The set of utilities and analytical tools needed to implement, explore and publish an ABM makes a strong case for using a general framework capable of providing functionality for all these tasks. This software platform should provide enough flexibility to go from exploratory models to predictive simulations using the same code. On the other hand social simulations are usually created inside interdisciplinary initiatives, and this has an important impact in the different steps of the methodology. The general work-flow for this development, independently of the platform, can be summarized as followed:

1) Definition of a research question
2) General definition of a model using Overview, Design Concepts and Details (ODD) protocol [9], Unified Modelling Language (UML) [10] or similar modelling tools)
3) Development of a prototype using a dynamic programming language
4) Exploratory visualization and verification of results
5) If needed, implementation of an efficient version of the model
6) Design and execution of a set of experiments
7) Analysis and dissemination of results

This classical modelling methodology is not the only way to create ABMs. Software engineering practices are increasingly being used in the field; for this reason, the framework should not force the user to create their models with a particular methodology, and it could be interesting to have support for agile methodologies, including the use of Test-Driven Development, intensive refactoring and collective ownership of the code of [11][12].

III. General Design

The general architecture of Pandora is shown in Figure 1. The abstract classes *World* and *Agent* are the core of the library, as they form the content of any model. The first one manages the different layers of continuous information that define the environment of the simulation. Following Geographical Information Systems modelling practices the environment is defined as a set of raster maps (bi-dimensional matrices of values), that can have read-only data and are encapsulated in instances of the class *StaticRaster*. They are complemented by *DynamicRasters*, whose values that can be modified over the span of the simulation. The class *Config* also provides the functionality needed to initialize any *ConcreteWorld*, usually from an eXtensible Markup Language (XML) file with all the parameters of a run (duration, size of the map, output directory, etc).

The *Agent* class encapsulates any entity of the model with internal state, decision making processes and behavior. Any *ConcreteAgent* needs to define at least a method *updateState* that is executed every time step as well as *serialize*, where its state will be stored in a file using the Hierarchical Data Format (HDF5) protocol [13].

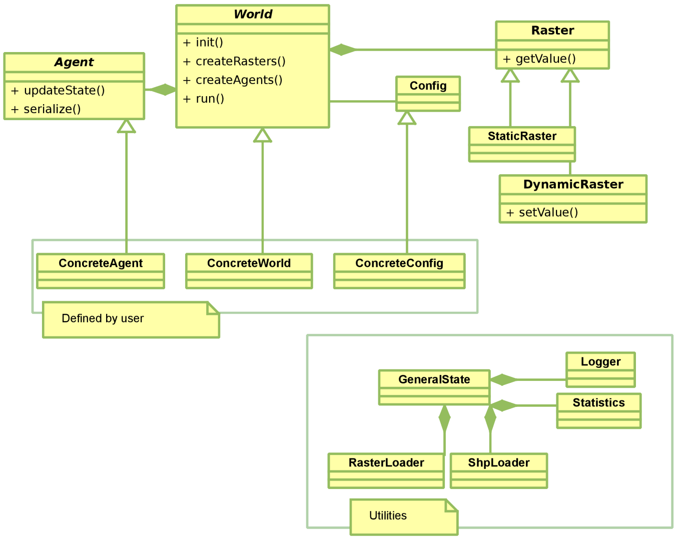


Figure 1. Class diagram of the framework

A set of different utilities and helpers provides functionality to load georeferenced data, log relevant information and generated pseudo-random numbers needed for stochastic processes. This architecture is specifically designed to host any potential ABM, from simple prototypes to complex and realistic models. Next sections define the different solutions that have been implemented to deal with the requirements, in particular (a) dynamic/static programming, (b) analytical tools and (c) scalability.

IV. Twin programming interface

One of the most important requirements for this platform is to facilitate the transition between simple and complex simulation environments. Pandora implements its functionality in C++, which can be accessed from a two-headed interface in C++ and Python. The first interface offers an efficient version that can also be distributed in a computer cluster, while the flexibility of Python allows any user with minimal programming skills to develop a model from scratch. The link between both interfaces uses the library boost-python, that provides a clear way of binding Python calls to C++ classes and functions.

As an example of the flexibility provided by this solution, a simple Agent that moves randomly can be defined in C++ as:

```
class BasicAgent : public Agent
{
public:
BasicAgent( const std::string & id ) : Agent(id) {}
~BasicAgent() {}

void updateState() {
    Point2D<int> newPosition = getPosition();
    newPosition._x += getUniformDistValue(-1,1);
    newPosition._y += getUniformDistValue(-1,1);

    if(getWorld().checkPosition(newPosition))
    {
        setPosition(newPosition);
    }
}
};
```

The same behavior is achieved in Python using this code:

```
class BasicAgent(Agent):
 def __init__(self, id):
  Agent.__init__( self, id)

 def updateState(self):
  newPosition = self.getPosition()
  newPosition._x += random.randint(-1,1)
  newPosition._y += random.randint(-1,1)

  if self.getWorld().checkPosition(newPosition):
   self.position = newPosition
```

Any code can be translated from one version to the other, as both languages are designed to implement the Object-Oriented paradigm and the interface to Pandora's functionality is exactly the same. An important side-effect of this solution is the maintainability of the framework. There is no duplicated functionality between version of the framework in different programming languages, as both interfaces call the same C++ code (i.e., complete examples can be found at [14]. This approach decreases the number of lines, thus diminishing the potential appearance of bugs in the code. In addition, both interfaces are checked by two unit testing suites, one using the basic python testing framework and the other one with the boost unit test toolbox. Finally, the analytical tools are common for both interfaces, as models implemented with any of the languages serializes result files with identical format, as they are calling the same serialization methods.

V. Multiple schedulers

ABMs are developed to explore a wide array of research questions. In essence any scenario where micro behaviors generate macro dynamics can be modelled with this technique. At the same time, there is a key component of any ABM that must be present in all possible models: the scheduler. This is the process that updates the set of agents and the environment that together form the model, and manages issues like the order of execution of the agents, and the way they interact with the other components (i.e., other agents, layers of raster maps, etc.). The heterogeneity of problems to be modelled suggest that there is no optimal scheduling algorithm. For example, a scheduler that proofs efficient in the sequential execution of simple models in a laptop will probably be incapable of managing distributed simulations with millions of agents. Besides, the way agents interact with each other is quite different, depending on the purpose of the model. For example, agents living in spatially structured models will have a completely different interaction with other entities than a model guided by social networks. The consequence of the flexibility of the ABM concept is that there is no optimal general strategy for a platform's scheduler.

The relevance of this issue grows exponentially when dealing with distributed executions [15]. Each agent needs to gather knowledge from the surrounding environment, including raster maps and other agents, before making any decisions. The agents then execute their decision-making processes and modify the environment. These mechanics translate, in terms of parallelization, in the need of sharing several layers of environmental data and agents between computer nodes, at every single time step. Furthermore, the execution of the agents' actions cannot usually be distributed within a single computer node (i.e., OpenMP), because there can be several conflicts related to agents accessing and modifying the same data at the same time.

Pandora's design follows the philosophy of versatility, combining the software engineering patterns *bridge* and *factory method* [16], as shown in Figure 2.

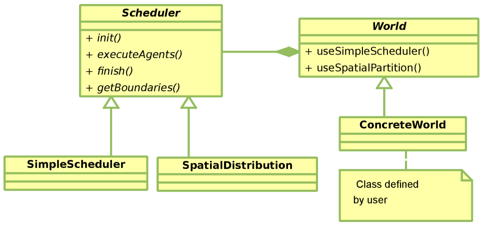


Figure 2. Class diagram of the Scheduling system

Any functionality of the base class World that needs to update the environment or the agents is delegated to the class Scheduler. Custom Schedulers implement different ways to manage these tasks without implying a change in the Application Program Interface (API) of the framework. The user only needs to specify the preferred Scheduler in the code. This is done as follows for multithreaded execution in a single computer (using OpenMP):

```
ConcreteWorld world(config,world.useOpenMPSingleNode());
```

or spatial distribution in different computer nodes using Message Passing Interface (MPI) and OpenMP:

```
ConcreteWorld world(config,world.useSpacePartition());
```

The solution also provides a clear interface for advanced users to develop their own schedulers if they need particular solutions for their models. In any case, the *Bridge* pattern allows that any Scheduler can be used in the execution of any simulation. This is particularly useful for exploring a model in different scales, as quick scenarios will use multithreading while the largest ones can be deployed in a High-Performance Computing (HPC) infraestructure.

A. Parallel execution

The only current Scheduler of Pandora designed for distributed execution is based on spatial partition. Each computer node owns a section of the entire simulated scenario, containing the different landscape data as well as the agent. This is one of the most popular ways of distributing an ABM, as is also the solution adopted by Repast-HPC [7].

The world of the simulation is evenly divided among computer nodes, and each one of them owns a section of the environment, as well as the agents located inside its boundaries. This layout is depicted in Figure 3. Information in the border between adjacent nodes (raster maps and agents) is communicated to neighbours every time step execution, in order to keep up-to-date data in the entire scenario. The size of this buffer border is defined as the maximum interaction range of any agent, being the absolute horizon of actions of any component of the simulation. The solution is scalable, given the fact that every computer node will need to communicate, at most, with 8 neighbouring nodes (if nodes own rectangular regions), independently of the total size of the simulation. On the other hand communication must be local, as agents can only communicate inside a given interaction range.

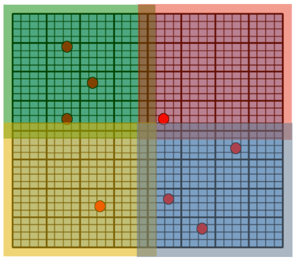
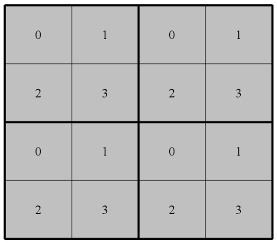


Figure 3. Spatial partitioning of an ABM. Each color represents the section of the world owned by a different computer node.

This solution does not solve to problem that two agents *living* in different computer nodes can modify the same bordering data at the same time. This potential collision can be avoided by different technique, but most of them can be computationally intensive (e.g., rollbacks). This overhead is affordable if agent behavior is CPU intensive and the possibility of conflict is low, but this is seldom the case with ABMs.

Pandora's spatial partition scheduler uses a simpler approach. The spatial section owned by a computer node is split into four equal parts numbering 0 to 3, as seen in Figure 4. The agents inside 0 sections are executed simultaneously; as they are not adjacent there is no possibility of collision between their actions. Once all 0's are finished, modified border data is sent to the neighbors, and a new section will begin its execution (1, 2 and finally 3). Once all of them are executed, the entire state of the simulation is serialized and a new time step can be evaluated.



0	1	0	1
2	3	2	3
0	1	0	1
2	3	2	3

Figure 4. Each computer node is divided into four different sections, that are executed sequentially

The pitfall of this solution is that agents in section 0 will always be executed before agents in sections 1-3. Depending on the model the consequences of this effect can be nonexistent, or introduce artifacts in the outcome. As usual, a careful choice between the different strategies is needed, based on the existing scenario; Pandora provides the way to implement and use any existing algorithm designed to distribute ABMs [17][18][19].

B. Simultaneous execution of agents

Simulation performance can also be increased using the complete set of CPU cores of every computer node to simultaneously execute the agents. Again, the problem of collisions between agents' actions must be solved. Performance analysis showed that most of the execution time is spent when agents (1) gather information, (2) choose a particular set of behaviors, and (3) execute them. All ABM platforms mix these phases in a single method, executed by every agent every time step: *tick* in Netlogo [1], *step* in MASON [2] and RePast [20].

Agents do not modify anything in phases (1) and (2), as they just evaluate potential course of actions depending on existing data; if we separate them from the action's executions they can be simultaneously executed without risk of collisions.

Pandora uses this approach to split the step of an agent in three different methods. In the first one, *updateKnowledge*, an agent cannot modify the environment or other agents; it only gathers information. In the second one, *selectAction*, the agent executes her decision-making process to chooses an action Once every agent has chosen what she wants to do Pandora executes the actions of the agents sequentially. Finally, the third method that a user can specify is *updateState*, where any agent can modify its internal state evaluating the results of her actions. This cycle *Explore* - *Decide* - *Apply* allows Pandora to distribute the execution amongst different Central Processing Unit (CPU) cores of a node, as the first two methods are declared *const* in C++ to be thread-safe, while the third one is executed sequentially.

This structure seems more complicated than just defining one method but, from a theoretical point of view, the division of an agent's execution in these three phases is more consistent than the traditional ABM approach. The single method implementation mixes the different stages of an agent's cycle, that should be correctly specified while building the model (see Figure 5). Dividing the execution in these phases avoids this issue, forcing coherence during the transition from theory to code.

Finally, from a theoretical point of view this solution is more elegant, as it matches the definition of an intelligent agent [21].

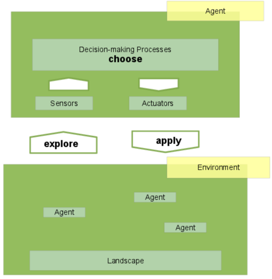


Figure 5. The execution cycle of an agent

To conclude, a performance analysis of Pandora's scheduling system [15], both in (a) supercomputers and (b) cloud HPC infraestructures. Results show again the diversity of challenges that any ABM platform need to face, as its execution in (b) is optimal when the balance between intensity of communication and CPU needs are shifted towards the later, while (a) is needed if the model has a high degree of interaction.

VI. Analysis

Any ABM framework is not complete without a set of tools to assist experiment design and analysis. Pandora's development team has created Cassandra [22], a GUI tool to help the user to perform the required tasks, from exploring a single run to the exploration of parameter space across thousands of executions.

A. Single run examination

The graphical interface allows to visualize the spatiotemporal dynamics of a model, as can be seen in Figure 6. This feature is useful to detect general patterns, as the user can check and track, at any given moment, the state of the different agents and layers of information.

B. Parameter space exploration

The *Laboratory* tool allows to define and run an experiment, based on the values that the used defines for each input parameter specified in the configuration. An HDF5 file is generated by each run, that are parsed in order to extract the needed summary statistics for the analysis.

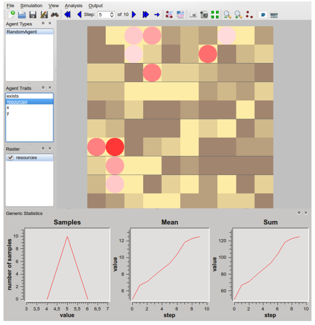


Figure 6. Spatial visualization of a single run

C. Exploratory Data Analysis

The output produced by thousands of ruins of an stochastic ABM can be extremely complex to study, and visual analytics can be an optimal choice to explore preliminary results [23]. At the present Cassandra includes two interactive tools: (a) heat maps to compare particular parameter values and (b) time series for understanding temporal dynamics, as seen in Figure 7.

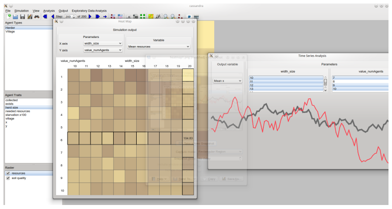


Figure 7. Exploratory visualization toolbox depicting the same set of runs with a heatmap and time series

D. Output

The final step is the use of the analytical toolbox to collect data from the dynamics of the system in different formats that can be used by different applications (i.e., Comma Separated Values for Statistical packages, GeoTiff and Shapefiles for Geographical Information Systems, etc.). Additional outputs ready to be used in publications and reports are composite mosaics encapsulating several runs in any video format, as well as Google Earth georeferenced movies.

VII. Conclusions and Future Work

The popularization of social simulation has increased in recent years the number and features of open-source ABM platforms [24]. This trend, while common and positive in any scientific software community, has been one of the reasons why replicability is scarce [25]. This issue is combined with a limited use of software engineering to control the quality of scientific code [26].

In addition, there is a technical gap between exploratory models created for theoretical research and realistic models developed for hypothesis testing and prediction. The first class of ABMs are implemented by social scientists in platforms with little or non-existent capability for multiplatform execution, while the second type of models is created by programmers with advanced programming skills in other software packages, and deployed in powerful hardware infrastructures. The paradox is that simple simulations cannot be distributed without a major coding effort, while distributed simulations are not easily executed in a standard computer. Pandora closes this gap in the increasingly diversified environment of social simulation; it provides enough flexibility to be extended by any user requirement, while maintaining its role of an all-purpose platform that can be used with any kind of model and/or infraestructure.

This versatility has been shown in a wide array of scenarios, from simple models exploring theoretical issues [27] to realistic simulations executing complex behavior for thousands of agents and parameter configurations [28][29]. Its use has also been extended through additional functionality like the use of Graphics Processing Unit (GPU) acceleration for particular agent actions [30], advanced decision-making processes using Markov Decision Processes [27] or its deployment in cloud HPC infraestructures [15]. Next steps include the development of a network-based scheduler, binary installation packages and additional visualization tools.

VIII. Acknowledgements

This research is part of the SimulPast Project (CSD2010-00034) funded by the CONSOLIDER-INGENIO2010 program of the Ministry of Science and Innovation Spain. We would like to thank the past and present member of the development team, and the anonymous reviewers for their comments. Pandora currently works under Linux and OS X operating systems and can be downloaded from http://xrubio.github.io/pandora.

References

[1] U. Wilensky, "Netlogo," Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999, retrieved: September, 2014. [Online]. Available: http://ccl.northwestern.edu/netlogo/

[2] S. Luke, "Multiagent simulation and the mason library," George Mason University, August 2011, retrieved: September, 2014. [Online]. Available: http://cs.gmu.edu/ eclab/projects/mason/manual.pdf

[3] M. J. North, N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko, "Complex adaptive systems modeling with Repast Simphony," Complex Adaptive Systems Modeling, no. 1:3, 2013.

[4] J. Pavón and J. Gómez-Sanz, "Agent oriented software engineering with ingenias," in Multi-Agent Systems and Applications III. Springer Berlin Heidelberg, 2003, pp. 394–403.

[5] K. Kahn and H. Noble, "The modelling4all project a web-based modelling tool embedded in web 2.0," in Proceedings of the 2Nd International Conference on Simulation Tools and Techniques, ser. Simutools '09. ICST, 2009, pp. 50:1–50:6.

[6] L. Gulyás, A. Szabó, R. Legéndi, T. Máhr, R. Bocsi, and G. Kampis, "Tools for large scale (distributed) agent-based computational experiments." in Proceedings of the Computational Social Science Society of America Annual Conference 2011, 2011.

[7] N. Collier and M. North, "Repast hpc: A platform for large-scale agent-based modeling," in Large-Scale Computing Techniques for Complex System Simulations, W. Dubitzky, K. Kurowski, and B. Schott, Eds. Wiley, 2011, pp. 81–110.

[8] B. Heath, R. Hill, and F. Ciarallo, "A survey of agent-based modeling practices (january 1998 to july 2008)," Journal of Artificial Societies and Social Simulation, vol. 12, no. 4, 2009, p. 9, retrieved: September, 2014. [Online]. Available: http://jasss.soc.surrey.ac.uk/12/4/9.html

[9] V. Grimm, U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske, and S. F. Railsback, "The odd protocol: A review and first update," Ecological Modelling, vol. 221, no. 23, 2010, pp. 2760–2768.

[10] H. Bersini, "Uml for abm," Journal of Artificial Societies and Social Simulation, vol. 15, no. 1, 2012, p. 9, retrieved: September, 2014. [Online]. Available: http://jasss.soc.surrey.ac.uk/15/1/9.html

[11] R. Mugridge, "Test driven development and the scientific method," in Agile Development Conference, 2003. ADC 2003. Proceedings of the, June 2003, pp. 47–52.

[12] M. Sletholt, J. Hannay, D. Pfahl, and H. Langtangen, "What do we know about scientific software development's agile practices?" Computing in Science Engineering, vol. 14, no. 2, March 2012, pp. 24–37.

[13] M. Folk, A. Cheng, and K. Yates, "Hdf5: A file format and i/o library for high performance computing applications," in In Proceedings of the 12th Conference on Supercomputing, Portland, November 1999.

[14] "Pandora: an agent-based modelling framework for large-scale distributed simulations," http://xrubio.github.io/pandora, retrieved: September, 2014.

[15] P. Wittek and X. Rubio-Campillo, "Scalable agent-based modelling with cloud hpc resources for social simulations," in Proceedings of the 4th International Conference on Cloud Computing Technology and Science, 2012, pp. 355–362.

[16] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-oriented Software. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[17] M. Scheutz and P. Schermerhorn, "Adaptive algorithms for the dynamic distribution and parallel execution of agent-based models," Journal of Parallel and Distributed Computing, vol. 66, no. 8, 2006, pp. 1037–1051.

[18] D. Chen, G. K. Theodoropoulos, S. J. Turner, W. Cai, R. Minson, and Y. Zhang, "Large scale agent-based simulation on the grid," Future Generation Computer Systems, vol. 24, no. 7, 2008, pp. 658–671.

[19] M. Kiran, P. Richmond, M. Holcombe, L. S. Chin, D. Worth, and C. Greenough, "Flame: simulating large populations of agents on parallel hardware architectures," in Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 1633–1636.

[20] M. North, T. Howe, N. Collier, and J. Vos, "A declarative model assembly infrastructure for verification and validation," in Advancing Social Simulation: The First World Congress, S. Takahashi, D. Sallach, and J. Rouchier, Eds., 2007.

[21] S. J. Russell and P. Norvig, "Intelligent agents," in Artificial Intelligence: A Modern Approach, 2nd ed. Pearson Education, 2003.

[22] "Cassandra: a lab for agent-based model analysis," https://github.com/xrubio/pandora/tree/master/cassandra, retrieved: September, 2014.

[23] J. J. Thomas and K. A. Cook, "A visual analytics agenda," IEEE Comput. Graph. Appl., vol. 26, no. 1, Jan. 2006, pp. 10–13.

[24] C. Nikolai and G. Madey, "Tools of the trade: A survey of various agent based modeling platforms," Journal of Artificial Societies and Social Simulation, vol. 12, no. 2, 2009, p. 2, retrieved: September, 2014. [Online]. Available: http://jasss.soc.surrey.ac.uk/12/2/2.html

[25] M. Lake, "Trends in archaeological simulation," Journal of Archaeological Method and Theory, vol. 21, no. 2, 2014, pp. 258–287.

[26] D. C. Ince, L. Hatton, and J. Graham-Cumming, "The case for open computer programs," Nature, vol. 482, no. 7386, 02 2012, pp. 485–488.

[27] P. Wittek, I. Lim, and X. Rubio-Campillo, "Quantum probabilistic description of dealing with risk and ambiguity in foraging decisions," in Quantum Interaction, ser. Lecture Notes in Computer Science, H. Atmanspacher, E. Haven, K. Kitto, and D. Raine, Eds. Springer Berlin Heidelberg, 2014, pp. 296–307.

[28] X. Rubio-Campillo, J. Cela, and F. Hernàndez, "Simulating archaeologists? Using agent-based modelling to improve battlefield excavations," Journal of Archaeological Science, vol. 39, 2012, pp. 347–356.

[29] X. Rubio-Campillo, J. M. Cela, and F. Hernàndez, "Development of new infantry tactics during the early eighteenth century," J Simulation, vol. 7, no. 3, Aug 2013, pp. 170–182.

[30] P. Wittek and X. Rubio-Campillo, "Social simulations accelerated: Large-scale agent-based modeling on a gpu cluster," in GPU Technology Conference, San Diego, 2013, poster DD06.

Interface-based Semi-automated Generation of Scenarios for Simulation Testing of Software Components

Tomas Potuzak
Department of Computer Science and Engineering
Faculty of Applied Sciences, University of West Bohemia
Univerzitni 8, 306 14, Plzen, Czech Republic
e-mail: tpotuzak@kiv.zcu.cz

Richard Lipka
Department of Computer Science and Engineering/
NTIS – European Center of Excellence
Faculty of Applied Sciences, University of West Bohemia
Univerzitni 8, 306 14, Plzen, Czech Republic
e-mail: lipka@kiv.zcu.cz

Abstract—**Using the component-based software development, an application is constructed from individual reusable components providing particular functionalities. The testing of their functionality, their quality of services, and their extra-functional properties is important, especially when each component can be created by a different manufacturer. We developed the SimCo simulation tool, which enables simulation testing of software components directly without the necessity to create their (potentially incorrect) models. The course of the testing is described in so-called scenarios. In this paper, the semi-automated generation of the scenarios based on the analysis of the interfaces of the particular software components and their mutual interactions is described. The described ideas demonstrate the usefulness of the simulation for the component-based software development.**

Keywords-software components testing; scenarios generation; component interface; component interaction.

I. INTRODUCTION

The component-based software development is an important trend in contemporary software engineering. Using this approach, an application is constructed from individual reusable software components, which provide particular functionalities as services through their public interfaces. Each service of a component can be utilized by multiple other components of the application and the invocation of a service is a basic interaction between the components. A single component can be utilized in multiple applications, which implies the reuse of the existing program code. At the same time, an application can be constructed from components, which were created by different developers [1].

The testing of software components is very important. Their functionality should be ensured by their developers. However, the potentially different developers of the particular components of a component-based application reinforce the need for testing of the interactions and mutual cooperation of the components. Moreover, besides the functionality of the components, it is also necessary to test their extra-functional properties (e.g., memory consumed by a service invocation) and quality of services (e.g., time required for a service invocation or the performance in general) [2][3].

During our previous research, we have developed the SimCo simulation tool, which enables the simulation testing of a single software component, a set of software components, or an entire component-based application [4] [5]. The components can be tested in the SimCo directly, without the necessity to create their (potentially incorrect) simulation models [5]. The course of the simulation testing is described in so-called *scenarios*, which are basically XML files loaded by the SimCo. The manual creation of the scenarios is a lengthy and error-prone process [6]. Therefore, we are now working on several approaches, which would provide a semi-automated generation of the scenarios.

In this paper, we describe one such approach, which is based on the analysis of the public interfaces of the particular software components and on the observation of their mutual interactions. The approach is focused on the situation when the software components are created by a third party and their source codes or any other descriptions of their behavior are unavailable to us.

The paper is structured as follows. The basic notions are briefly discussed in Section II. The SimCo is described in Section III. The existing approaches to generation of testing scenarios are discussed in Section IV. The interface-based scenarios generation, which is the main contribution of this paper, is described in Section V and demonstrated on a case study in Section VI. The paper is concluded in Section VII.

II. BASIC NOTIONS

In order to make the further reading more clear, the basic notions of the component-based software development and the simulation testing will be now briefly described.

A. *Software Components*

In the component-based software engineering, a software component is a black box entity with a well defined public interface and no observable inner state. Its particular functionalities are provided as services accessible via its public interface [1]. An application is then constructed from multiple components, which interact using their interfaces (e.g., invoking particular services). The particular components of an application can be created by different developers, which can be also different from the developer of the resulting application. At the same time, a component can be utilized in multiple applications. Although this definition is very abstract, a more specific definition is problematic, since the view of software components is very diverse [1].

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-371-1

A component model specifies the features, behavior, and interactions of software components. A component framework is then a specific implementation of a component model [5]. So, there can be (and often are) multiple implementations of a single component model [7]. Software components of different component models often have very different form. Hence, it is very difficult to transform an existing software component from one component model to another. Thus, the SimCo utilizes only one component model – the OSGi [8][9]. However, the ideas behind it can be used in other component models as well.

The OSGi is a dynamic component model for Java programming language widespread in both the academic and the industrial spheres including automotive industry, cell phones, and so on [7]. There are multiple implementations of the OSGi component model (i.e. OSGi component frameworks) [8], for example the Equinox [10]. For its widespread use, the OSGi is used in the SimCo [5].

Using the OSGi, the particular software components are referred to as *bundles*. The dynamic nature of the OSGi means that it is possible to install, start, stop, and uninstall any bundle without the necessity to restart the OSGi framework assuming that all bundle dependencies are satisfied [11]. The OSGi bundle has the form of a `.jar` file with additional OSGi-specific meta-information regarding the provided and required services and so on [9]. Since the `.jar` file can contain any number of Java classes, each bundle can provide arbitrary complex functionality [7]. Both provided and required services are described by standard Java interfaces. Hence, the particular services of a bundle have the form of standard Java methods [8].

B. Discrete-Event Simulation

A discrete-event simulation is a widely used simulation type [12]. The simulation run of a discrete-event simulation is divided into sequence of time-stamped events, which represent an incremental change to the simulation state. The time stamp specifies the simulation time, in which this change to the simulation state should happen [12].

The simulation time between two succeeding events can be arbitrary long, but the changes of the simulation state are associated with the events only. So, the simulation time “jumps” from the time stamp of an event to the time stamp of the next event [12]. The processing of events is handled by a so-called *calendar* with the list of events ordered by their time stamps. At the start of a simulation run, the calendar removes the first event from the event list, sets the simulation time to the value of the event’s time stamp, and performs the event’s change of the simulation state. This can potentially lead to the creation of multiple new events, which are inserted to the event list on the positions corresponding to their time stamps. Then, the next event is removed from the event list and so on. The entire process stops when a predefined condition is fulfilled or the event list is empty [12].

C. Simulation Testing

Considering only the discrete-event simulation from now on, we will describe the simulation testing of software components. For this purpose, two approaches can be used.

The first approach is to develop a model of the tested software component and use it in the simulation [2]. The advantage of this approach is that the created model is suited for the simulation. The disadvantage is that the creation of the model is time-demanding and potentially error-prone. Since the model usually incorporates only features, which are considered important for the simulation and testing, it is also possible to unintentionally omit some features, which in fact are important for the results of the testing [7]. Also, the consistency of the model has to be maintained.

The second approach is to use the tested software component directly in the simulation, instead of using its model. The advantage of this approach is that there is no necessity to create the model [4]. The disadvantage is that the simulation must enable the running of the tested component in a way it would run in a non-simulation environment [7].

Regardless the approach, the simulation testing can be divided according to the knowledge of the tested software component. If its source code is known, it can be (and usually is) used for the preparation of the course of testing. This is a *white box* testing. If the source code is not known, other information such as a specification of the component’s intended behavior, a description of its interface, and so on can be used for the preparation of the course of testing. This is a *black box* testing [13], on which this paper is focused.

D. Testing Scenarios

Regardless its type, the simulation testing lies in the subjecting of the tested software component (or its model) to a set of stimuli and observation of its corresponding reactions [14]. In most real cases, it is not feasible to provide a complete coverage of all stimuli. Hence, a subset of stimuli should be created instead. This subset should represent well all theoretical possible stimuli and, at the same time, be reasonably small [13].

The course of the simulation testing is described in so-called *scenarios*, which incorporate the particular stimuli and (optionally) the expected consequences. Using the discrete-event simulation (see Section II.B) for the simulation testing of software components, the stimuli correspond to particular events. Each event, in turn, corresponds to the invocation of a service of a tested software component [7]. The consequences can then be a return value, an exception, a subsequent invocation of a service of another component, and so on.

III. SIMCO SIMULATION TOOL

Now, as we discussed the basic notions, we can proceed with the description of the SimCo, which we developed during our previous research.

A. Purpose and Description of SimCo Simulation Tool

The SimCo enables testing of software components using the discrete-event simulation (see Section II.B). The SimCo itself is a component-based application, which enables its simple extensions and modifications [15]. It utilizes the OSGi component model [8][9]. Currently, it runs in the Equinox OSGi framework [10] (see Section II.A). Consequently, it is utilizable only for the testing of the OSGi bundles (OSGi software components) [7].

The tested components are directly incorporated into the simulation. Hence, the simulation environment must be constructed in a way, which enables a normal running of the tested components as they would be in a non-simulated environment. Among other things, the dependencies of the components must be satisfied. At the same time, the course of the simulation testing must be under control during the entire simulation run. These requirements are reflected in the structure of the SimCo, which consists of several software component types [15] (see Section III.B).

B. Types of Software Coponents in SimCo Simulation Tool

There are four types of software components (OSGi bundles) in the SimCo – the *core*, the *real tested*, the *simulated*, and the *intermediate* components [15].

The core components provide the functionality of the SimCo itself. Primarily, they ensure the running of the simulation according to the loaded testing scenario and the collection of the results. The most important core component is the `Calendar`, which ensures the progress of the simulation run by processing the events from the event list. The core components also provide supplementary functions such as logging, visualization of the simulation, and so on [15].

The real tested components are the OSGi bundles, which shall be tested using the SimCo. The source code of these components may be known or unknown, since the components can be created by a third party [15]. If the source code is known (i.e. white box testing) it can be used for the creation of the testing scenarios. However, in this paper, we focus on the case when the source code is not known (i.e. black box testing) and only the public interfaces of the components (with or without further description) are known. The real tested components run in the simulated environment, which brings several issues (e.g., discrepancy of the simulation and real time, undesirable remote communication, etc.) solvable by various means (see [5] and [15]).

The simulated components provide services, which are required by the real tested components. They are useful when a single component or a limited set of components are tested and these components require services of other components, which are not present in the simulation. Then, the simulated components provide the missing services required by the real tested components. When an entire component-based application is tested, the simulated components are not necessary, since the requirements of all real tested components should be satisfied. Another important function of the simulated components is the ensuring that all invocations of the services of these components are performed using the events and the calendar. This allows the SimCo to maintain the control of the simulation run. Additionally, the simulated components can be used for the speedup, since they do not have to perform all the calculation they real counterparts would. Instead, they can utilize random numbers generators or prerecorded values, depending on the situation [15].

The intermediate components are inserted between all pairs of real tested components. They ensure that the invocations on the real tested components are handled using the events and the calendar. Basically, an intermediate component is a proxy for a real tested component. It has the same public interface (i.e. provides the same services). All invocations of the services on the real tested component are in fact handled using this proxy (see Figure 1). When a service is invoked, the intermediate component ensures the use of the events and the calendar and invokes the corresponding service on the real tested components to obtain the return value and potentially trigger further consequences (e.g., invocation of a service of another component) [15].

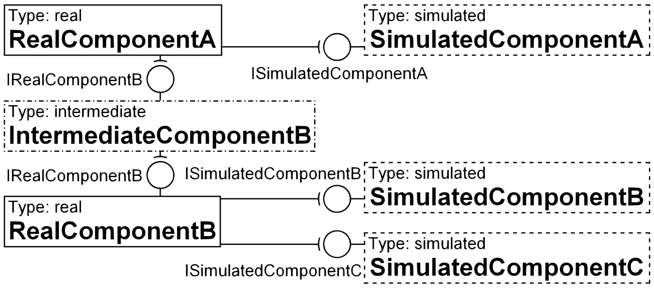


Figure 1. Relations of the three types of SimCo components

C. Testing Scenarios in SimCo Simulation Tool

The testing scenarios for the SimCo have the form of XML files. They contain primarily the events, which shall be processed during the simulation run [15]. The events correspond to the invocations of the particular services of particular components (see Section II.D). The scenarios also contain the configuration of the simulation and the layout of the components (e.g., information, which components are real tested, which are simulated, etc.).

Each event in the testing scenario is usually stored as a unique record containing the time stamp and the description of the invocation that shall be performed – the component, on which the invocation shall be performed, the service, which shall be invoked, and the parameter values of the invocation. The expected consequences of the invocation can be stored as well. They are useful for evaluation of the simulation tests – it can be easily compared whether the consequences of the invocation during the simulation run are the same as the expected ones. This is particularly useful during the testing of the correct functionality of the components. Nevertheless, the expected consequences are not an obligatory part of the testing scenarios, since it is also possible to use logging and probes for evaluation of the simulation tests, separately from the scenarios. This can be useful during the testing of the extra-functional properties and quality of services of the components.

As an alternative to describing every single event separately, it is also possible to specify a random numbers generator with an appropriate probabilistic distribution [15]. This way, it is possible to generate multiple events (with preset parameter values or even randomly generated parameter values) with a single description within the testing scenario. However, this approach can be used only in some cases.

IV. Scenarios Generation Approaches

The idea of the semi-automated generation of the testing scenarios is not new in the software development field. The research in this area begun at least in 1970s [16] and continues till today [6][13][14][17]. However, the research focused on the semi-automated generation of the testing scenarios directly for the software components is rare. So, several existing approaches, which are briefly discussed in the following sections, are mostly not intended for the software components, but still can serve as a source of inspiration. They have in common that they do not utilize the source code of the tested application, but rather a description of its behavior and requirements.

A. UML Behavioral Diagrams

Many approaches for the generation of the testing scenarios are based on UML behavioral diagrams (e.g., sequence diagram, activity diagram, etc.) [18].

The activity diagrams are used, for example, for a representation of concurrent activities [13]. The exhaustive exploration of the diagrams is then used for the generation of the testing scenarios. Since, for large applications, the exploration of all possible flows in the diagrams is infeasible, some constraints derived from the application domain are used to discard illegal or irrelevant scenarios [13]. The activity diagrams are also used in [17] where they are generated using multiple UML use case diagrams in order to express the concurrency of the particular use cases. The exploration of the diagrams is again used for the generation of the scenarios [17]. In [19], a layered activity diagram describing the workflow of the tested application is used for the extraction of the execution paths of the tested application. These execution paths are then used for the generation of the scenarios [19].

B. User Interface

Several approaches utilize the user interface for the generation of the testing scenarios. This is based on the assumption that the user interface provides access to the majority (if not all) functions of the tested application [20].

In [20], the testing scenarios are created by inducing inputs to the user interface and pairing them with corresponding outputs. An opposite approach, which is focused on the testing of software components created using the .NET platform, is described in [21]. In this case, the absence of the user interface of the components is stressed as a major setback for the testing. Hence, reverse engineering is used for determination of the classes, methods and attributes of the particular components. Using this data, a basic graphical user interface is generated for each component [21].

C. Natural Language Specification

Using the specification of the tested application written in natural language is an appealing approach for creating of the testing scenarios. However, this approach is still very difficult to implement due to the ambiguity, poor understandability, incompleteness, and inconsistency of the natural language [22]. Hence, some restrictions are often used to overcome these difficulties.

An example can be found in [23] where a restricted form of natural language is used for descriptions of the use cases. From these descriptions, a control-flow-based state machine is generated for each use case. Then, these state machines are combined into a global system level state machine. The testing scenarios can be then generated by exploration of this state machine [23].

A similar approach, based on the use case descriptions written in natural language, is considered for the SimCo as well. However, it employs different restrictions and different course of generation of the testing scenarios. The use-case descriptions are analyzed and transformed into an overall behavioral automaton (OBA) using the FOAM tool [24]. From this automaton, the testing scenarios can be generated [7]. However, the use cases descriptions must be written using the rules described in [25] and (manually) enriched by annotations describing the flow of the program and its temporal dependencies. For more details, see [7].

V. Interface-based Scenarios Generation

The approach to semi-automated generation of the testing scenarios, which is described in this paper in detail, is based on the analysis of the interfaces of the particular software components and observation of their interactions. The basic idea has been described already in [7], but its extension, the formulation of the algorithm, and its specific application using a case study is the main contribution of this paper.

A. Main Ideas, Features, and Limitations

The approach is intended for the situation when the source code of the particular software components is not known, but their implementations are at our disposal. The generated scenarios can then be used for testing of the extra-functional properties and the quality of services of the components. The specific requirements (e.g., maximal duration of an invocation) are not generated, but can be easily added by the user. The generated scenarios are also particularly useful when a new component shall replace its older version and we want to test whether the new component exhibits the same external behavior as the old one.

Assume now that there is an entire component-based application, for which the testing scenarios shall be generated, and the source code is not known for some or all of its components. This entire application can be then imported to the SimCo. During the import, the intermediate components are placed between each pair of the components, since all of them are real. Alternatively, some of the components can be replaced by their simulated counterparts as long as they exhibit the same external behavior as the original real components.

Once the import is complete, the public interfaces of the components (OSGi bundles) along with their services and their parameters are determined. Then, the parameter values of the particular invocations can be partially generated automatically and partially provided by a user. Using these parameter values, the services of the particular components are sequentially invoked and the consequences of the invocations are observed using the intermediate components (and simulated components if they are present). The consequences can be an exception, a return value, a subsequent invocation of a service or services on another component or other components, or solely a change of the

ISBN: 978-1-61208-371-1

inner state of the component. With the exception of the unobservable inner state, all consequences can be recorded.

From all the types of the consequences, the subsequent invocations are the most important ones. The reason is that, with each subsequent invocation of a service, its parameter values are observed. These parameter values can be then added to the parameter values, which were automatically generated or provided by the user (see previous paragraph), but, unlike them, the parameter values from the subsequent invocations are genuine, provided directly by the component, which invokes the particular service. Moreover, the parameter values can contain instances of objects, which are difficult to generate automatically. This useful feature is possible, because the parameters of the subsequent invocations are present in the components, but are hidden from us, since the source code is not known.

From the invocations and their consequences, the scenarios can be directly generated. It should be noted that, if there is an erroneous behavior in the component-based application (e.g., a subsequent invocation, which in fact should not occur), the errors will propagate into the generated scenarios. Therefore, the user is encouraged to check the generated scenarios. However, in most cases, we assume that the component-based application is working correctly.

B. *Algorithm*

The algorithm of our approach to the testing scenarios generation consists of three steps – the preparation of a tree data structure of all the components and all their services, the generation of invocations of particular services along with their parameter values, and the performing the particular invocations in order to determine their consequences and supplement additional invocations and their parameter values.

The tree data structure can be initiated easily using the analysis of the public interfaces of the software components. For this purpose, a service of the OSGi framework, which enables to determine the public interface (i.e. a Java interface – see Section II.A) of a component (OSGi bundle), can be used. With the Java interfaces of the particular components known, the Java reflection can be used for the determination of their services (i.e. Java methods – see Section II.A) with types of their parameters and return values.

For each service of each component in the initiated tree data structure, the list of invocations is added. Each invocation contains the set of parameter values. These parameter values can be partially generated and partially supplemented by the user, depending on the type of the parameters. For the `enum` type, all possible values including the `null` value are generated. For the `boolean` types, both possible values are generated. For the `char` type, all single-byte values are generated. For the number types, a set of representative values are generated (e.g., -128, -100, -10, -1, 0, 1, 10, 100, and 127 for the `byte` type). For an object type, only the `null` value is generated. If there are multiple parameters for a single service, all combinations are generated. In order to keep the number of various invocations feasible, the user can provide restrictions for the generation of the parameters. Moreover, the user can manually add parameters, which are difficult to be generated automatically (e.g., specific instances for an object type). The scheme of the tree data structure with filled lists of invocations is depicted in Figure 2.

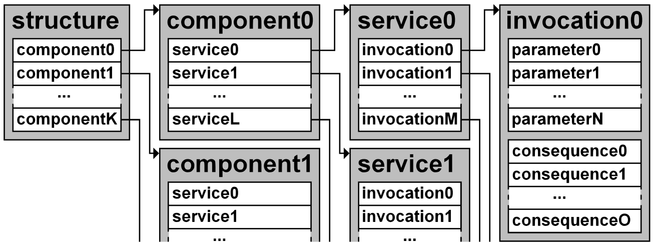


Figure 2. Scheme of the tree data structure

Once the tree data structure is initiated and filled with the invocations (see Figure 2), it is sequentially explored component by component, service by service, invocation by invocation. Each invocation from the tree data structure is performed on the corresponding component and its consequences are observed using the intermediate components (and the simulated components if they are present). These consequences (a return value, an exception, a subsequent invocation) are added to the tree data structure if they are not already present. If the consequence is a subsequent invocation on another component and it is not already present, this invocation along with its parameter values is added to the tree data structure as well.

```
//Initialization of the structure of components
structure.components = simco.getComponents();

//Initialization of their services
for (c: structure.components) {
  c.services = simco.getServicesOfComponent(c);

  //Generation and filling (by the user) of the
  //invocations
  for (s: c.services) {
    s.invocations = simco.generateAndReadInvocations();
  }
}

//Exploration of invocations and consequences
while (structure.isChanged()) {
  structure.setChanged(false);
  for (c: structure.components) {
    for (s: c.services) {
      for (i: s.invocations) {
        invocationConsequences =
          simco.performServiceInvocation(c, s, i);

        for (ic: invocationConsequences) {
          if (!i.consequences.contains(ic)) {
            i.consequences.add(ic);
            structure.setChanged(true);
          }
          if (ic.type == SUBSEQUENT_INVOCATION) {
            sc = ic.subsequentComponent;
            ss = ic.subsequentService;
            si = ic.subsequentInvocation;
            if (!structure.contains(sc, ss, si)) {
              structure.addInvocation(sc, ss, si);
              structure.setChanged(true);
            }
          }
        }
      }
    }
  }
}
```

Figure 3. The pseudocode of the entire algorithm

Once the invocation is completed and all its consequences are added to the tree data structure, the next invocation is performed and so on. When the entire structure is explored (i.e. all invocations were performed), it is checked whether new invocation consequences or subsequent invocations were added to the structure during the just finished exploration. If so, the exploration of the entire structure repeats. Otherwise, the algorithm ends (see Figure 3).

Once the algorithm is complete, its result is the tree data structure filled with the invocation consequences and the subsequent invocations. This filled data structure can be visualized as a tree (see Figure 2) or as a multilevel list (see Figure 5) for the user, if required. The structure can be then directly transcribed to the resulting scenario (an XML file). Should the extra-functional properties and the quality of services be tested, their descriptions must be added to the scenario by the user, since they cannot be determined from the components or they behavior without further description. The scenario can then be loaded by the SimCo, which performs the invocations with the parameters specified in the scenario and observes their consequences.

VI. Traffic Crossroad Control Case Study

The utilization of the described approach to the semi-automatic generation of testing scenarios will be demonstrated on a case study – the Traffic crossroad control.

A. Description of Traffic Crossroad Control

The Traffic crossroad control is a component-based application for the control of road traffic in a crossroad using traffic lights. It is expected to run in an OSGi framework on a specific hardware and operate a variety of hardware sensors and control units [5]. Since these sensors and control units are irrelevant for this paper, the components involving any direct contact with them were replaced by manually created simulated components in order to enable the running of the application on a standard desktop computer [7].

The scheme of the entire application is depicted in Figure 4. The application consists of eight components, from which four components are real and four components are simulated. Moreover, each real component has its own intermediate component (see Figure 4).

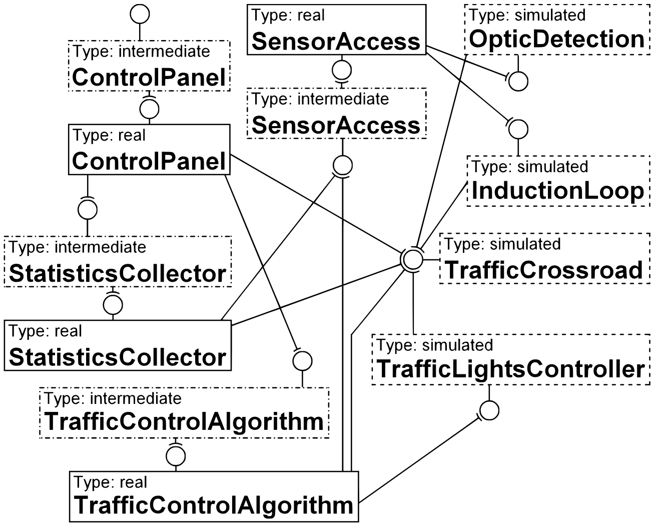


Figure 4. The scheme of the Traffic crossroad control application

The `TrafficCrossroad` component provides the information about the structure of the crossroad. Its simulated version also incorporates a nanoscopic road traffic simulation replacing the real traffic [7]. The `InductionLoop` and the `OpticDetection` components provide measured data from corresponding hardware sensors. Their simulated versions extract the data from the road traffic simulation instead. The `TrafficLightsController` component ensures desired settings of the traffic lights using corresponding hardware control units. Its simulated version sets the traffic lights in the road traffic simulation instead. The `ControlPanel` component provides the user interface for the entire application. The `TrafficControlAlgorithm` component contains an algorithm for the control of the traffic lights. This algorithm can require information from the sensors mediated by the `SensorAccess` component. From this component, the `StatisticsCollector` component periodically collects data and provides various road traffic statistics [7].

B. Scenarios Generation for Traffic Crossroad Control

Assume now the utilization of the algorithm described in Section V.B on the components of the Traffic crossroad control application. During the initiation of the tree data structure, the user provides eight sets of parameter values for two services of the `SensorAccess` component (string IDs of the particular sensors) and eight sets of parameter values for three services of the `TrafficControlAlgorithm` component (instances with parameters of the traffic control algorithm). The remainder of the parameter values is generated. The tree data structure is then explored and the invocation consequences and subsequent invocations are added to this structure.

The parts of the resulting filled structure are depicted in Figure 5. The components, services, invocations, and invocation consequences are marked with C, S, I, and IC, respectively. The value in the parentheses of I denotes the originator of the invocation (G – generated at the beginning, U – provided by the user, E*x* – added automatically during the *x*th exploration). The value in the parentheses of IC denotes the number of the exploration of the structure (starting with 1), in which the invocation consequence was added. The overall statistics of the filled tree data structure are summarized in Table I.

TABLE I. OVERALL STATISTICS OF THE FILLED TREE DATA STRUCTURE

Feature	Count
Invocations generated at beginning (G)	39
Invocations provided by user (U)	16
Automatically added invocations in 1st exploration (E1)	4
Automatically added invocations in 2nd exploration (E2)	4
Automatically added invocations in 3rd exploration (E3)	0
Generated consequences in 1st exploration (1)	68
Generated consequences in 2nd exploration (2)	21
Generated consequences in 3rd exploration (3)	0
Explorations	3

```
C: ControlPanel
 S: boolean isTrafficLightsActivated()
  I(G): isTrafficLightsActivated()
   IC(1): return value [true]
 S: void setTrafficLightsActivated(boolean)
  I(G): setTrafficLightsActivated(true)
   IC(1): subsequent invocation
    [TrafficControlAlgorithm.isActivated()]
   IC(2): subsequent invocation
    [TrafficControlAlgorithm.setActivated(true)]
  I(G): setTrafficLightsActivated(false)
   IC(1): subsequent invocation
    [TrafficControlAlgorithm.isActivated())
   IC(1): subsequent invocation
    [TrafficControlAlgorithm.setActivated(false)]
                                       ...
C: SensorAccess
 S: int getQueueLength(String)
  I(G): getQueueLength(null)
   IC(1): exception [NullPointerException]
  I(U): getQueueLength("E_01")
   IC(1): subsequent invocation
    [OpticDetection.getVehiclesCount("E_01")]
   IC(1): return value [2]
   IC(2): subsequent invocation
    [InductionLoop.isVehicle("E_01")]
   IC(2): return value [1]
                                       ...
 S: boolean isVehicle(String)
  I(G): isVehicle(null)
   IC(1): exception [NullPointerException]
  I(U): isVehicle("E_01")
   IC(1): subsequent invocation
    [OpticDetection.getVehiclesCount("E_01")]
   IC(1): return value [true]
   IC(2): subsequent invocation
    [InductionLoop.isVehicle("E_01")]
   IC(2): return value [true]
                                       ...
 S: DetectorType getDetectorType()
  I(G): getDetectorType()
   IC(1): return value [DetectorTypes.OPTIC]
 S: void setDetectorType(DetectorTypes)
  I(G): setDetectorType(null)
   IC(1): exception [NullPointerException]
  I(G): setDetectorType(DetectorTypes.OPTIC)
   IC(1): nothing observable
  I(G): setDetectorType(DetectorType.INDUCTION)
   IC(1): nothing observable

C: StatisticsCollector
 S: Statistics getStatistics()
  I(G): getStatistics()
   IC(1): return value [statistics]
                                       ...
C: OpticDetection
 S: int getVehiclesCount(String)
  I(G): getCurrentVehiclesCount(null)
   IC(1): exception [NullPointerException]
  I(E1): getCurrentVehiclesCount("E_01")
   IC(1): return value [2]
                                       ...
C: InductionLoop
 S: boolean isVehicle(String)
  I(G): isVehicle(null)
   IC(1): exception [NullPointerException]
  I(E2): isVehicle("E_01")
   IC(2): return value [true]
                                       ...
```

Figure 5. Selected parts of the explored and filled tree data structure

As seen in Table I, the tree data structure is explored three times in this case. During the third exploration, no new invocation consequences and no new invocations are generated, which means that the structure does not change and the algorithm ends (see Section V.B).

There are four newly generated invocations of the service `getVehiclesCount()` of the `OpticDetection` component added during the first exploration and four newly generated invocations of the service `isVehicle()` of the `InductionLoop` component during the second exploration. These invocations are direct consequences of the invocations of the `getQueueLength()` service of the `SensorAccess` component – not its `isVehicle()` service, which would have the same consequences, but is invoked after the `getQueueLength()` service, so the invocations are already present. In other words, the invocations provided by the user for the `SensorAccess` component propagate automatically to other components (`OpticDetection` and `InductionLoop` in this case) and enable their better testing. This is a very useful feature of our approach.

It should also be noted that both sets of the subsequent invocations described in the previous paragraph are generated by the same invocation, but in different explorations. The reason is that the inner state of the `SensorAccess` component is being changed during the exploration of the tree data structure by its `setDetectorType()` service. So, although the inner state of the components is unobservable, it can (and often does) influence the behavior of their services. Due to this, the services, which lack any observable consequence, but presumably change the inner state of the components, can influence the generation of the invocation consequences. So, the repetitive explorations of the tree data structure maximize the number of generated invocation consequences.

As it was mentioned in Section V.B, the filled explored tree structure can be directly transcribed to the resulting scenario, which is a XML file. The part of this scenario for the structure depicted in Figure 5 is depicted in Figure 6.

```
<scenario>
  <invocation time="1" componentName="ControlPanel"
      serviceName="isTrafficLightsActivated">
    <parameters>
    </parameters>
    <consequences>
      <consequence type="RETURN_VALUE"
          dataType="boolean" value="true" />
    </consequences>
  </invocation>
  <invocation time="2" componentName="ControlPanel"
      serviceName="setTrafficLightsActivated">
    <parameters>
      <parameter dataType="boolean" name="arg0"
          value="true">
    </parameters>
    <consequences>
      <consequence type="SUBSEQUENT_INVOCATION"
          componentName="TrafficControlAlgorithm"
          serviceName="isActivated">
        <parameters>
        </parameters>
      </consequence>
      <consequence type="SUBSEQUENT_INVOCATION"
          componentName="TrafficControlAlgorithm"
          serviceName="setActivated">
        <parameters>
          <parameter dataType="boolean" name="arg0"
              value="true">
        </parameters>
      </consequence>
    </consequences>
  </invocation>
                             ...
</scenarios>
```

Figure 6. A part of the resulting scenario

VII. CONCLUSION

In this paper, we described an approach to the semi-automated generation of the scenarios for the simulation testing of software components. The approach is based on the analysis of the interfaces of the particular software components and observation of their mutual interactions. The approach is intended for situations when the source code of the components is unknown, but their implementations are available. Then, it enables to partially analyze the behavior of the particular components. The resulting scenarios are useful for the testing of the functionality of new versions of the components, which shall replace their older versions, and as the basis for the testing of the extra-functional properties and quality of services of the components. In the latter case, however, the descriptions and constraints of the extra-functional properties and the quality of services must be filled into the generated scenario by the user.

The functioning of the approach was demonstrated on the Traffic crossroad control case study. The approach was designed for the SimCo simulation tool and OSGi component model, but its basic ideas are utilizable for other existing component models as well.

In our future work, we will focus on the better exploration of the structure, from which the scenarios are generated. This includes ordering of the particular service invocations and attempting to extrapolate the inner states of the components.

ACKNOWLEDGMENT

This work was supported by the European Regional Development Fund (ERDF), project "NTIS – New Technologies for the Information Society", European Centre of Excellence, CZ.1.05/1.1.00/02.0090.

REFERENCES

[1] C. Szyperski, D. Gruntz, and S. Murer, Component Software – Beyond Object-Oriented Programming, ACM Press, New York, 2000.

[2] S. Becker, H. Koziolek, and R. Reussner, "The Palladio component model for model-driven performance prediction," Journal of Systems and Software, vol. 82(1), 2009, pp. 3–22.

[3] P. C. Heam, O. Kouchnarenko, and J. Voinot, "Component Simulation-based Substitutivity Managing QoS Aspects," Electronic Notes in Theoretical Computer Science, vol. 260, 2010, pp. 109–123.

[4] T. Potuzak, R. Lipka, J. Snajberk, P. Brada, and P. Herout, "Design of a Component-based Simulation Framework for Component Testing using SpringDM," ECBS-EERC 2011 – 2011 Second Eastern European Regional Conference on the Engineering on Computer Based Systems, Bratislava, September 2011, pp. 167–168.

[5] T. Potuzak, R. Lipka, P. Brada, and P. Herout, "Testing a Component-based Application for Road Traffic Crossroad Control using the SimCo Simulation Framework," 38th Euromicro Conference on Software Engineering and Advanced Applications, Cesme, Izmir, September 2012, pp. 175–182.

[6] D. Xu, H. Li, and C. P. Lam, "Using Adaptive Agents to Automatically Generate Test Scenarios from the UML Activity Diagrams," Proccedings of the 12th Asia-Pacific Software Engineering Conference, December 2005.

[7] T. Potuzak and R. Lipka, "Possibilities of Semi-automated Generation of Scenarios for Simulation Testing of Software Components," International Journal of Information and Computer Science, vol. 2(6), September 2013, pp. 95–105.

[8] The OSGi Alliance, OSGi Service Platform Core Specification, release 4, version 4.2, 2009.

[9] R. S. Hall, K. Pauls, S. McCulloch, and D. Savage, OSGi in Action: Creating Modular Applications in Java, Manning Publications Co., Stamford, 2011.

[10] J. McAffer, P. VanderLei, and S. Archer, OSGi and Equinox: Creating Highly Modular JavaTM Systems, Pearson Education Inc., Boston, 2010.

[11] D. Rubio, Pro Spring Dynamic Modules for OSGiTM Service Platform, Apress, USA, 2009.

[12] R. M. Fujimoto, Parallel and Distributed Simulation Systems, John Wiley & Sons, New York, 2000.

[13] P. G. Sapna and H. Mohanty, "Automated Scenario Generation based on UML Activity Diagrams," International Conference on Information Technology, 2008, December 2008, pp. 209–214.

[14] S. J. Cunning and J. W. Rozenbiit, "Test Scenario Generation from a Structured Requirements Specification," IEEE Conference and Workshop on Engineering of Computer-Based Systems, 1999, Proceedings, March 1999, pp. 166–172.

[15] R. Lipka, T. Potuzak, P. Brada, and P. Herout, "Verification of SimCo – Simulation Tool for Testing of Component-based Application," EUROCON 2013, Zagreb, July 2013, pp. 467–474.

[16] J. Bauer and A. Finger, "Test Plan Generation Using Formal Grammars," Proceedings of the Fourth International conference on Software Engineering, Los Alamitos, September 1979, pp. 425–432.

[17] X. Hou, Y. Wang, H. Zheng, and G. Tang, "Integration Testing System Scenarios Generation Based on UML," 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering, August 2010, pp. 271–273.

[18] M. Shirole and R. Kumar, "UML Behavioral Model Based Test Case Generation: A Survey," ACM SIGSOFT Software Engineering Notes, Vol. 28(4), 2013, pp. 1–12.

[19] Y. Yongfeng, L. Bin, L. Minyan, and L. Zhen, "Test Cases Generation for Embedded Real-time Software Based on Extended UML," 2009 International Conference on Information Technology and Computer Science, Kiev, 2009, pp. 69–74.

[20] S. Liu and W. Shen, "A Formal Approach to Testing Programs in Practice," 2012 International Conference on Systems and Informatics, Yantai, 2012, pp. 2509–2515.

[21] F. Naseer, S. U. Rehman, and K. Hussain, "Using Meta-data Technique for Component Based Black Box Testing," 2010 6th International Conference on Emerging Technologies, Islamabad, 2010, pp. 276–281.

[22] V. A. De Santiago Jr. and N. L. Vijaykumar, "Generating model-based test cases from natural language requirements for space application software," Software Quality Journal, vol. 20(1), 2012, pp. 77–143.

[23] S. S. Somé and X. Cheng, "An Approach for Supporting System-level Test Scenarios Generation from Textual Use Cases," Proceedings of the 2008 ACM symposium on Applied computing, Fortaleza, 2008, pp. 724–729.

[24] V. Simko, D. Hauzar, T. Bures, P. Hnetynka, and F. Plasil, "Verifying Temporal Properties of Use-Cases in Natural Language," Springer-Verlag, vol. 7741, 2013, pp. 350–367.

[25] A. Cockburn, Writing Effective Use Cases. Addison-Wesley, Boston, 2000

A Scalable Framework for Advanced Driver Assistance Systems Simulation

Kareem Abdelgawad, Mohamed Abdelkarim, Bassem Hassan, Michael Grafe, and Iris Gräßler
Heinz Nixdorf Institute
University of Paderborn, Germany
{Kareem.Abdelgawad, Mohamed.Abdelkarim, Bassem.Hassan, Michael.Grafe, Iris.Graessler}@hni.uni-paderborn.de

***Abstract*—Advanced Driver Assistance Systems (ADAS) are mechatronic vehicle systems that contribute to improving road safety and increasing driving comfort. Apart from all technical challenges regarding control algorithms and sensor quality, customer acceptance of ADAS is an important concern to automobile manufacturers. Demonstrating new ADAS to customers in real traffic environments is impractical and leads to significant efforts and costs. This paper presents the structure of a scalable framework used to implement ADAS virtual prototypes. The design approach ensures maximum flexibility and scalability for integrating new ADAS functions. The framework is composed of modular functional units that enclose real-time capable simulation models developed with MATLAB/Simulink. The design of the functional units and the input-output relationships of their models are presented. Prototypical implementation of two innovative ADAS is presented to show the usability and validity of the framework for ADAS demonstration and training purposes. The developed framework was integrated in an existing PC-based driving simulator used to interactively demonstrate ADAS by means of a simulated traffic environment.**

Keywords-Driving simulators; Advanced Driver Assistance Systems (ADAS); Virtual prototyping; MATLAB/Simulink

I. INTRODUCTION

Improving road safety standards is one of the main concerns in the automotive industry. Advanced Driver Assistance Systems (ADAS) are complex mechatronic vehicle systems that monitor vehicle surroundings, as well as driving behavior. They provide drivers with essential information and take over difficult or repetitive tasks. In critical driving situations, these systems warn and may intervene actively to support the drivers, and hence, lead to increased road safety. ADAS belong to the active safety systems, which help to prevent accidents or at least minimize possible consequences [1].

By utilizing diverse sensor technologies (camera, radar, ultrasonic, etc.) and decision algorithms, different levels of assistance are achieved [2]. On the one hand, some ADAS, like, e.g., Lane Departure Warning [3], only alert the driver to critical situations by means of optical, acoustic and/or haptic feedback. On the other hand, other ADAS do not only recognize driving situations and warn the driver, but also intervene actively in order to prevent possible collisions. A common example of the latter type is Emergency Brake Assist [4], which applies full braking if driver fails to respond to obstacles in front of the vehicle.

Automobile manufacturers and suppliers are confronted with considerable technical challenges while developing ADAS. However, there are additional challenging aspects related to ADAS deployment and public acceptance. Firstly, a flexible test environment is required in order to validate ADAS concepts and assess their decision logic. Secondly, a clear concept for driver-vehicle interface has to be addressed in early development phases; this ensures that drivers can handle the systems appropriately. On the other hand, demonstrating safety and comfort benefits of ADAS to consumers is a key factor for smooth market penetration and development.

However, validating and demonstrating ADAS in real traffic environments are impractical and lead to significant efforts and costs. Moreover, real traffic environments are principally random and do not allow for standardized driving tests or reproducible research results. Driving simulators offer a potent virtual prototyping platform to test and verify ADAS in different development phases [5]. For demonstration and training purposes, they can be utilized to make drivers familiar with new ADAS, and hence, accelerate the learning phase.

The project TRAFFIS (German acronym for Test and Training Environment for ADAS) is carried out at the University of Paderborn with the target of supporting industrial development, testing and training of modern ADAS using a reconfigurable driving simulator [6]. It is funded by the European Union “ERDF: European Regional Development Fund” and the Ministry of Economy, Energy, Industry, Trade and Craft of North Rhine Westphalia in Germany.

One objective of the project TRAFFIS is the development of a platform for ADAS demonstration and training purposes. As shown in Figure 1, three driving simulator variants with different complexity levels and simulation fidelity have been built: TRAFFIS-Light, TRAFFIS-Portable and TRAFFIS-Full. The TRAFFIS-Full variant incorporates a complex motion platform and a surround projection system. The motion platform can fully simulate vehicle lateral and longitudinal accelerations. Moreover, real vehicle cabins can be used, so that drivers experience realistic control cues. The TRAFFIS-Portable variant has a pneumatic motion platform and a four-wall projection system. The TRAFFIS-Light variant is simple a PC-based driving simulator with no motion platform. These driving simulator variants along with an innovative configurability concept offer a flexible test and training environment for various in-vehicle systems [6]. However, the focus is given mainly to the development of ADAS.

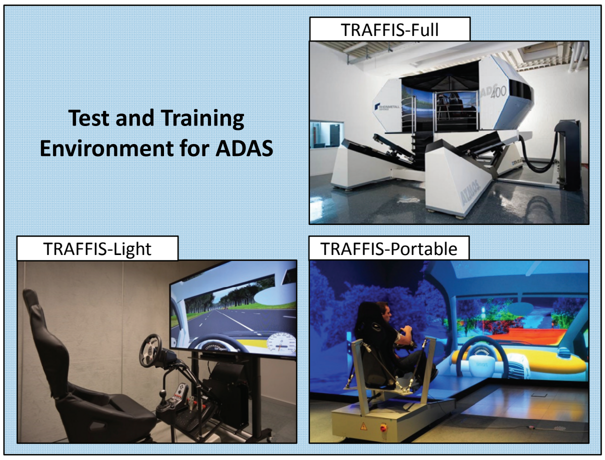


Figure 1. Driving simulators with different complexity levels.

A simulation framework with flexible prototyping concepts is required for easy and convenient ADAS demonstration and training. This paper presents the structure of a virtual prototyping framework for implementing ADAS and demonstrating their benefits with driving simulators. The developed framework consists of several functional units enclosing simulation models that were implemented with MATLAB/Simulink. The models are arranged in a modular architecture and developed, so that they communicate in a loosely coupled fashion. Adaptation of models interfaces can be performed with minimum effort. The design approach ensures maximum flexibility and scalability for implementing any ADAS virtual prototypes. The design of the functional units is discussed along with input-output relationships of the underlying models. All models are real-time capable, i.e., the simulation runs in real time using the Real-Time Windows Target library from Mathworks.

As a first step, the developed ADAS simulation framework was integrated with an existing simulation environment of the PC-based driving simulator, i.e., TRAFFIS-Light, which represents the simplest driving simulator variant within the project TRAFFIS. Furthermore, virtual prototypes of two innovative ADAS are presented to show and validate the capability of the developed framework for ADAS demonstration and training purposes.

This paper is structured as follows: Section 2 shows briefly the driving simulation environment with which the developed framework was integrated. Section 3 presents the design approach of the developed ADAS virtual prototyping framework along with the concepts of its functional units and models. Section 4 demonstrates two ADAS prototypes realised with the developed framework and demonstrated using the PC-based driving simulator. Finally, Section 5 derives the conclusion and summarizes the benefits of the presented approach.

II. Driving Simulation Environment

The simulation environment of the PC-based driving simulator consists of two main functional units: a vehicle dynamics model and a traffic model. Figure 2 illustrates its structure and the direction of information flow.

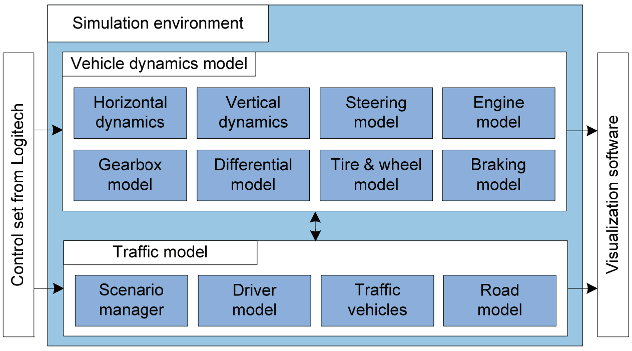


Figure 2. Simulation environment of the PC-based simulator.

Each functional unit consists of real-time capable sub-models implemented with MATLAB/Simulink. The visualization software represents the main feedback cue of the PC-based driving simulator. 3D models for the main vehicle, road and traffic participants are controlled through the corresponding sub-models of the driving simulation environment. The visualization software was implemented with Unity [7]; a development engine that provides rich and easy functionalities for creating interactive 3D tools. The PC-based driving simulator incorporates a racing wheel-transmission-pedals set from Logitech and a racing seat from Speedmaster. It is still fully interactive with respect to steering, gears, acceleration and brake controls. This simulator variant and its simulation environment are considered in this work. The next sub-sections discuss each functional unit briefly.

A. *Vehicle dynamics model*

Modeling realistic vehicle dynamics is essential for the development of different in-vehicle systems. The design of ADAS controllers relies primarily on the underlying vehicle dynamics. The utilized vehicle dynamics model produces the actual physical characteristics of the main vehicle and allows for a total of 16 Degrees Of Freedom (DOF) [8]. The chassis has three translational and three rotational motions [9]. Each wheel has a relative vertical translational motion and a rotational motion around the wheel axis. In addition, each of the front wheels has a relative rotational motion around the vertical direction of the road. The vehicle dynamics model receives control signals from the hardware control set and calculates the resultant motions; these are exported mainly to the visualization software to update vehicle position and orientation on the screen. The traffic model provides information about the road, i.e., height and friction under each of the vehicle tires; these in turn are used by the vehicle dynamics model to update the calculations of the vehicle position, orientation and speed. The vehicle dynamics model is composed of various sub-models [9]. It implements the blocks shown in Figure 2 as modular Simulink subsystems.

B. *Traffic model*

The traffic model is used to simulate the surrounding vehicles and the road [10]. It simulates realistic behavior of the traffic vehicles and their interactions, which is necessary to give realistic feedback cue to the driver on the one hand, and to efficiently test ADAS functions on the other hand. The traffic model consists mainly of four sub-models: road model, traffic vehicles models, driver model and a scenario manager model, which composes arbitrarily different traffic situations. The traffic model receives current position, orientation and speed of the main vehicle from the vehicle dynamics model; these are used mainly by the driver model to arrange for appropriate traffic flow without collisions with the main vehicle.

III. ADAS Simulation Framework

The vehicle dynamics model and traffic model constitute the central functional units of a simulation environment for a simple driving simulator. However, a comprehensive simulation framework is still required to conveniently simulate different ADAS functionalities. Active safety in general and ADAS in particular exhibit continuous development. New ADAS functions are developed to achieve safer traffic flow and more comfortable driving. Moreover, the availability of a wide range of sensors and the possibility to integrate different sources of information allow the development of more new reliable ADAS. Hence, one principal requirement for building a flexible ADAS test and training environment is to maintain maximum modularity and scalability. The developed ADAS virtual prototyping framework is structured in a modular form that ensures its scalability. That is, new ADAS prototypes can be added almost without significant input-output interface adjustments. Furthermore, different ADAS can be integrated together to implement more advanced capabilities such as autonomous driving.

Driving is a multitasking activity where drivers have to manage their attention between various actions and reactions within a dynamic traffic environment [11]. The design approach of the developed ADAS simulation framework is based on an analogy between human driving behavior and the functionality of ADAS. Figure 3 shows the structure of the ADAS simulation framework. It consists of four functional units or stages: user interface stage, recognition stage, guidance stage and control stage. The latter three functional units resemble the activity model the human driver mainly follows while driving a vehicle. The recognition stage represents the senses of human drivers for recognizing road path and other traffic participants, i.e., current traffic situation. The guidance stage corresponds to the reasoning capabilities of the human driver and compromises made according to the recognized traffic situation, i.e., decisions to accelerate, brake, steer or to make a certain maneuver. The control stage simulates the actual physical actions the human driver performs to carry out appropriate decisions. Related approaches for human driving models are presented in [12] and [13]. The analogical comparison with human drivers is valid under the assumption that any ADAS can be represented as an assisting automatic driver that warns the driver and/or takes over the driving tasks in critical traffic situations.

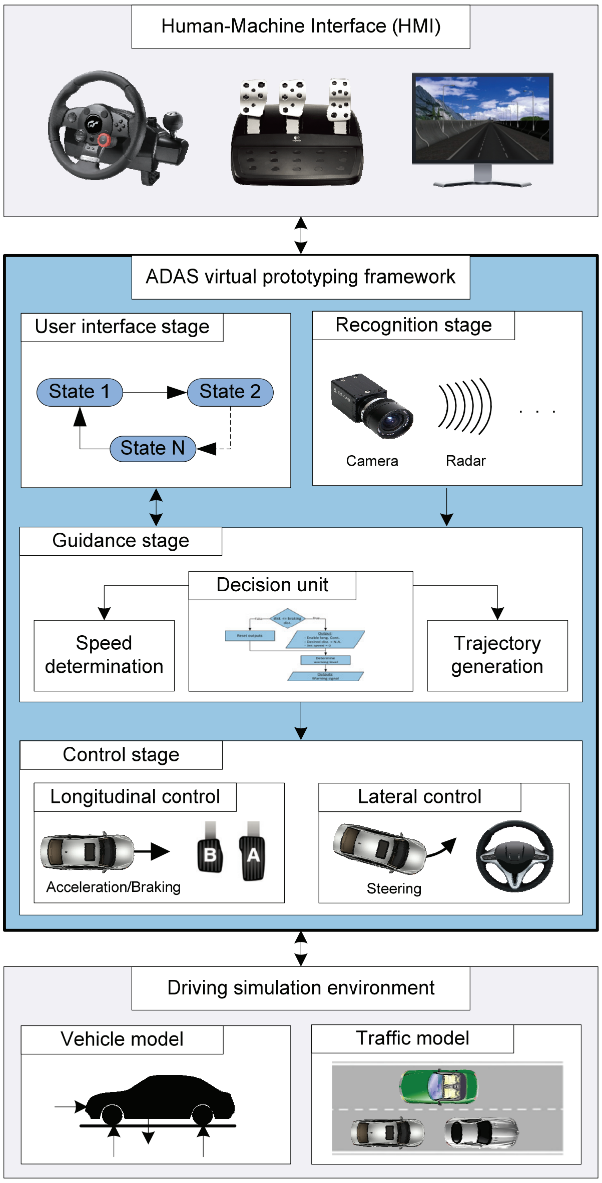


Figure 3. ADAS framework and its relation with the driving simulation environment and HMI.

As shown in Figure 3, the ADAS virtual prototyping framework is connected to the other functional units of the driving simulation environment and the hardware controller set along with the visualization software (HMI) of the PC-based driving simulator. The ADAS simulation framework receives inputs from the HMI to set the ADAS states, i.e., activate, deactivate or alter some parameters. It eventually applies force feedback on the steering wheel according to the driving situation and the type of the activated ADAS. The ADAS simulation framework gets the states of the main vehicle, i.e., position, orientation and speed, which are calculated by the vehicle dynamics model. In case of ADAS with active intervention, it overrides the requests of the human driver and controls the states of the vehicle. The ADAS simulation framework notifies the traffic model regarding the activated ADAS, the traffic model invokes in turn predefined traffic scenarios and provides information about the traffic participants. The following sub-sections discuss the design of each functional unit of the ADAS simulation framework and the fundamental input-output signals.

A. User interface stage

The user interface stage accounts for the interaction between user, i.e., simulator driver, and the ADAS simulation framework. It implements the logic required for transitioning between different ADAS functional states, like, e.g., on, off, standby, etc. Each ADAS user interface is modeled separately as a Stateflow sub-model (a control logic tool used to model event-driven systems within Simulink). Figure 4 shows the structure of the user interface stage and the main input-output signals.

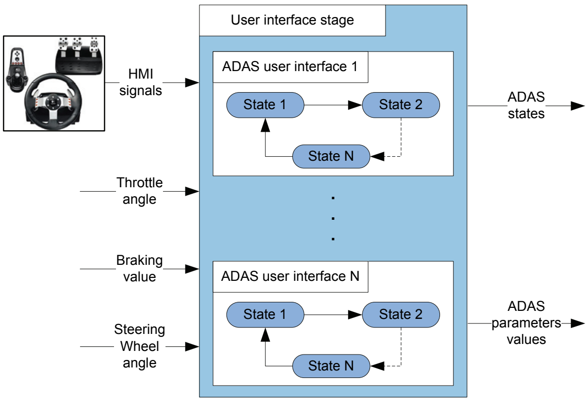


Figure 4. ADAS user interface stage.

Each sub-model receives an enable/disable signal from the buttons set, as well as the values of the acceleration and brake pedals, gear selector and steering wheel of the Logitech controller. Furthermore, it gets feedback signals indicating the desired maneuvers of ADAS controllers, namely, throttle angle, braking value and steering wheel angle. These are compared with corresponding signals indicating the intention of the driver, which is provided through the HMI. If there is a difference, and taking ADAS type into account, the corresponding sub-model decides if ADAS should make a transition from one functional state to another. For instance, while an autonomous driving function will be deactivated if the driver moves the steering wheel slightly; an emergency braking function will not be deactivated for such an action. As outputs, indications for ADAS functional states along with the desired ADAS parameter values are exported to the corresponding ADAS sub-routines within the guidance stage, discussed in a later section. This arrangement for the user interface stage conforms to the modularity and scalability requirement of the ADAS simulation framework. For modeling new ADAS, corresponding Stateflow sub-models have to be implemented separately within the user interface stage using the same set of input-output interfaces.

B. Recognition stage

Driver assistance systems require surrounding recognition capabilities to be able to perceive the traffic environment. Any ADAS must incorporate one or more sensors, like, e.g., GPS, cameras, radar, ultrasonic, laser, lidar. Many variants already exist in market; moreover, a lot of new sensor technologies and concepts are being developed, like, e.g., sensor fusion [14]. Hence, there are a lot of sensor models to be integrated in order to achieve a comprehensive ADAS virtual prototyping framework. The recognition stage is composed mainly of two units: a detection unit containing different sensor models and a relevance filter unit. Figure 5 shows the structure of the recognition stage and the essential input-output signals.

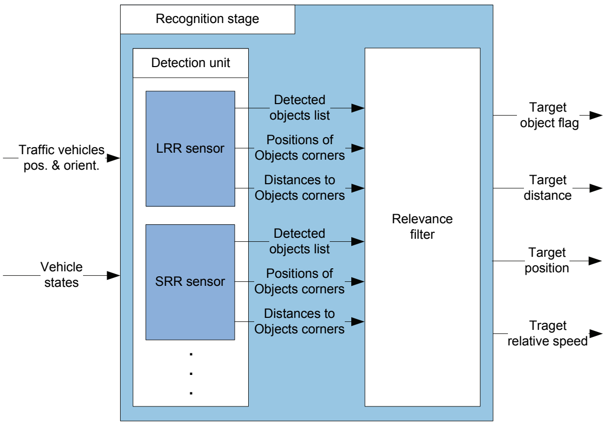


Figure 5. ADAS recognition stage.

Information about road and traffic participants is provided through the traffic model. Vehicle position, orientation and speed, i.e., vehicle states, are provided by the vehicle dynamics model. The detection unit is designed in the form of a bowl that contains different sensor models, like, e.g., radar sensor model, ultrasound sensor model, etc. The main output from a sensor model is a list of objects characterized with detection flags, i.e., detected objects list. In addition, each sensor model provides the positions and distances of detected object corners. Short-Range Radar (SRR) and Long-Range Radar (LRR) sensor models have been implemented within the detection unit. Both models are based on the mathematical description or geometry of detection area [15]. The long-range radar model is ideally suited for detection distance longer than 30 meters; it can typically detect objects 250 meters away. On the other hand, the short-range radar model provides wider view and detection distance below 30 meters. All parameter values can be modified to alter the geometrical description of detection area if necessary, i.e., the geometrical coverage and detection range are adjustable, so that sensor characteristics can be changed arbitrarily.

Within the relevance filter unit, detected objects are further filtered according to the position and orientation of the main vehicle relative to the road. That is, the outputs of all sensor models are forwarded to a relevance filter, which generates a flag indicating the most relevant object to the main vehicle, i.e., target object. Moreover, relative speed of the target object and distance and position of its nearest corner are calculated. Figure 6 illustrates the selection functionality of the relevance filter unit.

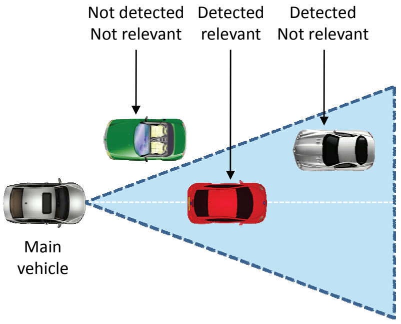


Figure 6. Target object selection of the relevance filter unit.

The detection unit is extensible for additional sensor models to be developed, whereas the functionality of the relevance filter unit has not to be altered. However, the relevance filter unit considers only sensors of the same direction of detection and determines only one target object. If other sensor models for other directions of detection are to be implemented, like, e.g., right and left sides of the vehicle, corresponding relevance filter units have to be designed conforming to the structure of the recognition stage and the same set of input-output signals.

C. *Guidance stage*

As mentioned previously while making analogy between the developed ADAS simulation framework and the human driving model, the guidance stage represents the understanding of recognized traffic situations and the decisions required for safe or comfortable driving. The guidance stage derives its central role from being in the middle of a detection phase, i.e., recognition stage, and an action phase, i.e., control stage. On the one hand, it interprets the information provided by the recognition stage, i.e., it evaluates the perceived traffic situations. On the other hand, it determines the actions required to avert undesirable traffic situations. Figure 7 shows the structure of the guidance stage and the main input-output signals.

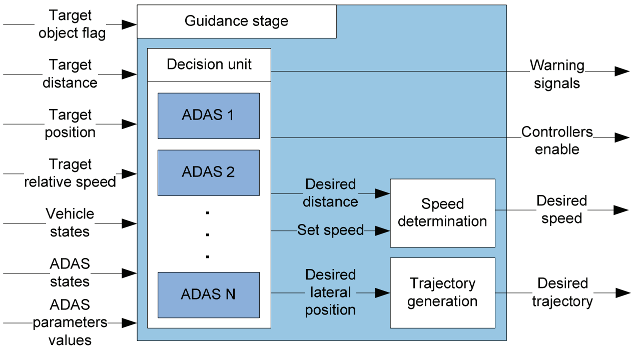


Figure 7. ADAS guidance stage.

The guidance stage is consisted of three sub-functions: Decision unit, speed determination and trajectory generation. These sub-functions are discussed next.

- Decision unit

The logic of each ADAS is implemented within the decision unit as a separate sub-routine. The decision unit receives indication for the presence of a target object along with its relative speed, distance to and position of its nearest corner from the recognition stage. Figure 8 shows a flow chart for the main function of the decision unit.

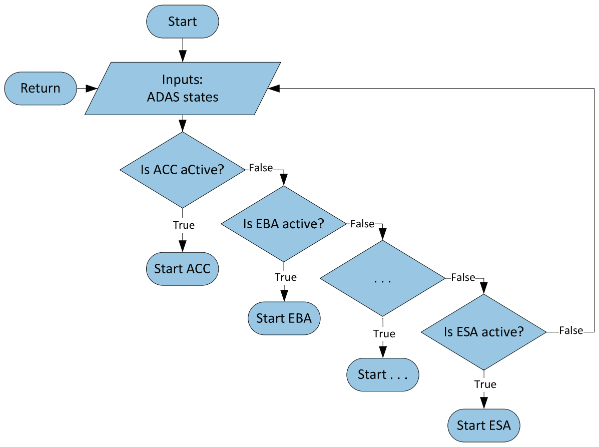


Figure 8. Transition logic between ADAS sub-routines.

The user interface stage implies which ADAS is to be activated with which parameter values. The main function of the decision unit loops through all the implemented ADAS sub-routines. Only that of the chosen ADAS is executed while other ADAS sub-routines are ignored. It considers the traffic situation detected by the recognition stage, ADAS states and parameter values exported by user interface stage and vehicle states provided by vehicle dynamics model. Accordingly, it determines desired distance to a target object, set speed or desired lateral position required to alter the path of the main vehicle. In addition, it sends enable signals to corresponding vehicle controllers, i.e., longitudinal and/or lateral controller, discussed in a later section. The activated ADAS generates warning signals required to trigger some display elements within the visualization software.

Similar to the user interface stage and recognition stage, the decision unit is extensible, so that any logic for new ADAS prototypes can be simply added as new separate sub-routines. The set of input-output signals is comprehensive and suitable for almost all active and passive ADAS.

- Speed determination

This function maintains constant time headway space to a target object that eventually drives with lower speed than that of the main vehicle [16]. Principally, the headway distance varies with main vehicle speed; this allows for a fixed margin in time for the ADAS to react to changes in the speed of the target object. The speed determination function is basically a distance controller that determines the speed required to maintain the desired headway space, taking the speed of the target object into account. It is based on the so-called slide mode control [17]. It is a simple control method that proves good stability especially where the control actions are discontinuous functions of system states and inputs.

The speed determination function handles the orders of the decision unit with respect to the longitudinal direction. While the desired headway space is provided by the decision unit, i.e., the sub-routine of an activated ADAS, a speed command is generated to obtain this distance accordingly. Figure 9 shows the difference between the desired and actual headway distances.

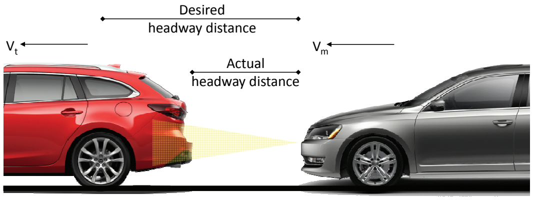


Figure 9. Headway distance control and speed determination.

Moreover, the function selects the minimum of the ADAS set speed, like, e.g., set speed of an adaptive cruise control, and that required for following a target object while preserving constant headway space. Finally, the desired speed is forwarded to the longitudinal controller discussed in a later section.

- Trajectory generation

This function generates the trajectory required to guide the vehicle through the road or to move it from one lateral position to another. The function encloses the mathematical description of the road, so that the generated trajectory reconsiles with road path. The trajectory is generated in the form of a moving point in front of the vehicle. The activated ADAS within the decision unit determines the desired lateral position required to adjust the vehicle path or to avoid a collision for example. The function limits the rate of lateral position change generated within the decision unit in order to obtain reasonable and realistic lateral transitions. Although it handles the orders of the decision unit mainly with respect to the lateral direction, the function adds a predetermined offset to the longitudinal component of current vehicle position. Hence, the location of the moving point is updated continuously and gradually to form the desired trajectory, as shown in Figure 10.

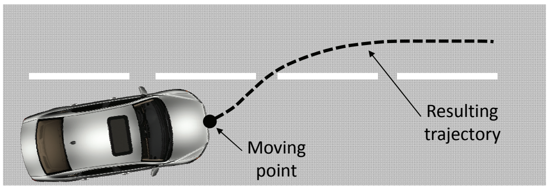


Figure 10. Moving point for trajectory generation.

The desired trajectory represented as postion updates is forwarded to the lateral controller discussed in a later section.

D. *Control stage*

A motion controller is required in order to control the state of the vehicle in case of active ADAS intervention. As shown in Figure 11, decoupled longitudinal and lateral controllers were implemented to execute the orders of the guidance stage and guide the vehicle accordingly.

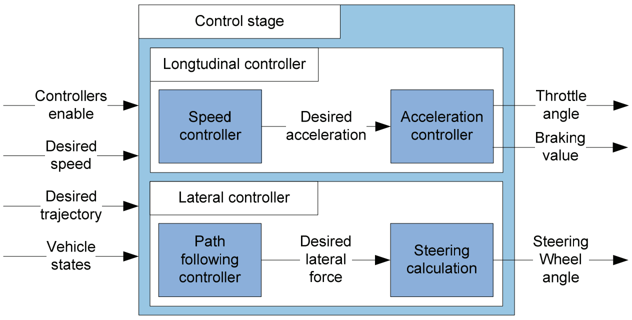


Figure 11. ADAS control stage.

The control stage gets an enable signal from the guidance stage that indicates which controller is to be activated, and hence, moving the vehicle with a desired speed in a desired direction. These controllers are discussed next.

- Longitudinal controller

The longitudinal controller is a cascaded speed-acceleration control loop system [18]. It is composed of two successive controllers: speed controller and acceleration controller. The speed controller is a Proportional-Integral (PI) type that constitutes the outer loop of the longitudinal controller. The speed command from the guidance stage is compared with the actual speed of the vehicle to generate a speed error. The speed controller generates an acceleration value required to overcome the speed error. It is followed by an anti-windup function to prevent output saturation [19]. The desired acceleration is forwarded then to the acceleration controller.

The acceleration controller constitutes the inner loop of the longitudinal controller. The desired acceleration is compared with the actual acceleration of the vehicle to generate an acceleration error. The acceleration controller implements the inverse form of vehicle dynamics and drivetrain of the vehicle model [20]. The acceleration controller is composed mainly of three sub-models, as shown in Figure 12.

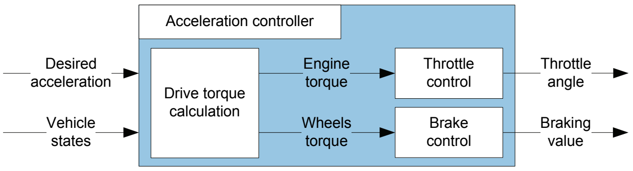


Figure 12. Sub-models of the acceleration controller.

The drive torque calculation sub-model generates the wheels torque and engine torque required to achieve the desired acceleration. It is based on the dynamics equations of the vehicle model. The throttle control sub-model generates the throttle angle according to the required engine torque. It is based on the engine model within the vehicle dynamics model. Similarly, the brake control sub-model generates the braking value according to the required wheels torque [21]. It is based on the braking model within the vehicle model. The longitudinal controller exports the throttle angle or braking value to the vehicle dynamics model. For comfort driving and realistic vehicle behavior, the throttle and brake control sub-models do not allow the acceleration and deceleration to exceed predetermined limits.

- Lateral controller

The lateral controller handles the path following control problem, i.e., how to control the vehicle, so that it can faithfully follow a prescribed path. As shown in Figure 11, it is composed mainly of two sub-models. The path following controller sub-model gets the trajectory generated by the guidance stage in the form of a moving point, i.e., a point directly in front of the vehicle that updates its location on a certain path. It calculates the front axle force required to let the vehicle adjust its orientation, and hence, follow the moving point to pursue the desired trajectory. The path following controller is based on the feedback linearization control method [22]. The basic idea is to convert the closed-loop control system including the plant, i.e., the horizontal vehicle dynamics model in this case, into linear system dynamics. The method was applied to the bicycle vehicle model [8] and showed optimal robustness even at stability borders, such as rapid steering maneuvers or driving at relatively high speeds in sharp curves. According to the horizontal vehicle dynamics, the steering calculation sub-model determines the steering angle, which corresponds to the desired lateral force. Moreover, it calculates the steering wheel angle using the inversion of the steering model within the vehicle dynamics model. Finally, the lateral controller exports the steering wheel angle required to guide the vehicle in the desired direction to the vehicle dynamics model, and hence, following a certain trajectory. The designed longitudinal and lateral controllers can serve a variety of active ADAS functions where a spontaneous rapid maneuver or the whole driving task is taken over by an automated intervention. The generality and simplicity of the interface between the developed guidance and control stages make it convenient to develop and plug new ADAS functions. The following section presents the logic of two innovative ADAS functions implemented in the decision unit within the guidance stage.

IV. ADAS Prototypical Implementation

To prove the usability of the developed ADAS virtual prototyping framework in general and to show the benefits of its modular structure in particular, prototypes for two new ADAS were implemented: Emergency Brake Assist and Emergency Steer Assist. Both functions represent the state of the art in ADAS development and have different types of intervention. However, both functions have been implemented without any special interface adjustments due to the modularity and scalability of the ADAS simulation framework described in this paper.

A. Emergency Brake Assist

Emergency Brake Assist (EBA) is an ADAS sub-routine implemented within the decision unit of the guidance stage. According to recognized moving or standing objects in front of the vehicle, it initiates automatic braking in the case of a potential rear-end collision provided that the driver has not responded to prior warnings signals [23]. Figure 13 shows a flow chart for a simplified version of the EBA sub-routine.

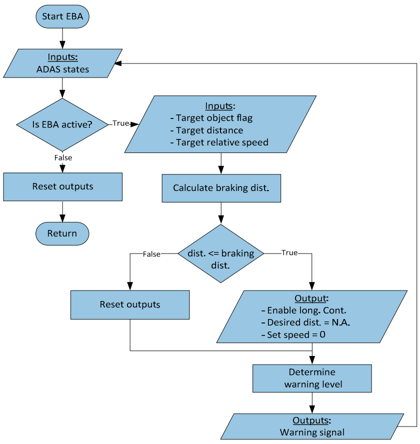


Figure 13. Simplified version of EBA logic within the decision unit.

The intention of the driver is observed through the user interface stage, and hence, is embedded within the ADAS states signal. The EBA sub-routine gets the distance and relative speed of a target object existing in front of the vehicle from the guidance stage. The critical braking distance, i.e., safe distance, is calculated from the provided inputs. This means, the safe distance is variable and depends mainly on the relative speed of the target object. If the actual distance to the target object gets close to the safe distance within predefined limits, the function initiates optical and acoustic warning signals to be handled by the visualization software. The optical warning has three levels: a green cautionary signal if the target object ahead is close, a yellow alert signal if the safe distance is reached and a red critical signal if the actual distance is equal to or fell below the safe distance. In the latter case, if the driver fails to take braking or steering actions, i.e., when an emergency situation is fully confirmed and the state of the target object flag does not change, the EBA sub-routine enables the longitudinal controller and sets the speed to zero. The sub-routine overrides the acceleration request of the driver who is effectively taken out of the loop. However, the driver still can retain control anytime by taking an appropriate steering action, and hence; changing the state of the target object flag. The function was tested and validated with many test scenarios, where different values for the speed of the main vehicle and traffic vehicle ahead were considered. Figure 14 illustrates the switching point between warnings and active intervention distances of the EBA sub-routine

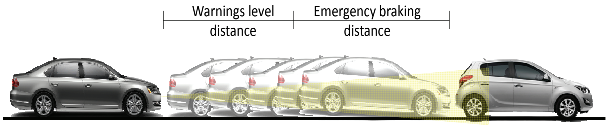


Figure 14. EBA intervention in case of no driver response.

B. *Emergency Steer Assist*

Emergency Steer Assist (ESA) is an ADAS function implemented within the decision unit of the guidance stage. The function supports the driver in the lateral driving task [23]. According to recognized sudden right or left incursion from a traffic object and if the driver has no time left for braking, the function initiates rapid automatic steering intervention in the case of predicted collision, as shown in Figure 15.

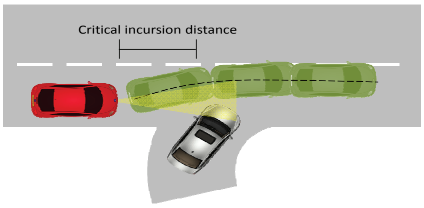


Figure 15. ESA intervention due to sudden road incursion.

Almost similar to Emergency Brake Assist function, the intention of the driver is observed through the user interface stage. If a target object appeared suddenly within the lane of the vehicle, the function decides to steer the vehicle abruptly in the opposite direction. This decision takes the form of a desired (x, y) point, which is exported to the lateral controller. The speed of the vehicle, the distance at which the target object appeared and the intention of the driver are factored in the decision of the function. The critical incursion distance is variable and depends mainly on the speed of the vehicle. The function was tested and validated with test scenarios where different values for the speed of the main vehicle, as well as different distances to the incurring target vehicle were considered.

V. SUMMARY AND CONCLUSION

Advanced Driver Assistance Systems (ADAS) gain importance due to their safety and comfort features. The ADAS virtual prototyping framework described in this paper offers a flexible solution to efficiently validate ADAS concepts and easily demonstrate their benefits to customers. The presented approach is based on an analogy between the functionality of ADAS and the human driving model. This resulted in a comprehensive architecture, which is composed of modular and extensible functional units.

The developed ADAS virtual prototyping framework was integrated with an existing real-time simulation environment of a PC-based driving simulator. To validate the approach and the capabilities of the developed ADAS simulation framework, prototypical implementation of two innovative ADAS functions was presented. Although both functions show different types of intervention, no special signal interface adjustments were necessary. The design of the other functional units of the simulation environment, i.e., vehicle dynamics model and traffic model, has not to be adjusted for any future ADAS prototypes.

A group of test persons were involved in the behavioral validation process of the driving simulator after integrating the ADAS virtual prototyping framework [24]. In other words, an assessment of how drivers react and perform with respect to the implemented ADAS prototypes has been made. The test persons have been subjected to near collision situations, where different values for the speed of the main vehicle and traffic vehicle ahead were considered. The behavioral validation process showed how the test persons could reasonably handle ADAS warnings and active interventions with very good learning curves. Effectiveness, proper operation and drivers' acceptance of the implemented ADAS were evaluated.

The presented approach added new capabilities to the PC-based driving simulator for assessing ADAS algorithms and performing drivers training by means of a driving simulation environment. In general, the modularity and scalability requirement of an ADAS training environment for the project TRAFFIS was fulfilled.

References

[1] T. Hummel, M. Kühn, J. Bende, and A. Lang, "Advanced Driver Assistance Systems – An investigation of their potential safety benefits based on an analysis of insurance claims in Germany," German Insurance Association - Insurers Accident Research, Research Report FS 03, Berlin, 2011

[2] J. Golias, G. Yannis, and C. Antoniou, "Classification of driver assistance systems according to their impact on road safety and traffic efficiency," In Transport Reviews - Journal of Intelligent Transportation Systems, Taylor & Francis Group, vol. 22, 2002, pp. 179-196.

[3] P. Hsiao, K. Hung, S. Huang, W. Kao, and C. Hsu, "An embedded lane departure warning system," IEEE 15th International Symposium on Consumer Electronics (ISCE), Singapore, June 2011, pp. 162-165, ISSN: 0747-668X, ISBN: 978-1-61284-843-3.

[4] R. Zheng, K. Nakano, S. Yamabe, M. Aki, and H. Nakamura, "Study on Emergency-Avoidance Braking for the Automatic Platooning of Trucks," IEEE Transactions on Intelligent Transportation Systems, China, August 2014, Vol. 15, No. 4, pp. 1748-1757, DOI: 10.1109/TITS.2014.2307160, ISSN: 1524-9050.

[5] F. Colditz, L. Dragon, R. Faul, D. Meljnikov, and V. Schill, "Use of Driving Simulators within Car Development," In proceedings of Driving Simulation Conference North America, Iowa City, USA, September 2007.

[6] B. Hassan and J. Gausemeier, "Concept of a Reconfigurable Driving Simulator for Testing and Training of Advanced Driver Assistance Systems," IEEE Transl. ISAM 2013 China, vol. 2, July 2013, pp. 337-339.

[7] A. Gloria, F. Bellotti, R. Berta, and E. Lavagnino, "Serious Games for Education and Training," International Journal of Serious Games, Vol. 1, No. 1, 2014, pp. 100-105, ISSN: 2384-8766.

[8] H. True, "The dynamics of vehicles on road and on tracks," Swets & Zeitlinger B.V., Lisse, Netherlands, vol. 37, April 2003, pp. 96–105.

[9] R. N. Jazar, "Vehicle dynamics: Theory and application," Springer Science+Business Media, LCC, New York, USA, 2008, pp. 37-279, e-ISBN 978-0-387-74244-1.

[10] J. Barcelo, "Fundamental of traffic simulation," Springer Science+Business Media, LCC, New York, USA, 2008, pp. 15-63, ISSN: 0884-8289, e-ISBN: 978-1-4419-6142-6.

[11] C. C. Macadam, "Understanding and modeling the human driver," Journal of Vehicle System Dynamics 49, vol. 40, nos. 1-3, 2003, pp. 101-134.

[12] D. T. Mcruer, R. W. Allen, D. H. Weir, and R. H. Klein, "New results in driver steering control models," Journal of Human Factors and Ergonomics Society, 19(4), SAGE Publications, California, USA, August 1977, pp. 381-397, DOI: 10.1177/001872087701900406.

[13] G. A. Bekey, G. O. Burnham, and J. Seo, "Control Theoretic Models of Human Drivers in Car Following," Journal of Human Factors and Ergonomics Society, 19(4), SAGE Publications, California, USA, August 1977, pp. 399-413, DOI: 10.1177/001872087701900406.

[14] R. Altendorfer, S. Wirkert, and S. Heinrichs-Bartscher, "Sensor Fusion as an Enabling Technology for Safety-critical Driver Assistance Systems," SAE International Journal of Passenger Cars - Electronic and Electrical Systems, October 2010, SAE International, USA, 2010, pp. 183-192, ISSN 0148-7191.

[15] T. Akenine-Möller, "Fast 3D triangle-box overlap testing," Journal of Graphics Tools archive, Vol. 6, No. 2, September 2001, pp. 29-33.

[16] N. Benalie, W. Pananurak, S. Thanok, and M. Parnichkun "Improvement of Adaptive Cruise Control System based on Speed Characteristics and Time Headway," IEEE/RSJ international conference on intelligent robots and systems, Missouri, USA, October 2009, pp. 2403-2408.

[17] J. E. Slotine and W. Li, "Applied nonlinear control," Prentice Hall Englewood Cliffs, New Jersey, USA, ISBN: 0-13-040890-5, 1991, pp. 276–307.

[18] V. V. Sivaji and M. Sailaja, "Adaptive Cruise Control Systems for Vehicle Modeling Using Stop and Go Manoeuvres," International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, vol. 3, Issue 4, July 2013, pp.2453-2456.

[19] C. Poussot-Vassala, O. Senameb, L. Dugardb, and S. M. Savaresic, "Anti-windup Schemes for Proportional Integral and Proportional Resonant Controller," In proceedings of Power Electronics Conference, Roorkee, India, June 2010.

[20] K. Yi, Y. Cho, S. Lee, J. Lee, and N. Ryoo, "A Throttle/Brake Control Law for Vehicle Intelligent Cruise Control," In proceedings of Seoul 2000 FISITA World Automotive Congress, Seoul, Korea, June 2000.

[21] C. Poussot-Vassala, O. Senameb, L. Dugardb, and S. M. Savaresic, "Vehicle Dynamic Stability Improvements Through Gain-Scheduled Steering and Braking Control," Journal of Vehicle System Dynamics 49, vol. 00, no. 00, January 2009, pp. 1597-1621, DOI : 10.1080/00423114.2010.527995.

[22] M. Abdelkarim, T. Butz, and A. Moutchiho, "A nonlinear path following controller for lateral vehicle guidance - Ein nichtlinearer Bahnfolgeregler zur Fahrzeugquerführung," Fahrermodellierung in Wissenschaft und Wirtschaft, Fortschritt-Berichte VDI, vol. 22, no. 35, VDI Verlag, Düsseldorf, Germany, June 2013, pp. 135-145, ISBN: 978-3-18-303522-9.

[23] A. Eckert, B. Hartmann, M. Sevenich, and P. Rieth, "Emergency Steer & Brake Assist – A Systematic Approach for System Integration of two Complementary Driver Assistance Systems," Continental AG. Germany: Paper Nr. 11-0111.

[24] Z. Mao, X. Yan, H. Zhang, and C. Wu, "Driving Simulator Validation for Drivers' Speed Behavior," In proceedings of the Second International Conference on Transportation Engineering, Chengdu, China, July 25-27, 2009, pp. 2887-2892, ISBN: 9780784410394.

Microscopic Simulation of Synchronized Flow in City Traffic: Effect of Driver's Speed Adaptation

Gerhard Hermanns
and Igor N. Kulkov
and Boris S. Kerner
and Michael Schreckenberg

University of Duisburg-Essen
Email: gerhard.hermanns@uni-due.de,
igor.kulkov@uni-due.de,
boris.kerner@uni-due.de,
michael.schreckenberg@uni-due.de

Peter Hemmerle
and Micha Koller
and Hubert Rehborn

Daimler AG
Email: peter.hemmerle@daimler.com,
micha.koller@daimler.com,
hubert.rehborn@daimler.com

***Abstract*—Recently, the synchronized flow traffic pattern has been found in empirical data of oversaturated city traffic. Traffic simulation models based on classical theories cannot reproduce synchronized flow. We present simulation results based on Kerner's three-phase traffic theory. By using a version of the Kerner-Klenov stochastic microscopic traffic flow simulation model, we are able to reproduce all traffic patterns found in city traffic, including synchronized flow. The key role in understanding the emergence of synchronized flow within the model plays the driver's speed adaptation effect.**

Keywords–Microscopic simulation; Three-phase traffic theory; City traffic.

I. Introduction

On an urban road with a traffic signal various congested traffic patterns can be observed. The understanding of these city traffic phenomena is important for the development of effective tools for traffic management and intelligent transportation systems.

There are two main types of city traffic on multilane roads at traffic signals: under- and over-saturated traffic. In accordance with the classical theories [1][2][3][4][5][6], when the flow rate increases above some signal capacity value, a transition from under- to over-saturated traffic occurs. In undersaturated traffic, all vehicles, which are waiting within a queue during the red phase, can pass the signal during the green phase. An opposite case occurs in oversaturated traffic: some of the vehicles in the queue cannot pass the signal location during the green phase resulting in the queue growth (see Figure 1 (a, b)). In accordance with the classical theories [1][2][3][4][5][6], well-developed oversaturated traffic consists of a sequence of moving queues with stopped vehicles separated by regions in which vehicles move from one moving queue to the adjacent downstream moving queue; the mean duration of the vehicle stop within a moving queue does not usually change while the moving queue propagates upstream of the signal (see Figure 1 (c)).

Based on simulations in the framework of the three-phase theory [7][8], Kerner et al. have recently predicted that in addition to classical sequences of moving queues (see Figure 1), in oversaturated traffic synchronized flow pattern(s) (SP) should also occur [9]. In comparison with [9], in this article we make the following new developments: i) we simulate and discuss single vehicle (microscopic) acceleration related to synchronized flow in oversaturated city traffic and ii) we make a briew comparison of simulated synchronized flow with empirical synchronized flow revealed recently in measured field data in oversaturated city traffic. For simulations of spatiotemporal features of SPs in oversaturated traffic we use the Kerner-Klenov stochastic microscopic traffic flow model [10][11][12] in the framework of Kerner's three-phase theory.

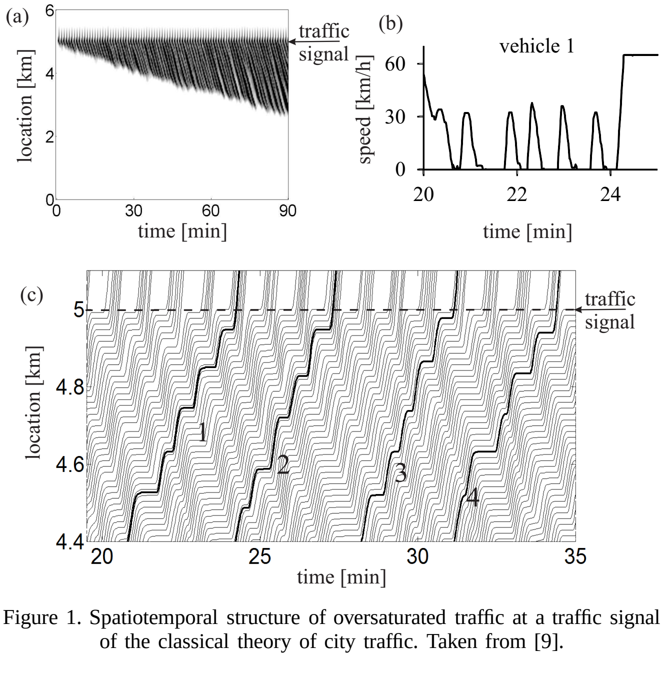


Figure 1. Spatiotemporal structure of oversaturated traffic at a traffic signal of the classical theory of city traffic. Taken from [9].

This article is organized as follows. In Section 1, a short introduction to the theory of synchronized traffic flow in a city is given, and the simulation model is described. In Section

2, empirical findings of synchronized flow in city traffic are presented and compared with simulation results. In Section 3, a conclusion and an outlook to further work are given.

II. Kerner-Klenov simulation model

Urban traffic on a multilane road with a traffic signal at its end can either be oversaturad or undersaturated. In undersaturated city traffic all vehicles waiting in a queue in front of a traffic signal can pass the signal in the next green light phase. In oversaturated city traffic on the other hand, not all vehicles can pass and therefore the queue of waiting vehicles grows with each cycle of the signal. In classical theories the oversaturated city traffic consists of a series of moving queues, sequences of stopped vehicles interrupted by sequences of vehicles that move from one moving queue to the next queue downstream. In general the duration of the vehicle stop within a moving queue does not change in average while the moving queue propagates upstream of the signal (see Figure 1).

It is usually assumed [1][2][3][4][5][6] that the transition from under- to over-saturated traffic, i.e., traffic breakdown at the signal, occurs at a classical capacity of traffic signal $C_{\rm cl} = q_{\rm sat} T_{\rm G}^{\rm (eff)}/\vartheta$, where $q_{\rm sat}$ is the saturation flow rate, i.e., the mean flow rate from a queue at the signal during the green phase when vehicles discharge from the moving queue to their maximum free speed $v_{\rm free}$. $\vartheta = T_{\rm G} + T_{\rm Y} + T_{\rm R}$ is the cycle time of the signal and assumed to be constant, $T_{\rm G}$, $T_{\rm Y}$, and $T_{\rm R}$ are the durations of the green, yellow, and red phases of the signal, respectively. $T_{\rm G}^{\rm (eff)}$ is the effective green phase time that is the portion of the cycle time during which vehicles are assumed to pass the signal at constant rate $q_{\rm sat}$.

Within anonymized vehicle probe data from oversaturated city traffic provided by TomTom, rather than this stop-and-go pattern built by the moving queues, synchronized flow according to the three-phase traffic theory by Kerner has been found [13].

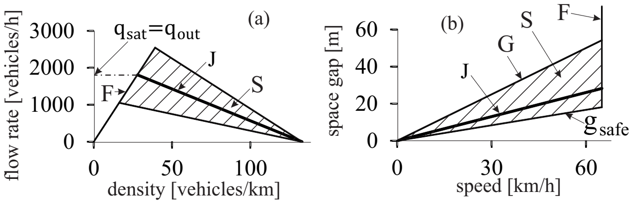


Figure 2. 2D-regions for steady states of synchronized flow in the flow–density (a) and the space-gap–speed planes (b) [10][11][12][18][19]. F – free flow, S – synchronized flow, J – line J [7][8].

In contrast with two-phase traffic flow models with a fundamental diagram (e.g., [14][15][16][17]), in the stochastic model [10][11][12] used for all simulations, there is a 2D-region of synchronized flow associated with the fundamental hypothesis of three-phase theory (see Figure 2) [7]. When a driver approaches a slower moving preceding vehicle and she/he cannot overtake it, then the driver begins to decelerate and adapts its speed to the speed of the preceding vehicle, when the gap g to the preceding vehicle becomes smaller than a synchronization gap G (see Figure 2 (b)). This driver's speed adaptation occurs under condition $g_{\rm safe} \leq g \leq G$, where $g_{\rm safe}$ is a safe gap.

We use a discrete version of a stochastic three-phase microscopic model of Kerner and Klenov [10][11][12]. The physics of the model variables are explained in [7][8]. The parameteres of the model have been adapted for city traffic in [18][19], the vehicle speed v_{n+1}, the coordinate x_{n+1}, and the acceleration A_{n+1} at time step $n+1$ are found from the following equations:

$$v_{n+1} = \max(0, \min(v_{\rm free}, \tilde{v}_{n+1} + \xi_n, v_n + a_{\rm max}\tau, v_{{\rm s},n})), \tag{1}$$

$$x_{n+1} = x_n + v_{n+1}\tau, \tag{2}$$

$$A_{n+1} = (v_{n+1} - v_n)/\tau, \tag{3}$$

$$\tilde{v}_{n+1} = \min(v_{\rm free}, v_{{\rm s},n}, v_{{\rm c},n}), \tag{4}$$

$$v_{{\rm c},n} = \begin{cases} v_{{\rm c},n}^{(1)} & \text{at } \Delta v_n + A_{\ell,n}\tau < \Delta v_{\rm a}, \\ v_{{\rm c},n}^{(2)} & \text{at } \Delta v_n + A_{\ell,n}\tau \geq \Delta v_{\rm a}, \end{cases} \tag{5}$$

$\Delta v_{\rm a}$ is constant.

$$v_{{\rm c},n}^{(1)} = \begin{cases} v_n + \Delta_n^{(1)} & \text{at } g_n \leq G_n, \\ v_n + a_n\tau & \text{at } g_n > G_n, \end{cases} \tag{6}$$

$$\Delta_n^{(1)} = \max(-b_n\tau, \min(a_n\tau,\ v_{\ell,n} - v_n)), \tag{7}$$

$$v_{{\rm c},n}^{(2)} = v_n + \Delta_n^{(2)}, \tag{8}$$

$$\Delta_n^{(2)} = k_{\rm a} a_n \tau \max(0, \min(1,\ \gamma(g_n - v_n\tau))), \tag{9}$$

$$a_{\rm max} = \begin{cases} a & \text{at } \Delta v_n + A_{\ell,n}\tau < \Delta v_{\rm a}, \\ k_{\rm a} a & \text{at } \Delta v_n + A_{\ell,n}\tau \geq \Delta v_{\rm a}, \end{cases} \tag{10}$$

$$a_n = a\Theta(P_0 - r_1),\ b_n = a\Theta(P_1 - r_1), \tag{11}$$

$$P_0 = \begin{cases} p_0 & \text{if } S_n \neq 1, \\ 1 & \text{if } S_n = 1, \end{cases} \qquad P_1 = \begin{cases} p_1 & \text{if } S_n \neq -1, \\ p_2 & \text{if } S_n = -1, \end{cases} \tag{12}$$

$$S_{n+1} = \begin{cases} -1 & \text{if } \tilde{v}_{n+1} < v_n, \\ 1 & \text{if } \tilde{v}_{n+1} > v_n, \\ 0 & \text{if } \tilde{v}_{n+1} = v_n, \end{cases} \tag{13}$$

$r_1 = {\rm rand}(0,1)$, $\Theta(z) = 0$ at $z < 0$ and $\Theta(z) = 1$ at $z \geq 0$, $p_0 = p_0(v_n)$, $p_2 = p_2(v_n)$, p_1 is constant.

$$\xi_n = \begin{cases} \xi_{\rm a} & \text{if } S_{n+1} = 1, \\ -\xi_{\rm b} & \text{if } S_{n+1} = -1, \\ \xi^{(0)} & \text{if } S_{n+1} = 0, \end{cases} \tag{14}$$

$$\xi_{\rm a} = a^{\rm (a)}\tau\Theta(p_{\rm a} - r),\ \xi_{\rm b} = a^{\rm (b)}\tau\Theta(p_{\rm b} - r), \tag{15}$$

$$\xi^{(0)} = a^{(0)}\tau \begin{cases} -1 & \text{if } r \leq p^{(0)}, \\ 1 & \text{if } p^{(0)} < r \leq 2p^{(0)} \text{ and } v_n > 0, \\ 0 & \text{otherwise}, \end{cases} \tag{16}$$

$r = {\rm rand}(0,1)$; $a^{\rm (b)} = a^{\rm (b)}(v_n)$; $p_{\rm a}$, $p_{\rm b}$, $p^{(0)}$, $a^{\rm (a)}$, $a^{(0)}$ are constants; synchronization gap G_n and safe speed $v_{{\rm s},n}$ are

$$G_n = G(v_n, v_{\ell,n}), \tag{17}$$

$$G(u,w) = \max(0, \lfloor k\tau u + a^{-1}\phi_0 u(u-w)\rfloor), \tag{18}$$

$$v_{{\rm s},n} = \min\left(v_n^{\rm (safe)}, g_n/\tau + v_\ell^{\rm (a)}\right), \tag{19}$$

$$v_n^{\rm (safe)} = \lfloor v^{\rm (safe)}(g_n,\ v_{\ell,n})\rfloor, \tag{20}$$

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-371-1

$v^{(\mathrm{safe})}\tau_{\mathrm{safe}} + X_{\mathrm{d}}(v^{(\mathrm{safe})}) = g_n + X_{\mathrm{d}}(v_{\ell,n})$, $X_{\mathrm{d}}(u) = b\tau^2\left(\alpha\beta + \frac{\alpha(\alpha-1)}{2}\right)$, $\alpha = \lfloor u/b\tau \rfloor$, $\beta = u/b\tau - \alpha$, $v_\ell^{(\mathrm{a})} = \max(0, \min(v_{\ell,n}^{(\mathrm{safe})}, v_{\ell,n}, g_{\ell,n}/\tau) - a\tau)$, τ_{safe} is a safe time gap; b, $k > 1$, a, k_{a}, and ϕ_0 are constants; $\lfloor z \rfloor$ denotes the integer part of a real number z. In (1)–(20), $n = 0, 1, 2, ...$ is the number of time steps, $\tau = 1\,\mathrm{s}$ is a time step, $a_{\max}$ is a maximum acceleration, v_{free} is a maximum speed in free flow, $\tilde{v}_n$ is the vehicle speed without speed fluctuations ξ_n, ℓ marks the preceding vehicle, $g_n = x_{\ell,n} - x_n - d$ is the space gap between vehicles, d is the vehicle length, $\Delta v_n = v_{\ell,n} - v_n$; x_{n} and v_{n} are measured in units $\delta x = 0.01\,\mathrm{m}$ and $\delta v = 0.01$ m/s, respectively.

In the model, if a vehicle reaches the upstream front of a moving queue at a signal it decelerates as it does at the upstream front of a wide moving jam propagating on a road without traffic signals [7][8]. During the green light phase, vehicles accelerate at the downstream front of the moving queue (queue discharge) with a random time delay as they do at the downstream jam front. During the yellow phase a vehicle passes the signal location, if the vehicle can do it until the end of the yellow phase; otherwise, the vehicle comes to a stop at the signal.

A key role in the simulation of synchronized flow plays the driver's speed adaptation for which we use a stochastic description through the probabilities p_1 and p_2 in (12). We write these probabilities as follows:

$$p_1 = \min(1, (1+\varepsilon)p_1^{(0)}), \quad p_2 = \min(1, (1+\varepsilon)p_2^{(0)}(v_n)), \tag{21}$$

where $p_1^{(0)} = 0.3$, $p_2^{(0)}(v_n) = 0.48 + 0.32\Theta(v_n - v_{21})$, ε is the coefficient of speed adaptation. The larger ε, the stronger the speed adaptation and, therefore, the larger the mean space gap (the longer the mean time headway) between vehicles in synchronized flow. This can be explained as follows.

If on a multilane road a driver approaches a slower moving vehicle in front and if he cannot change lane, e.g., because of traffic on the other lane not permitting a sufficient safety gap for a lane change, than the driver has to decelerate. In the model he can start to decelerate as soon as he enters the synchronization gap to his leader. The driver will decelerate if his distance to the leader becomes smaller than the safety gap, and he will accelerate if the distance becomes greater than the synchronization gap (see Figure 2 (b)). But in contrast to models based on classical theories, in three-phase traffic flow models a vehicle doesn't try to abide to a specific distance to the preceeding car, usually the safety gap. Instead, the vehicle tends to adapt its speed to the speed of the preceeding car, but it will do this while taking an arbitrary distance to it as long as it stays within the synchronization gap.

Therefore, the driver could decelerate gradually to the speed of the leader as soon as he enters the synchronization gap, which would lead to a short headway to the preceeding vehicle (see Figure 3 (a)). This situation, described with $\varepsilon = 0$, is called usual or weak speed adaptation and leads to the classical behaviour of stop-and-go traffic: the mean duration of the stops of the vehicles remain almost constant along the moving queues that build upstream of the signal [9]. Alternatively, the driver can decelerate more sharply when reaching the synchronization gap distance, which would lead to a large headway to the preceeding vehicle (see Figure 3 (b)). This situation, described with $\varepsilon > 0$, is called strong speed adaptation and leads to the behaviour that cannot be described by classical theories: the mean duration of the stops of the vehicles decreases the further upstream of the traffic signal the moving queue is located [9]. The growth of the space gap leads to the dissolution of the jam upstream of the traffic signal. So in oversaturated city traffic adjacent to the traffic signal there is still the classical stop-and-go behaviour of the vehicles, but further away from the signal synchronized flow emerges due the dissolution of the jam.

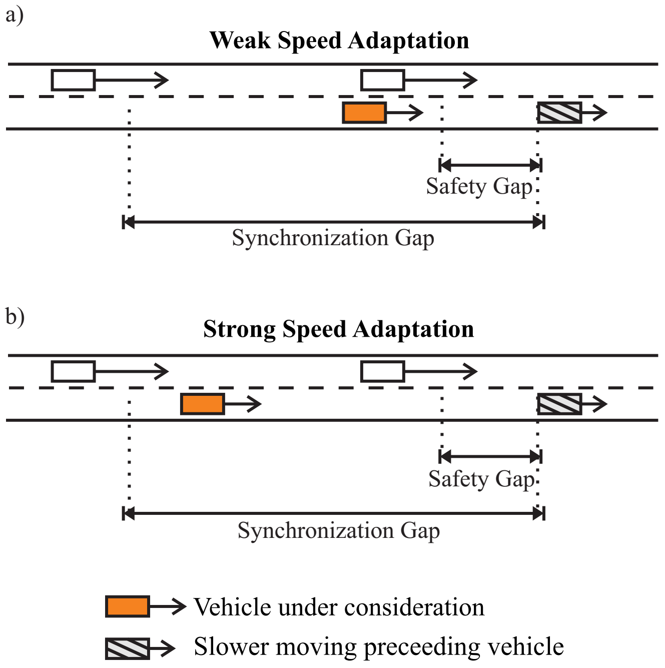


Figure 3. Explanation of speed adaptation effect: Weak speed adaptation (a) and strong speed adaptation (b).

III. Synchronized flow in city traffic

For the simulation of synchronized flow in urban traffic we use a multilane road with a traffic signal at the end. We chose a road section of Völklinger Straße in the city of Düsseldorf (see Figure 4) as congested traffic is observed here regularly in the data of a stationary video detector [13]. The length of the section is 630 m, the speed limit is 60 km/h, and there are no junctions in between.

A. Empirical findings

Figure 5 shows typical examples of anonymized empirical vehicle probe data provided by TomTom, the dotted (red) line denotes the point in time at which the vehicles pass the position of the traffic signal. Figure 5 (a) shows the velocity profile of a vehicle driving on Völklinger Straße in the morning rush hour, exhibiting the classical moving queue pattern in front of a traffic signal in oversaturated city traffic.

Figure 5 (b) shows an example of a velocity profile in the synchronized flow phase in the late morning rush hour: the vehicle at first drives with a speed between 20 km/h and 30 km/h without coming to a stop. Then as it moves further down the road it has to decelerate to a full stop about 100 m

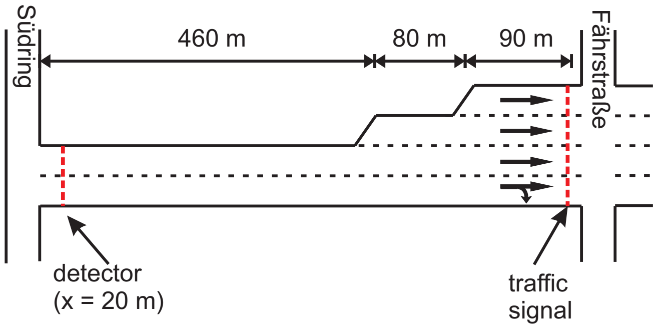


Figure 4. Layout of test track Völklinger Straße from Südring to Fährstraße. Video detector at $x = 20\,$m, traffic light at $x = 630\,$m (cycle time $\vartheta = 70\,$s, red phase duration $T_{\rm R} = 35\,$s, yellow phase duration $T_{\rm Y} = 4\,$s), speed limit $v_{\max} = 60\,$km/h.

in front of the traffic signal as it reaches the upstream front of the queue. In accordance with the microscopic criteria for traffic phases in congested traffic [7][8], this is a typical empirical example of synchronized flow in oversaturated city traffic. Following classical theories one would expect either a stop-and-go pattern along the road, or a higher speed at the beginning of the section followed by one or more stops in front of the traffic signal.

Finally, Figure 5 (c) shows an example of a vehicle driving through heavy congested traffic: the travel time for the 630 meters of Völklinger Straße is about 12 minutes.

B. *Simulation of synchronized flow*

For our simulations we used infrastructure information from OpenStreetMap [20] that was improved by the use of infrastructure plans from the city of Düsseldorf that also provided the traffic signal cycle plans. The amount of vehicles emitted per minute into the simulation was matched to the data of the video detector at the beginning of the road. In all simulations we use the following parameters of model ((1)–(21)): $\tau_{\rm safe} = \tau = 1$, $d = 7.5\,$m, $v_{\rm free} = 18.0558\,{\rm ms}^{-1}$ (65 km/h), $b = 1\,{\rm ms}^{-2}$, $a = 0.5\,{\rm ms}^{-2}$, $k = 3$, $\phi_0 = 1$, $\Delta v_a = 2\,{\rm ms}^{-1}$, $k_a = 4$, $\gamma = 1$, $p_b = 0.1$, $p_a = 0.03$, $p^{(0)} = 0.005$, $p_0(v_n) = 0.667 + 0.083\min(1, v_n/v_{01})$, $v_{01} = 6\,{\rm ms}^{-1}$, $v_{21} = 7\,{\rm ms}^{-1}$, $a^{(a)} = a$, $a^{(0)} = 0.2a$, $a^{(b)}(v_n) = 0.2a + 0.8a\max(0, \min(1, (v_{22} - v_n)/\Delta v_{22})$, $v_{22} = 7\,{\rm ms}^{-1}$, $\Delta v_{22} = 2\,{\rm ms}^{-1}$, $\varepsilon = 0$. Open boundary conditions have been used in all simulations.

Figure 6 (a) shows the speed profile of a vehicle driving in free flow, and Figure 6 (b) shows the according acceleration profile: after entering the road section, the vehicle drives (nearly) at maximum speed at first. Then it decelerates and comes to a stop in front of the traffic signal. In the next green light phase it passes the traffic signal position and again accelerates to maximum speed. This is a typical simulation example for the situation of undersaturated city traffic. Since we describe free flow, the speed adaptation effect doesn't come into it.

Figures 7 (a) and (b) show the speed and acceleration profile of a vehicle driving in synchronized flow. The pattern is very similar to the one found in the empirical data (see Figure 5 (b)): the vehicle first drives with a speed between 10 km/h and 20 km/h. As it moves further down the road it first decelerates to a full stop at the end of the queue in front of the traffic signal and then a second time about 20 seconds later within the queue. This is an example of synchronized flow pattern reproduced in the simulation.

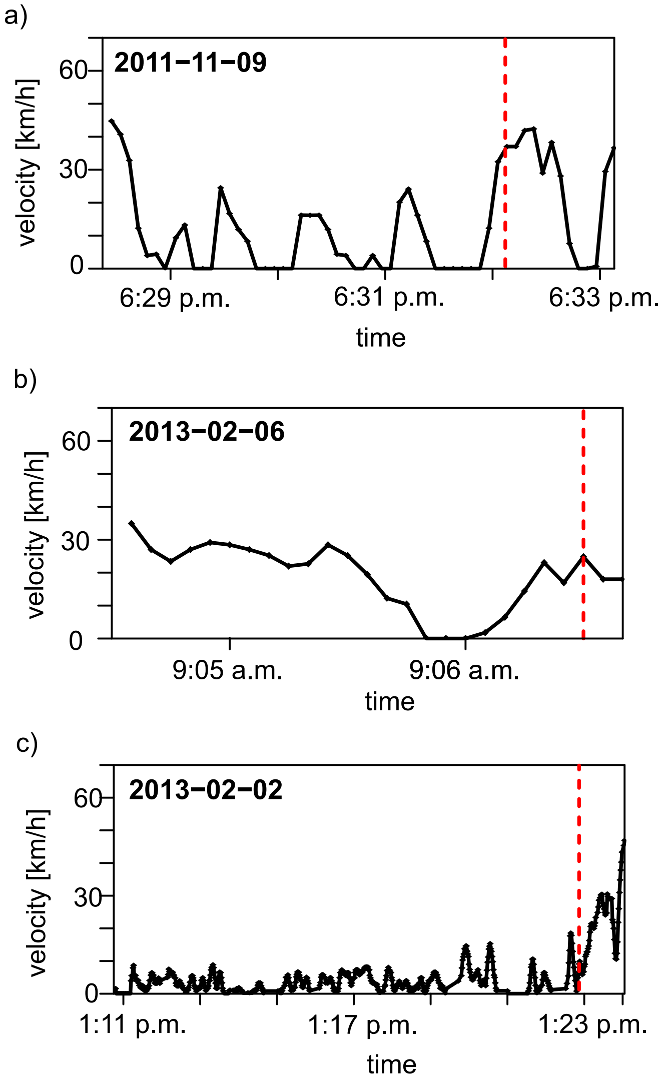


Figure 5. Empirical examples of speed profiles: classical stop-and-go pattern (moving queues) (a), synchronized flow (b) and highly congested traffic pattern (c). The dotted (red) line denotes the position of the traffic light.

Finally, Figures 8 (a) and (b) show the speed and acceleration profile of a vehicle driving in heavily congested traffic. The vehicle is following a typical stop-and-go pattern along the entire strech of the road up to the position of the traffic signal.

All simulation results shown above are representatives of traffic patterns that were found in the empirical data. This demonstrates that we can reconstruct in our simulations the urban traffic patterns known from classical theories like stop-and-go traffic, as well as synchronized traffic that is unknown to classical theories but found in real data. By alternating the strength of the speed adaptation effect we can reproduce all traffic patterns found in the empirical data for synchronized flow.

As described in Section II, the stronger the speed adaptation effect, the larger the space gap (time headway) that a driver chooses on average to the preceeding vehicle. Since the drivers in this case have enough space in front and can decelerate gradually, they come to a stop less often and therefore the

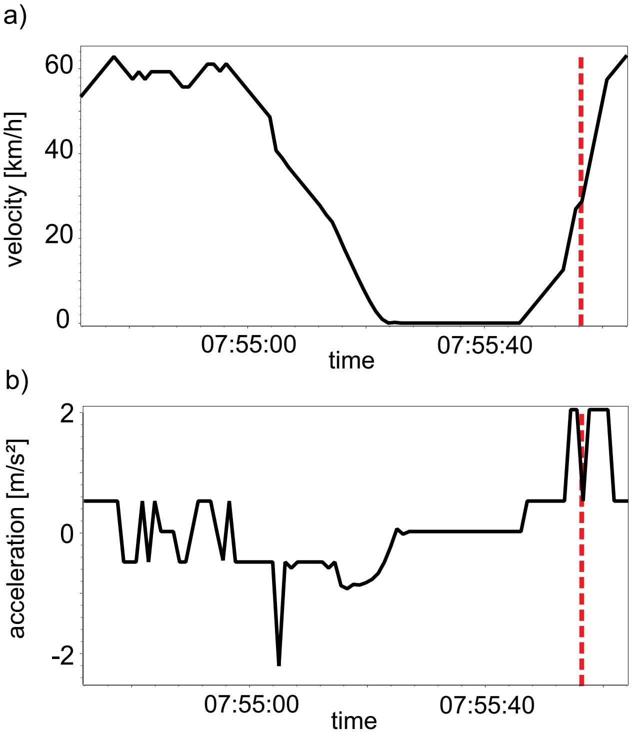


Figure 6. Typical example of simulation results for free flow: time-dependance of microscopic (single) vehicle speed (a), and the according acceleration profile (b) The dotted (red) line denotes the position of the traffic light. $\varepsilon = 0$.

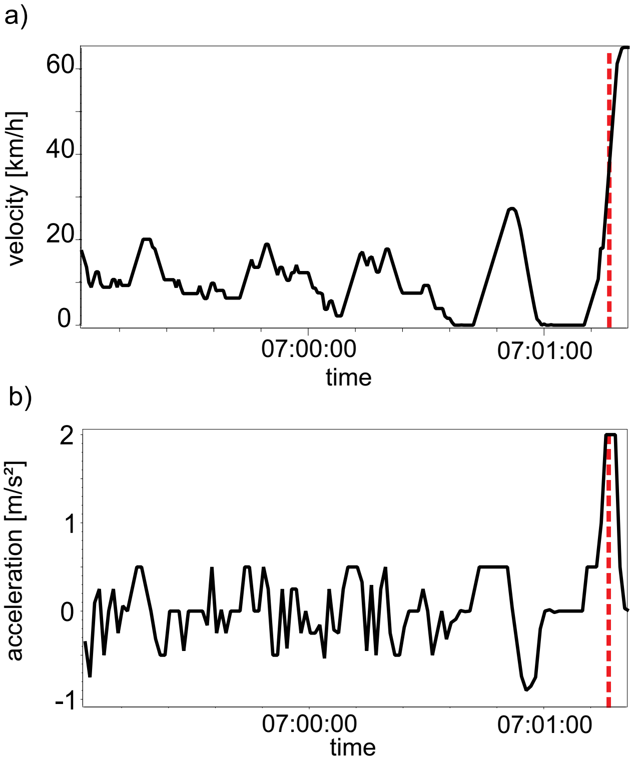


Figure 7. Typical example of simulation results for synchronization flow. The dotted (red) line denotes the position of the traffic light. Model parameters are the same as those in Figure 6 with the exception of the value of ε, that is taken as $\varepsilon = 1.333$.

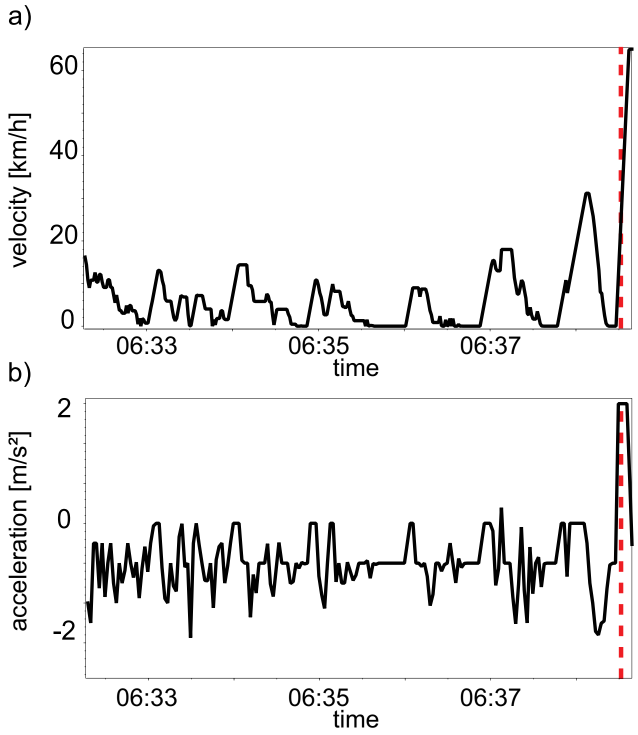


Figure 8. Typical example of simulation results for congested traffic. The dotted (red) line denotes the position of the traffic light. Model parameters are the same as those in Figure 6 with the exception of the value of ε, that is taken as $\varepsilon = 1.333$.

mean stop duration of the vehicle within the moving queue decreases. As a result, the absolute value of the upstream front velocity of a moving queue becomes smaller than that of the downstream front of the moving queue, resulting in moving queue dissolution. The moving queues dissolve into the synchronized flow phase. The strength of the speed adaptation effect influences the distance from the traffic signal, at which the moving queues dissolve. The greater ε, the shorter the distance from the traffic signals at which the moving queues dissolve [9].

IV. Conclusion and Future Work

Empirical probe vehicle data measured by TomTom navigation devices show synchronized flow in oversaturated city traffic [13]. We have simulated oversaturated city traffic with the discrete stochastic microscopic Kerner-Klenov three-phase traffic flow model. The simulations show that under strong speed adaptation synchronized flow patterns which are very close to empirical data can be reproduced with this model. Strong speed adaptation is associated with an average increase of space gaps (time headways) which drivers choose moving in very dense city traffic. The stronger the speed adaptation effect, the shorter the distance at which moving queues dissolve into synchronized flow. Further work will include the evaluation of the strength of the speed adaptation effect as well as comparisons to more empirical examples.

Acknowledgment

The authors would like to thank our partners for their support in the project “UR:BAN - Urban Space: User oriented assistance systems and network management”, funded by the

German Federal Ministry of Economic Affairs and Energy by resolution of the German Federal Parliament.

REFERENCES

[1] F. V. Webster, “Traffic signal settings,” Road Research Technical Paper, vol. 39, 1958.

[2] G. F. Newell, “Approximation methods for queues with application to the fixed-cycle traffic light,” SIAM Review, vol. 7, no. 2, 1965, pp. 223–240.

[3] J. D. C. Little, “The synchronization of traffic signals by mixed-integer linear programming,” Operations Research, vol. 14, no. 4, 1966, pp. 568–594, ISSN: 0030-364X.

[4] D. I. Robertson, Transyt: A traffic network study tool. Transportation and Road Research Laboratory, Crowthorne, UK, 1969, Report no. LR 253, in Transportation and Road Research Laboratory Report, ISBN: 0968-4093.

[5] N. H. Gartner, and C. Stamatiadis, Traffic Networks, Optimization and Control of Urban. Springer, New York, 2009, pp. 9470–9500, in Meyers, R. A., Ed., Encyclopedia of Complexity and System Science, ISBN: 978-0-387-75888-6.

[6] F. Dion, H. Rakha, and Y.-S. Kang, “Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections,” Transportation Research Part B: Methodological, vol. 38, no. 2, 2004, pp. 99–1222, ISSN: 0191-2615.

[7] B. S. Kerner, The Physics of Traffic. Springer, Berlin, New York, 2004, ISBN: 978-3-540-40986-1.

[8] B. S. Kerner, Introduction to Modern Traffic Flow Theory and Control. Springer, Berlin, New York, 2009, ISBN: 978-3-642-02605-8.

[9] B. S. Kerner et al., “Synchronized flow in oversaturated city traffic,” Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, vol. 88, no. 5, 2013, 054801.

[10] B. S. Kerner and S. L. Klenov, “A microscopic model for phase transitions in traffic flow,” Journal of Physics A: Mathematical and General, vol. 35, no. 3, 2002, L31-L43.

[11] B. S. Kerner and S. L. Klenov, “Microscopic theory of spatial-temporal congested traffic patterns at highway bottlenecks,” Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, vol. 68, no. 3, 2003, 036130.

[12] B. S. Kerner and S. L. Klenov, “Phase transitions in traffic flow on multilane roads,” Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, vol. 80, no. 5, 2009, 056101.

[13] P. Hemmerle et al., Increased Consumption in Oversaturated City Traffic Based on Empirical Vehicle Data. Springer International Publishing, 2014, pp. 71–79, in Fischer-Wolfarth, J. and Meyer, G., Eds., Advanced Microsystems for Automotive Applications 2014. Lecture Notes in Mobility, ISBN: 978-3-319-08086-4.

[14] P. G. Gipps, “A behavioural car-following model for computer simulation,” Transportation Research Part B: Methodological, vol. 15, no. 2, 1981, pp. 105–111, ISSN: 0191-2615.

[15] K. Nagel and M. Schreckenberg, “A cellular automaton model for freeway traffic,” Journal de Physique I, France 2, 1992, pp. 2221–2229.

[16] R. Barlović, L. Santen, A. Schadschneider, and M. Schreckenberg, “Metastable states in cellular automata for traffic flow,” The European Physical Journal B - Condensed Matter and Complex Systems, vol. 5, no. 3, 1998, pp. 793–800, ISSN: 1434-6028.

[17] S. Krauß, P. Wagner, and C. Gawron, “Metastable states in a microscopic model of traffic flow,” Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, vol. 55, no. 5, 1997, pp. 5597–5602.

[18] B. S. Kerner, “Theory of self-organized traffic at light signal,” arXiv:1211.2535v1 [physics.soc-ph], retrieved: August, 2014.

[19] B. S. Kerner, “The physics of green-wave breakdown in a city,” Europhysics Letter, vol. 102, no. 2, 2013, pp. 28010.

[20] www.openstreetmap.org

An Optimal Multiobjective Production System: A Case Study

Hector Miguel Gastelum Gonzalez
Information Technologies
Universidad de Guadalajara
Guadalajara, México
gastelumg@icloud.com

Maria Elena Meda Campaña
Information Systems Department
Universidad de Guadalajara
Guadalajara, México
mmeda.campana@gmail.com

***Abstract*— In this paper, a case study is presented to improve the performance of a Tires Production System implemented as a Material Requirement Planning. In our proposal, the Master Production Schedule is calculated from a percentage of the demand's forecast per period, and production begins with the arrival of customer orders. The improvement of Tires Production Systems comprises to work with the multi-objective genetic algorithm method NSGA-II and with the results comparison of the duo Simulated Binary Crossover - Parameter based against the Whole Arithmetical Crossover Mutation - Mutation Uniform. The simulation results show that the minimum values are obtained with the pair Simulated Binary Crossover - Mutation-based Parameter, and also in fewer generations number.**

Keywords-multiobjective optimization; nsga-II; production systems; genetic algorithm

I. Introduction

This research starts with the need to improve the performance of a Tire Production System (TPS). TPS is implemented as a Material Requirements Planning (MRP) System. The improvement is made at the Planning and Production Control System (PPCS) of the TPS in order to satisfy demand without increasing inventory.

A PPCS can be implemented as a Push System (PHS) or as a Pull System (PLS). A MRP is a type of Push System [1], while a Kanban System is considered to be a Pull System [1]. In a PHS, the Forecasted Demand represents the signal to start production, and its main feature is to satisfy the demand in exchange for an inventory of finished products. A Kanban System starts production with the arrival order signal; its main feature is to decrease inventory in exchange of the increasing the risk to do not meeting the demand.

Both PPCS approaches (PLS and PHS) were created for different manufacturing environments. PLS works well in environments of assembly of components, where components are assembled into finished products, while PHS works well in processes that supply products to other production processes as well as processes that involve perishable products [2].

Since late 80's, some researchers have explored different combinations of the advantages of PHS and PLS and they have defined a new kind of PPCS known as Hybrid Push-Pull Systems (HPPS). These kinds of systems merge the characteristics of PHS and PLS in order to improve Production Systems performance [1].

The objective of this paper is to improve the performance of a TPS. This is accomplished by taking the advantages from both PHS and PLS. It is proposed to use MRP for Master Production Schedule (MPS) and Material Plan (MP). For the execution of the production, the arrival of the orders as a signal to start production is proposed. The performance is measured with two indicators: Unmet Demand and Raw Material Inventory. A Multiobjective Genetic Algorithm (MGA) is used in order to find the minimum values of performance indicators.

The paper is organized as follows: the related literature with the HPPS that improve the performance of a MRP system is cited in Section II. Section III presents the proposal to improve the TPS on study, considering the elements to integrate to the MRP system implemented in the TPS, the MGA method and the crossover and mutation techniques; in the materials and methods section, the general methodology is defined as well as the structure of the MGA method and the information related to the case study. The results and conclusions are presented in Section V and Section VI, respectively.

II. Literature Review

This section considers the researches related with HPPS that improve the performance of a production system from a MRP system. Below, authors and description of the HPPS are presented:

- Hall [3] uses MRP and Kanban synchronized in the joint point of shop floor and planning, in order to start to produce.
- Vaughn [4] uses MRP for medium to long term planning and uses Kanban for shop floor.
- Lee [5] integrates MRP and Just in Time in a single framework, to take the planning advantage of MRP and the execution advantages of JIT.

ISBN: 978-1-61208-371-1

- Ke et al. [6] use a strategy composed by a Push element for the procurement and a Pull element for production.
- Takahashi et al. [7] integrate the Push and Pull controls to calculate orders for the assembly and distribution stages using PHS and PLS respectively.
- Nagendra [8] uses JIT/Kanban in the shop floor and MRP for planning process.
- Gupta et al. [9] make a selection of the number and size of Kanban implemented in an MRP system.
- O'Grady [10] presents a model in which the demand is the signal for MRP to make the planning stage, and the execution stage works with PLS.
- Bushée et al. [11] make a method for scheduling job shops that combines PHS and PLS.
- Beamon et al. [12] apply the PHS from the beginning of the process until the components are produced; in these points the system changes to PLS to assemble the components into finished products.
- Flapper et al. [13] embed JIT into MRP in three steps, from a production process operated by MRP. Step 1: create a logical line flow through rapid material handling; Step 2: use a PLS on the logical line; Step 3: make the layout in a flow line.
- Huq et al. [14] use a PLS in a job shop with some variations in: processing times, load levels, and machine breakdowns.
- Lin et al. [15], base the production system on the forecast of production to produce the components of the final products with the push system, and the final products are assembled according to customer orders, with Pull system.

HPPS were made to improve the indicators performance of a PPCS, which highlights the inventory of raw materials and demand satisfaction, among other performance indicators.

According to the literature reviewed, the proposed HPPS are structured so that the main advantages of each PHS and PLS, i.e., MRP in the planning stage and JIT at the production stage.

The proposed HPPS do not consider changes in the proportion of Forecasted Demand in the planning production stage, while working with PHS and PLS in the production stage.

III. PROPOSAL

Based on the literature review, a proposal is made to improve a TPS. The TPS under study operates according to a MRP system. The improvement is based under the following considerations: the TPS in the stages of MPS and MP will operate as a MRP system [16], In the Push stage, the arrival orders as the starting production signal is proposed, like in a Pull System [17].

To improve the performance of the TPS, it is proposed to calculate the MPS from a percentage of the forecasted demand and to produce according to demand. TPS improvement is measured through the performance indicators: Raw Materials Inventory and Unmet Demand. The Raw Material Inventory is the amount of raw material accumulated at the beginning of the production process at end of period. Unmet Demand is the amount of demand that is not satisfied at the end of the period.

To improve the TPS, a MGA is used. The MGA are used when an optimization problem has two or more objective functions and the search space is very large. The problem to improve the TPS is an optimization problem consisting in two objective functions and continuous decision variables. The search space size is calculated with a combination of the decision variables of all the periods. This problem is defined later. According to our previous statements, it can be said that it is appropriate to use a MGA.

There exists some methods of MGA in the literature. NSGA-II proposed by Deb [18] is proven to have a better performance than other similar methods, such as Pareto-archived Evolution Strategy (PAES) [19] and Strength-Pareto Evolutionary Algorithm (SPEA) [20]. These three MGA methods use elitism. NSGA-II gets better results with real code representation than binary code representations [18]. These are the reasons why this method is selected to improve the TPS.

NSGA-II proposes to work with Simulated Binary Crossover (SBX) and the Parameter-based Mutation (PBM), for validation experiments [21]. However, in this research, the mutation and crossover techniques are alternated, in one side we use SBX along with PBM, and on the other side the Whole Arithmetical Crossover (WAX) is used along with Uniform Mutation (UM) [22]. All crossover and mutation techniques used in this work, are defined in the next section.

To improve the TPS, there is an optimization problem where the objective functions are: to Minimize Raw Material Inventory and to Minimize Unmet Demand, the decision variables are the Percentages of Demand Forecasted per period and the Demand has random arrivals according to a Poisson Probability Distribution with known mean.

To get the minimum values of performance variables for the TPS, both crossover and mutation techniques are compared in order to find which of the two pairs defined previously show better values on performance indicators: Raw Materials Inventory and Unmet Demand, as well as which of the techniques takes less time. The two pairs are chosen because both are applied to real-coded representation in genetic algorithms and both use bounds for the decision variables; moreover, the crossover is applied to all elements of the chromosome, by genetic algorithms.

IV. MATERIALS AND METHOD

In this section, the following is described: 1) the methodology applied to improve the TPS, 2) the MGA used for simulation including the crossover and mutation techniques as well as the evaluation function, and 3) the case study.

A. *Methodology*

The methodology used in this research is mentioned below:

- Propose the improvement of TPS with MGA.
- Describe the Multiobjective Genetic Algorithm to improve the TPS.
- Define the case study.
- Simulate the Multiobjective Genetic Algorithm to improve TPS, considering the two pairs of crossover and mutation.
- Compare the results of the two pairs of crossover and mutation.
- Analyze the results.
- Select the pair of crossover and mutation with better results.

All elements of the methodology are described and performed in different sections of the paper.

B. *Description of the Multiobjective Genetic Algorithm to improve the TPS.*

NSGA-II method is the basis for building the MGA to improve the TPS performance, it has the following characteristics: the representation of the variables is in real code, and Inventory of Raw Materials and Unmet Demand in the system are the two objective functions to evaluate. SBX - PBM and WAX-UM are used for crossover and mutation.

1) *Crossover and Mutation techniques*

In this work, two methods of crossover and mutation are applied. They are described below.

a) *Simulated Binary Crossover (SBX)*

The formulas for Simulated Binary Crossover [21] are presented below:

- Create a random number *u* between 0 and 1.
- Find a parameter β_q, as follows:

$$\beta_q = \begin{cases} (u \propto)^{\frac{1}{nc+1}} & if\ u \le \frac{1}{\propto} \\ \left(\frac{1}{2-u\propto}\right)^{\frac{1}{nc+1}}, & otherwise \end{cases}$$

where $\propto = 2 - \beta^{-(nc+q)}$ and β is calculated as follows:

$$\beta = 1 + \frac{2}{y2 - y1} \min[(y_1 - y_l), (y_u - y_2)]$$

The parameter y is assumed to vary in the interval $[y_l, y_u]$.

The children solutions are then calculated as follows:

$$c_1 = 0.5\ [\ (y_1 + y_2) - \beta q\ |y_2 - y_1|\ ]$$
$$c_2 = 0.5\ [\ (y_1 + y_2) + \beta q\ |y_2 - y_1|\ ]$$

It is assumed that $y_1 < y_2$.

b) *Parameter Based Mutation (PBM)*

The methodology to calculate the Parameter Based Mutation [21] is presented below:

- Create a random number *u* between 0 and 1.
- Calculate the parameter δ_q as follows:

$$\delta_q = \begin{cases} [2u + (1-2u)(1-\delta)^{nm+1}]^{\frac{1}{nm+1}}] - 1, \\ if\ u\ \le 0.5 \\ 1 - \ [\ 2(1-u) + 2(u-0.5)(1-\delta)^{nm+1}]^{\frac{1}{nm+1}}, \\ otherwise \end{cases}$$

$$\delta = \min\ \{[(y - y_l\ ),((y_u - y)]/(y_u - y_l)\}.$$

- Calculate the mutated child as follows:

$$c = y + \ \delta q\ (\ y_u - \ y_l\ )$$

c) *Whole Arithmetical Crossover (WAX)*

The formulas for Whole Arithmetical Crossover [22] are presented below:

$$a\ \epsilon \begin{cases} [\max(\propto, \beta), \min(\gamma, \delta)] & si\ v_k\ >\ w_k \\ [0,0] & si\ v_k =\ w_k \\ [\max(\gamma, \delta), \min(\propto, \beta)] & si\ v_k <\ w_k \end{cases}$$

In order to calculate *a*:

$$\propto = \frac{l_k^w - w_k}{v_k - w_k} \quad \beta = \frac{u_k^v - v_k}{w_k - v_k} \quad \gamma = \frac{l_k^v - vw_k}{w_k - v_k} \quad \delta = \frac{u_k^w - w_k}{v_k - w_k}$$

Where each value v_k exist within range $[l_k^v, u_k^v]$ and each value w_k exist within range $[l_k^w, u_k^w]$.

The children are constructed as follow:

$$P_1 = (v_1, \ldots, v_m) \text{ and } P_2 = (w_1, \ldots, w_m)$$

$$c_1 = [\ w_1 * a + v_1 * (1-a), \quad \ldots, \quad w_m * a + v_m * (1-a)]$$
$$c_2 = [\ v_1 * a + w_1 * (1-a), \quad \ldots \quad , \quad v_m * a + w_m * (1-a)]$$

d) *Uniform Mutation*

The methodology to calculate Uniform Mutation [22] is described below:

Given $P = [\ V_1, V_k', V_m\ ]$, the mutated individual will be $P' = [\ V_1, V_k', V_m\ ]$. $V_k' = rnd(low\ bound, upper\ bound)$.

The minimum and maximum ranges of the variables are used.

2) *Procedure for evaluation function*

The procedure to calculate evaluation function or fitness is described as follows. These steps are added to the NSGA-II method, and they are related to the case study:

- To input the Demand Forecasted and the number of planning periods.
- To calculate MPS. This is the multiplication of the Percentage of Forecast by Demand Forecasted, per period. The percentage of forecast takes values between 0 and 1.5.
- To generate Demand. The demand is generated randomly, following a Poisson distribution with known mean.
- To calculate the evaluation functions or Fitness: Raw Material Inventory and Unmet Demand.

```
if MPS ( i , j ) = Demand ( i , j )
        Raw Material Inventory ( i , j )  = 0
        Unmet Demand ( i , j )  = 0

if MPS ( i , j )  <  Demand ( i , j )
        Unmet Demand ( i , j ) = (Demand ( i , j ) – MPS ( i , j ) / Demand ( i , j )
        Raw Material Inventory ( i , j )  = 0
if MPS ( i , j )  > Demand ( i , j )
        Raw Material Inventory ( i , j ) = ( MPS ( i , j ) – Demand ( i , j ) / MPS ( i , j )
        Unmet Demand ( i , j )  = 0;

Fitness ( i ) = [ ∑  Unmet Demand ( i , j )
                  ∑    Inventory ( i , j ) ]
```

- To obtain the final solution: get the set of nondominated solutions of the final population P, and then select the minimum value, one that provides the mimimum accumulated values of Raw Materials Inventory and Unmet Demand.

Calculations of Raw Material Inventory and Unmet Demand are performed for each set of solutions generated by the algorithm; *i* represents the set of solutions of each population, and *j* represents the number of planning periods.

To construct the MGA that improves the TPS the steps of the NSGA-II method are applied. The way of making the crossover and mutation must be changed with SBX-PBM and WAX-UM, with the respective change on the steps above described for the evaluation function.

The values of the parameters for the MGA are: population size of 100, probability of crossover 90%, mutation probability of 17%, maximum number of generations of 250. The probability of Mutation is (1 / (number of decision variables)). The value of *nc* for SBX calculation is 20 and *nm* and for PBM calculations is 20 as well. These values are set as recommended by NSGA-II method. Chromosome is formed as follows: [Percentage of forecast to period 1 … Percentage of forecast of period *p*], the number of genes depends on the number of periods.

C. *Case Study*

The TPS produces 30 different tire sizes; it has a total production capacity of 400,000 tires monthly, TPS makes its planning for the next 6 months, the average demand is 390,000 tires per month. The forecast of the demand for the 6 periods is in Table I.

TABLE I. DEMAND FORECASTED BY PERIOD

Forecast of Demand					
Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
389.509	390.558	390.210	390.905	389.351	388.695

Simulations are performed to find the minimum values of Inventory of Raw Material and Unmet Demand for TPS. The results are analyzed by SBX-PBM and WAX-UM; the duo that generates the minimum values for both objective functions and do it in less generations, is defined as the final solution for the case study. The simulations are performed in MATLAB 2012-A ®.

D. *Simulation Procedure*

The simulation procedure for MGA to improve TPS starts with the input data of the case study and the parameters of NSGA-II method, considering the evaluation function shown in B 1. The simulation is carried out using MGA with crossover and mutation SBX-PBM and with WAX-UM techniques.

V. RESULTS

The results of the simulation with SBX-PBM and WAX-UM are presented in Table II and Table III, respectively. These are the percentages of the forecast. Figures 1 to 5 show the values of Raw Material Inventory and Unmet Demand, with SBX-PBM and WAX-UM, for the five simulations with 250 generations.

The graphics in the left side in all figures show the results obtained by the SBX-PBM; they achieve the minimum values for Raw Materials Inventory and Unmet Demand, between 50 and 100 generations. Also, the graph shows that from generation 100 the values achieved were very similar, and very close to zero. They also notice that the values generated for the two objectives initially, reach a value of 50. Both objectives are minimized at the same time for SBX-PBM.

The graphics on the right side in all figures show that the values of the objectives reach values up to 75 at the beginning, for WAX-UM. The behavior of the values Inventory of Raw Material and Unmet Demand fail to stabilize at 250 generations and values generated do not reach the minimum in both objectives at once.

TABLE II. RESULTS OF SBX-PBM BY SIMULATION AND BY PERIOD

Simulation	Period					
	1	2	3	4	5	6
1	1.0018	0.9986	0.9978	0.9974	1.0003	1.0036
2	1.0030	0.9997	0.9983	0.9977	1.0021	1.0034
3	1.0003	1.0009	0.9970	0.9981	1.0025	1.0031
4	1.0007	0.9962	0.9985	0.9957	1.0017	1.0035
5	1.0013	0.9986	0.9994	1.0007	1.0015	1.0020

TABLE III. RESULTS OF WAX-UM BY SIMULATION AND BY PERIOD

Simulation	Period					
	1	2	3	4	5	6
1	0.9982	0.9936	1.0187	0.9897	1.0098	1.0074
2	0.9735	0.9929	0.9853	0.9901	0.9909	1.0079
3	0.9909	1.0314	1.0109	0.9938	0.9994	0.9778
4	0.9981	0.9887	0.9997	1.0029	1.0076	1.0415
5	1.0041	0.9772	0.9844	0.9929	0.9994	0.9996

VI. CONCLUSIONS AND FUTURE WORK

After carrying out the simulation of the MGA to optimize the TPS improved by making changes in crossover and mutation techniques using the pairs SBX-PBM and WAX-UM, and based on the results depicted in the graphics of the section above, it can be concluded that:

- SBX-PBM gets better results for the two objectives: Raw Materials Inventory and Unmet Demand than WAX-UM.
- SBX-PBM reaches minimum values in fewer generations than WAX-UM, for both objectives.
- SBX-PBM gest values closer to zero than WAX-UM.

The best performance in relation to Inventory of Raw Material and Unmet Demand is generated for the pair SBX-PBM. Therefore it can be concluded for this case study that TPS is improved with MGA using the techniques of crossover and mutation SBX and PBM.

It is recommended to apply the improvements proposed for TPS, with demand arrivals and production capabilities proven in the case study, as well as demand arrivals following a Poisson distribution and the parameters fixed for the NSGA-II.

According to the objective, MGA should be used to improve TPS, with the technique of crossover and the technique of SBX mutation with PBM, with the following parameters *nc* = 20 and *nm*= 20, with 90% probability of crossover, 17% probability of mutation, 250 generations, population size of 100, for 6 planning periods.

As future work, it is planned to set this work as a Hybrid Push-Pull System. It is planned also to experiment with different production capabilities.

REFERENCES

[1] J. Geraghty and C. Heavey, "A review and comparison of hybrid and pull-type production control strategies", OR Spectrum, vol. 27, Jun 2005, pp. 435-457, DOI: 10.1007/s00291-005-0204-z

[2] F. R. Jacobs, W. L. Berry, D. C. Whybark, T. E. Vollman, Manufacturing Planning and Control for Supply Chain Management, McGrawHill, 2011, ISBN 978-0-07-175031-8

[3] R.W. Hall, "Syncro MRP: Combining Kanban and MRP – The Yamaha PYMAC System in Driving the Productivity Machine", Production Planning and Control in Japan, APICS, 1981, pp. 43-56

[4] O. Vaughn, "Are MRP and JIT compatible?," Journal of Applied Manufacturing Systems, vol. 2, 1988, pp. 17-22

[5] C. Y. Lee, "A recent development of the integrated manufacturing system: a hybrid of MRP and JIT", International Journal of Operations and Production Management, vol. 13, no. 4, 1993, pp. 3-17, DOI: 10.1108/01443579310027752

[6] K. Ke, Y. Jin, and H. Zhang, "A Hybrid Push-Pull Model Based MultiAgent Supply-Chain System With Equilibrium Analysis", in IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07), 2007, pp. 457-463, DOI 10.1109/IAT.2007.84

[7] K. Takahashi and N. Nakamura, "Push, pull, or hybrid control in supply chain management", International Journal of Computer Integrated Manufacturing, vol. 17, no. 2, Mar 2004, pp. 126-140, DOI: 10.1080/09511920310001593083

[8] P.B. Nagendra and S.K. Das, "MRP/sfx: a kanban-oriented shop floor extension to MRP", Production Planning & Control, vol. 10, no. 3, Apr 1999, pp. 207-218, DOI: 10.1080/095372899233172

[9] S. M. Gupta and L. Brennan, "A knowledge based system for combined Just-in-Time and material requirements planning", Computers Electrical Engineering, vol. 19, no. 2, 1993, pp. 157–174, DOI: 10.1016/0045-7906(93)90044-R

[10] P. J. O'Grady, "Putting the just-in-time philosophy into practice: a strategy for production managers", 1988, ISBN 978-94-011-7810-5

[11] D.C. Bushée and J.A. Svestka, "A bi-directional scheduling approach for job shops", International Journal of Production Research, vol. 37, no. 16, 1999, pp. 3823–3837, DOI:10.1080/002075499190077

[12] B. M. Beamon and J.M. Bermudo, "A hybrid push/pull control algorithm for multi-stage, multi-line production systems", Production Planning & Control, vol. 11, no. 4, 2000, pp. 349–356, DOI:10.1080/095372800232072

[13] S. D. P. Flapper, G. J. Miltenburg, and J. Wijngaard, "Embedding JIT into MRP", International Journal of Production Reserach, vol. 29, no. 2, 1991, pp. 329-341, DOI:10.1080/00207549108930074

[14] Z. Huq, and F. Huq, "Embedding JIT in MRP: The case of job shop", Journal of Manufacturing Systems, vol. 13, no. 3, 1994, , pp. 153-164

[15] J. Lin, X. Shi, and Y. Wang, "Research on the Hybrid Push/Pull Production System", in WEB 2011, LNBIP 108, Shanghai, China, 2012, pp. 413–420.

[16] C. A. Ptak, CFPIM, CIRM, C.J. Smith, Orlicky´s Material Requirements Planning, McGrawHill, 2011, ISBN 978-0-07-175563-4

[17] D. Sipper and R. Bulfin, Production: Planning, Control, and Integration, New York: McGraw-Hill, 1997. ISBN 0070576823

[18] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan,. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, April 2002, pp. 182-197, DOI: 10.1109/4235.996017

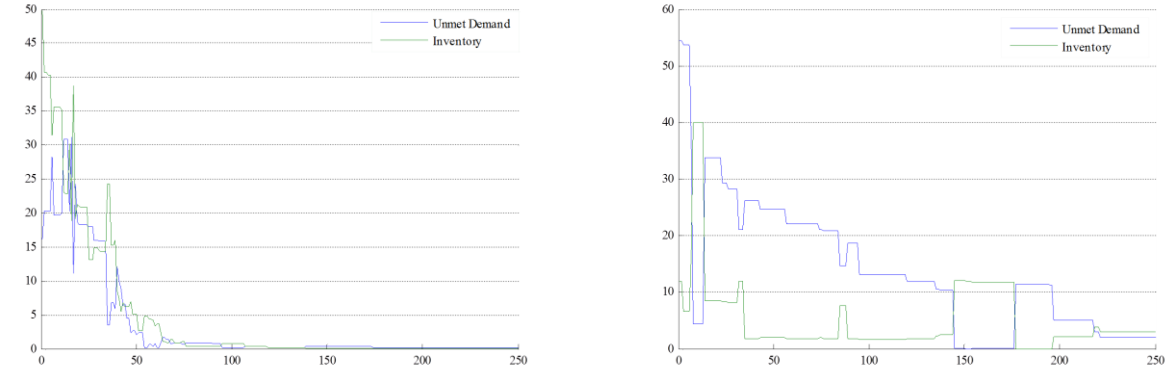
[19] J. Knowles and D. Corne, "The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation", Proceedings of the 1999 Congress on Evolutionary Computation, July 1999, IEEE, vol. 1, pp. 95-105, DOI: 10.1109/CEC.1999.781913

[20] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", IEEE

Transactions on Evolutionary Computation, vol. 3, number 4: November 1999, pp. 257–271, DOI: 10.1109/4235.797969

[21] K. Deb and S. Agrawal, "A Niched-Penalty Approach for Constraint Handling in Genetic Algorithms", Proceedings of the International Conference in Artificial Neural Nets and Genetic Algorithms, Portoroz, Slovenia, Springer Vienna, 1999, pp. 235-243. DOI: 10.1007/978-3-7091-6384-9_40

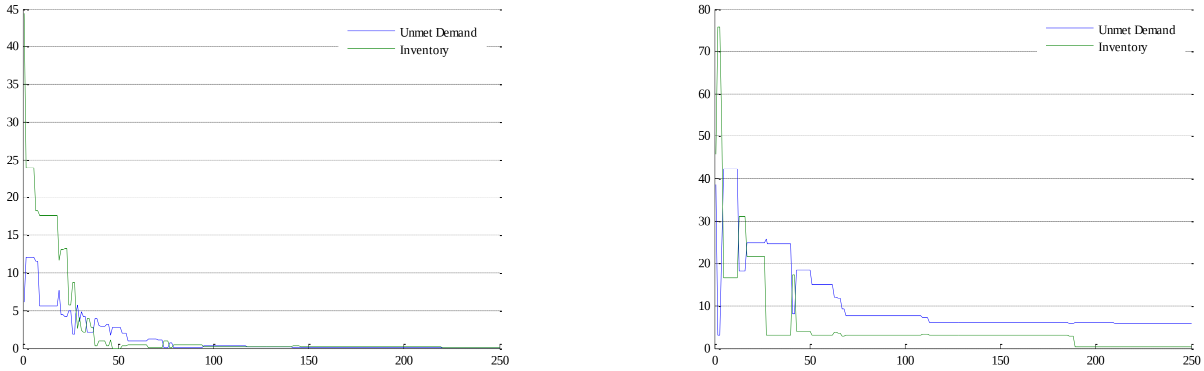
[22] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Germany: Springer-Verlag Berlin Heidelberg, 2003, ISBN 978-3-540-40184-1.



SBX-PBM

WAX-UM

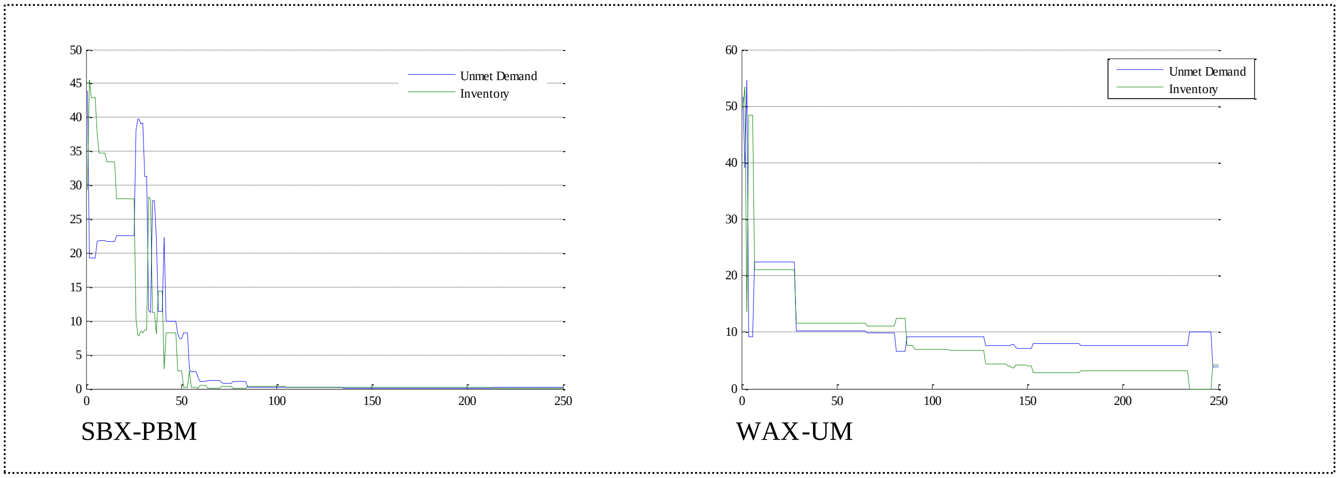
Figure 1. Simulation 1



SBX-PBM

WAX-UM

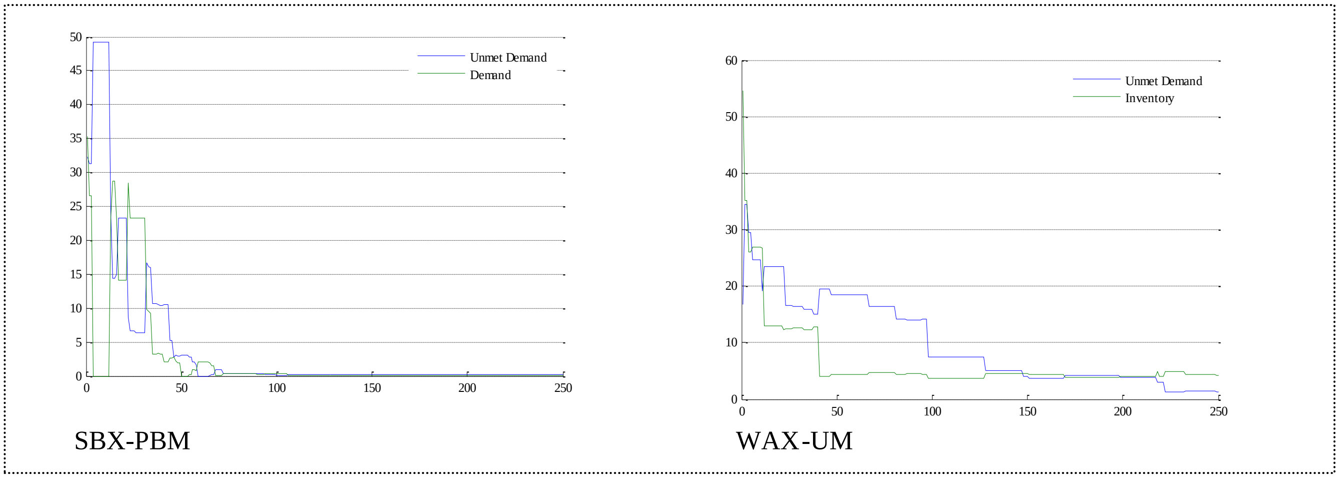
Figure 2. Simulation 2



SBX-PBM

WAX-UM

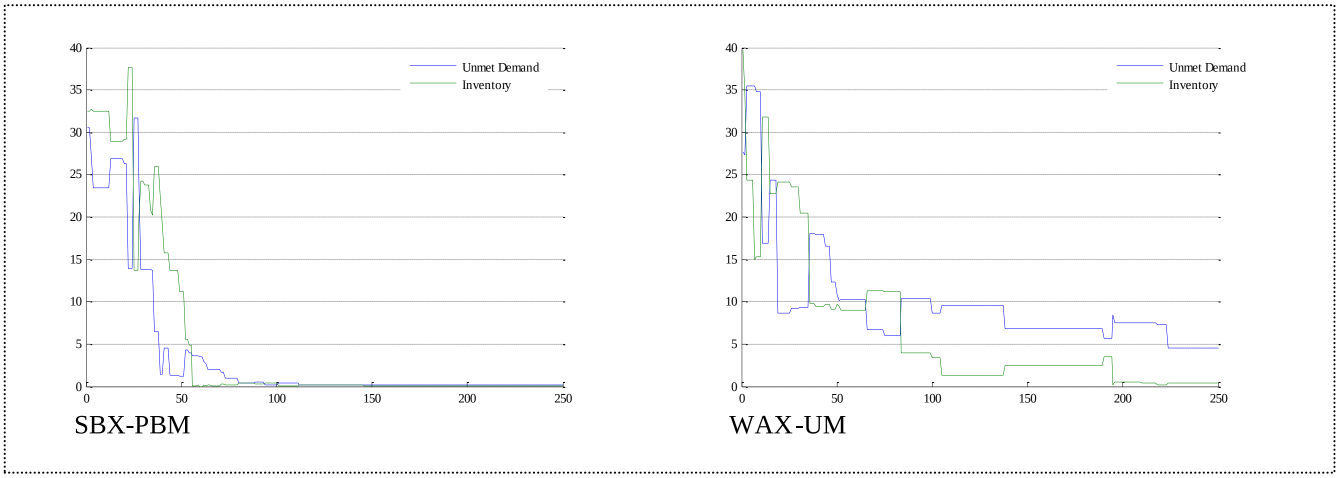
Figure 3. Simulation 3



SBX-PBM

WAX-UM

Figure 4. Simulation 4



SBX-PBM

WAX-UM

Figure 5. Simulation 5

A Generalized Agent-Based Model to Simulate Emergency Departments

Zhengchun Liu, Eduardo Cabrera, Dolores Rexachs and Emilio Luque

Computer Architecture and Operating Systems Department
Universitat Autònoma de Barcelona
Bellaterra, Barcelona, Spain
Email: {lzhengchun, ecabrera}@caos.uab.es, {dolores.rexachs, Emilio.Luque}@uab.es

***Abstract*—Computer simulation based methods have enjoyed widespread use in healthcare system investigation and improvement in recent years. Healthcare systems are based on human interactions and Emergency Departments (ED) are one of the key components of the healthcare system. The efficiency and quality of service in ED have a great influence on the whole healthcare system. The first step to intensively study the emergency department, to find its underlying problem or to provide the best service with limited budget, should be to create a realistic computational model of the ED. Agent-Based Modeling and Simulation (ABMS) is an excellent tool to deal with complex system like ED. This research introduces a generalized ABMS-based computational model of ED. The model has been implemented and verified in a Netlogo modeling environment and can be used to simulate different EDs through a tuning process.**

Keywords–Emergency Department; Healthcare; Agent-Based Modeling and Simulating; Complex system.

I. INTRODUCTION

The Emergency Department (ED), a medical treatment facility specializing in acute care of patients who arrive without prior appointment, needs to operate 24 hours per day, 365 days per year. Hundreds of people attend ED per day looking for healthcare services. It is an important entry point to access the healthcare service system. Patients frequently arrive with unstable conditions. Some of them arrive unconscious, and their information such as their medical history, allergies, and blood type may be unavailable. Thus, they should be treated quickly.

In order to set which patient should be visited first, it is mandatory to classify them. Triage is the process of determining the priority of patients' treatments based on the severity of their condition. This can efficiently improve patients' treatment process when resources are insufficient for everyone to be treated immediately [1]. The Spanish scale of triage is very similar to the worldwide Canadian one; it consists of 5 levels, with 1 being the most critical (resuscitation), and 5 being the least critical (non-urgent). The triage process also determines the order and priority with which the patient must be attended and the treatment area where they will be treated. This research has been performed with the participation of the ED Staff of the Hospital of Sabadell (a University tertiary level hospital in Barcelona, Spain that provides care service to a catchment area of 500,000 people, and attends 160,000 patients per year in the ED). The model and the simulator will be verified and validated with the data taken from this Hospital.

In general, there are two separate treatment areas (labeled as A and B in this study) in some big EDs to provide diagnosis and treatment service after the triage process. Area A is for those patients with acuity levels 1, 2 and 3 whereas area B is for patients with acuity levels 4 and 5. Area A is occupied by the most urgent patients and is made up of careboxes. A carebox is a small room which contains essential medical equipment and supplies that could be used for patients' treatment in ED. Patients attended in area A will stay in their own carebox during all the diagnosis and treatment process. Area B is for patients with an acuity level of 4 and 5, which for the Hospital of Sabadell represents 60% of the patients attended in the ED. In area B, there are 3 or 4 attention boxes in which doctors and nurses interact with patients, and a large waiting room in which all patients will remain while not having interaction with the ED Staff. Area A occupies more space than area B.

As for the category of the research object, although the term complexity may have different definitions, according to the definition by Tan et al. [2], a complex system consists of interconnected components that work together, interchange resources and information with the environment in order to meet an objective. This kind of system exhibits several major characteristics: a large number of interactive parts; interactive complexity and self-organization. Thus, there is no doubt that the Emergency Department is a specific case of a complex system. There are no standard models to describe such systems, and analytical models cannot easily represent the complex system caused by random events. With the development of high performance computing techniques, computer based simulation could be one of the best solution to study this kind of system.

The purpose of this work is to develop a general model and simulator that could be used to simulate any emergency department by using an Agent-Based Modeling and Simulation (ABMS) approach. The final objective is to develop a simulator that, used as a decision support system, aids the managers of the EDs, to analyze risks, facilitate coordination implementation, allocate the resources and identify weaknesses in service of resource. In addition, it could be used for studying other related problems in the healthcare service system and as a sensor of ED to generate data concerning different simulation scenarios for finding some unusual knowledge of the healthcare service by using big-data and data mining techniques. These are three ongoing research lines of the High Performance Computing for Efficient Application and Simulation (HPC4EAS) research group at Universitat Autònoma de Barcelona research group based on the ED simulator. Our previous studies have created the simulator of area B for patients with acuity level 4 and 5 [3][4]. This research is continuing with the previous work to create the model of area A in the ED.

The rest of this article is organized as follows. Section II gives the literature review, a brief introduction to our previous work and the main improvements of this study. Section III is the main part of this article, which has four parts: Section III.A describes the modeling approach for this kind of complex system; Section III.B gives the structure of the model, the definition of the agents and agent behaviors; Section III.C explains the way to model the interactions between agents, and section III.D presents the mathematical-computational model of the diagnosis and treatment phase in detail. Finally, Section IV closes the article with conclusion and future work.

II. RELATED WORK

Rising et al. [5] are among one of the earliest publications on using computer modeling and simulation for improving healthcare service. The authors use the Monte Carlo simulation model for analyzing the effects of alternative decision rules for scheduling appointment periods during the day to increase patient throughput and physician utilization. Hancock et al. [6] developed a computer-based simulator of the hospital systems, which is used for predicting the size of nursing staff configurations under different scenarios.

Concerning the development of the computational model of ED, Paulussen et al. [7] describe a multi-agent based approach for patient scheduling in hospitals. In such a system, patients and hospital resources are implemented as autonomous agents in which the resource agents view the patients as entities to be treated, and the patient agents view the medical actions as tasks that need to be performed. The coordination of patients is achieved through a market mechanism. Patient agents negotiate with each other over scarce hospital resources, using state health dependent cost functions to compute bid and ask prices for time slots. Within this concept, stochastic processing times and variable pathways are considered. Unfortunately, the system does not take into consideration patient variety or the different kinds of healthcare staff. But in fact, the variety of patients and staff has great influence on the performance of ED.

As the use of simulation approach for studying EDs, Badri and Hollingsworth [8] developed an Emergency Room (ER) simulation model incorporating the major activities. The model allows the evaluation of “what if?” questions through changing the values of the variables and simulating the results. The ER simulation model determines the effects of changes in the scheduling practices, allocation of scarce resources, patient demand patterns, and priority rules for serving patients. In the study of Gove and Hewett [9], they examined the problem of capacity in hospitals and proved that: due to the complexity of the hospital and its departments, simulation was an ideal choice to study. Moreover, Diefenbach et al. [10] found that varying the number of beds, physical layouts, access to radiology and pathology services etc. in the ED has an exponential effect on expression of the system. The simulation results under the change of the configurations can provide valuable reference for management decision making. Kuljis et al. [11] compared the healthcare system with business and manufacturing, and provided the feasibility of using modeling and simulation methods to improve the Quality of Service (QoS) in healthcare system.

For the use of ABMS approach for simulating EDs, Macal et al. [12] gave a tutorial to create an agent-based model for the complex system, and they suggested that ABMS promises to have far-reaching effects in the future on how to use computers to support decision making. As for the reason for chosing ABMS approach for simulating ED, Escudero-Marin et al. [13] gave the reason that ABMS is better for modeling the EDs than others. The authors also provided a general description of the possible potential use of ABMS in healthcare application.

The previous studies in our HPC4EAS research group mainly included creating the simulator of area B in ED [3], balancing between the budget and QoS, finding the optimal and sub-optimal resource configurations of ED to achieve better QoS with limited budget by using K-means methods and pipeline scheme [4][14].

Unlike area B, area A is the area dedicated to the critical patient. It is more complex and quite different with area B mainly because the patients in this area usually cannot move by themselves; consequently, the doctor and other auxiliary staff need attend these patients in their carebox. These cases lead to a greater amount of restrictions and interactions between the agents in ED. Compared with our previous model, the main improvements and contribution of this article include: considering some more agents, modifying the behavior of some exist agents, introducing a new way to define and simulate the interactions between agents and state transition of the agents, and providing an easy-tuning model to simulate the diagnosis and treatment phase.

The model created in this study is a generalized model. A tuning process is necessary before simulation. In this study, the tuning process is a process to adapt the generalized model to the real ED to be simulated. It is done by using the historical data of the given ED. The difference between the simulation results and the real data will correct the value of the model's internal parameters. After a series of feedback and iterations, when the difference reaches to an acceptable value, the model is adapted to the real ED.

III. SIMULATION MODEL

Conducting a valid simulation is both an art and a science. One of the main challenges when developing a general simulation model is to keep a model as simple as possible whilst including all the key system information to achieve the objectives of the simulation. One feasible way to do this is through the following three steps: (1) survey multitude real models; (2) analyze the concept structures of these real models; and (3) abstract and generalize from these real models to develop a reusable generic pattern model. This section detailed the general model of EDs.

A. Modeling approach

When faced this kind of complex system, it is almost impossible to model all its functionality directly because there are large numbers of factors that can affect the result and need to consider. A good way to model is by using a bottom-up-modeling approach. Starting from the bottom subsystem (agents, agents' behavior and interactions between agents), the execution of the simulator will cause a large amount of interactions between these agents, and then these interactions will emerge the functionality of the emergency department indirectly.

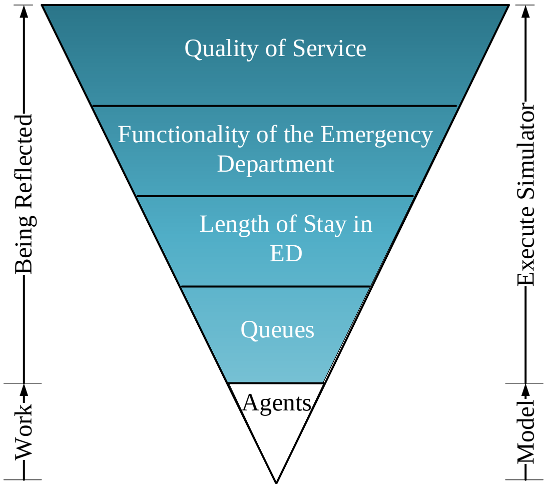


Figure 1. Bottom up modeling approach.

As shown in Figure 1, it works by modeling the agents, their behaviors and interactions between them. Then, when executing the simulator, the state of agents will be changed by their interactions, and the queue for the interactions and their Length of Waiting Time (LoWT) will emerge, by such analogy, the functionality of the ED will emerge indirectly through the execution of the simulator. Furthermore, the QoS can be evaluated through the results of different simulation scenarios.

Discrete Event Simulation (DES), System Dynamics (SD) and ABMS are the three main approaches used when simulating this kind of complex system. There is a large body of literature describing the use of DES models in ED studies, whilst there is considerably less literature on the use of ABMS for this purpose. As healthcare systems are based on human actions and interactions, combined with our experience and requirement, it can be more properly to model with ABMS [13]. ABMS models can offer ways to provide a deep insight view and to generate hypotheses about system behavior by representing this as a result of the interaction between the agents.

B. Agents and Agent Behaviors

In agent-based terms, a system is modeled as a set of heterogeneous agents that will create the overall behavior of the system through their interactions in the execution process of model. When developing ABMS, it is crucial to represent the two main parts of an agent-based model: agents and agent behaviors.

The definition of agents should include their capabilities, the actions they can perform and the characteristics of the environment that surrounds them. In this article, each kind of agents is defined as:

$$A = \{V \cup B\} \tag{1}$$

where V is a set of state variable to represent the agents' characteristics:

$$V = \{V_1, V_2, \ldots, V_m\} \tag{2}$$

The states of the agents are indicated by the values of the state variables. And set B contains all the behaviors of the agents in this category:

$$B = \{B_1, B_2, \ldots, B_n\} \tag{3}$$

Each kind of agent has its own definition of state variable V and each state variable has a set of possible values in its range:

$$V_i = \{Y_1, Y_2, Y_3, \ldots, Y_{K_i}\}(0 \le i \le m) \tag{4}$$

After some survey of several EDs with the participation of sanitary staff from hospital of Sabadell. The agents considered in ED and their behaviors are shown in Table I.

TABLE I. AGENTS AND AGENTS' BEHAVIOR

Agent	**Behavior**
Patient	Waiting for service. Accepting service. Waiting for treatment takes effect.
Admission Staff	Provide admission service for patient. Waiting for next patient.
Triage Nurse	Provide triage service for patient. Waiting for next patient.
Doctor	Look over test result. Provide diagnostic service. Arrange test for patient. Arrange treatment plan. Waiting for task.
Auxiliary Staff	Moving patient to the specific place. Waiting for task.
Nurse	Take and send samples for laboratory test. Provide treatment service. Waiting for task.
Laboratory Test	Accept sample from nurse. Analyze samples of patient. Send analyzing result to the corresponding doctor. Waiting for samples.
Internal Test	Provide test service. Send analyzing result to the corresponding doctor. Waiting for next patient.
External Test	Provide test service. Send analyzing result to the corresponding doctor.
Ambulance	Providing service to patients. Waiting for task.
Carebox	Providing treatment place to patient. Waiting for next patient.

In Table I, each kind of agent has its own behavior. The behaviors are generalized that do not represent one specific action, instead, the combination of the value of their state variables and the generalized behavior will represent the real action. For example, if the value of state variables indicates that the patient stays in the waiting room, waiting for service after admission, which means that they are waiting for triage service instead of other services.

As to the diagnosis service, there are different kinds of tests, for example blood test, x-ray, B ultrasonic, and so on. In reality, most of the time spent on diagnosis was on waiting,

waiting for the service and waiting for the result. The LoWT depends on the length of the waiting queue which emerged as a result of the agents' interactions. The length of the waiting queue depends on the number of patients, their acuity level distribution and the amount of available resources in ED. That is the reason why the LoWT emerged by executing the simulation instead of being modeled directly. From this point of view, it is better to take the test service as agents. In this manner, the patients need to interact with these agents to know their body condition, because of the limited number of these agents, the patient usually needs to wait for the interaction, as shown in Figure 1, the queues will emerge in this way. Therefore, in Table I, all these tests were classified into three types according to their interactive mode:

Laboratory test: It is a kind of test performed in a carebox and laboratory, when the doctor ordered this kind of test, the nurse will go to the carebox to take some samples and then send them to the laboratory (or analysis by nurse directly), after a period of time, the doctor will get the result. Due to the test being an agent, the LoWT could emerge from simulation. The main characteristic of this kind of test is that the patients do not need to move, hence there is no interaction with the auxiliary staff, but instead, the nurse will have some interactions to perform with the patient and laboratory.

Internal test: The internal test means the diagnostic equipment is owned and used only by ED, thus the length of the waiting queue for the service is under-control and can emerge through simulation. Unlike a laboratory test, in order to do this kind of test, the patients need to move, so they need to interact with the auxiliary staff, which may cause longer LoWT if the configuration of auxiliary staff is inadequate.

External test: In ED, there are some types of diagnostic equipment shared with the hospital wards or even shared with other EDs, therefore the length of the waiting queue cannot emerge because part of the agents outside ED who need to interact with these test agents but do not appear in the simulator. One way to simulate is by using a period of time delay (based on statistical data and following probability distribution obtained from tuning process) to model this kind of test. As with an Internal test, in order to do this kind of test, the patients need to interact with auxiliary staff to move them to the corresponding test room.

In addition to this, the ambulance and hospital wards were also considered because the behavior of these two agents also have obvious effects on the functionality of ED:

Ambulance: Some patients come by ambulance, especially the patients in area A. Part of them will do admission and triage in the ambulance, when the patient is critical enough, and on arrival they will go to a carebox directly or stay in the ambulance until a free carebox is available. At the same time, some patients may need to go home by ambulance. But usually the ambulance has arrival delay. Under this circumstance, they will keep using the carebox. Hence, the quality of ambulance service is one of the factors that may cause overcrowding in ED.

Hospital ward: A hospital ward is a main exit way for the patients in area A. It is common that the hospital ward does not have enough free beds, thus the patient will keep using the carebox even though this is not necessary, so the throughput of hospital wards also has direct influence on the performance of ED. For simulating the hospital ward, it is similar to the external test agent. The number of free beds and available time will be simulated through the probability distribution and the parameters of the distributions are obtained from analyzing the real data in the tuning process.

C. Model of the Interactions

The functional behavior of any system can be specified by a state machine (also called an object) [15] . In this research, to model the interactions between the agents and their states, the Finite State Machine (FSM) was used.

According to the definition of agents through (1)-(4), the state of agents are presented by the value of their state variables and each state variable has a set of possible values. Based on the actual situation, the transition of agents' state is caused by interaction with other agents or in some cases with time elapse. Thus, the value of the state variables are changed by one of their behaviors or time elapsing, as in (5):

$$Y_{K_i} = f(B_j, T)(0 \leq i \leq m, 0 \leq j \leq n) \tag{5}$$

where B_j represents the corresponding behavior with other agents, it is an element of the behavior set B. T represents the elapsing of the time because sometimes the state of the agents, e.g., patients' body condition after medicating, can change with time goes on without any interactions.

As shown in Figure 2, the state machine accepts commands and produces outputs, which means that when the agents interact with other agents and/or with the time elapsing (accept input), the value of one or several variables will be changed (because of the outputs produced). Any one of the variables' value changing will represent the state transition.

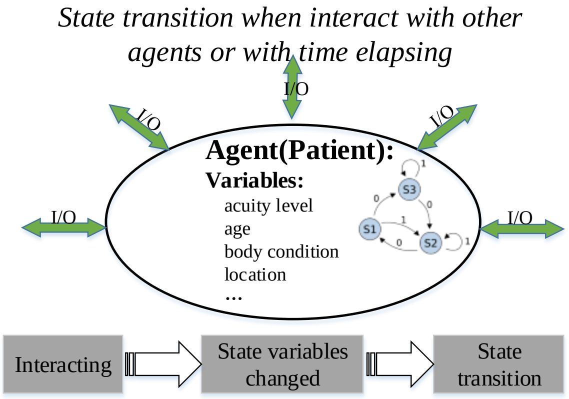


Figure 2. A typical patient's conceptual state transfer model.

Therefore, the set of one kind of agents' states is the cartesian produce of each state variable's possible value set(see (6)). The state set of a specific agent in this type is a subset of S, which is determined by the specific configuration of the agent.

$$S = \{S_0, S_1, S_2, \ldots, S_t\} = \{V_1 \times V_2 \times V_3 \times \ldots \times V_n\} \quad (0 \leq t \leq \prod_{i=1}^{n} K_i) \tag{6}$$

ISBN: 978-1-61208-371-1

TABLE II. A PART OF A PATIENT'S STATE TRANSITION.

State index	Source State	Destination state	Input
...	...	...	...
S_t	Waiting for service (free carebox).	Waiting for service (Doctor's diagnosis).	Notice from IS with a free care box.
S_{t+1}	Waiting for service (doctor's diagnosis)	Accepting Service(meet with doctor)	Doctor arrive at patient's carebox.
S_{t+2}	Accepting Service(meet with doctor)	Waiting for service (X-Ray test service)	Doctor order X-Ray test for patient.
S_{t+3}	Waiting for service (X-Ray test service)	Accepting Service(X-Ray test service)	X-Ray service available.
S_{t+4}	Accepting Service(X-Ray test service)	Waiting for service (Doctor's review of the test result)	X-Ray service finished.
...	...	...	...

By combining with Table I about the generalized behavior of agents, (1) - (6) for the definition of agents and Figure 2 about the conceptual state transfer, it is feasible to list all the agents' evolution during the stay in ED, Table II gives a part of one patients' state transition. In Table II, although some of the state is the same as *Waiting for service*, the value of its state variables will determine the specific service the patient waiting for.

Above all, by means of defining the agents through state variables, it will be feasible to deal with the huge amount of states of the agents. At the same time, it will be easy to add/remove states simply by adding/removing elements in the set of possible values of the state variables and their corresponding behavior. For the study of other ED related problems, for example, the virus propagation in ED, some new state variables and their possible values will become easy to be added to indicate some more states. With the same approach, by the execution of the model, some new functionality of the research object will emerge from these new states.

D. Diagnosis and Treatment Phase

Before the diagnosis and treatment phase, for the patients, they need to do admission and triage, actually these two phases take very little time in reality. Figure 3 indicates the common process in the emergency department. For the patients, especially the patients with acuity levels 1, 2 and 3, most of their Length of Stay (LoS) in the emergency department is spent in the carebox for having various kinds of tests, receiving treatment and waiting for the treatment to take effect. This is the most important part of the model because most of the state transitions take place during this phase.

For the patients in area A (with acuity levels 1, 2 and 3), they always stay in their own carebox during the diagnosis and treatment phase. The general process in the carebox is that the patient takes some tests (x-ray, ultrasound, blood test and so on), then the doctor reviews the test result and provides one treatment plan or asks to do further tests. After that, the nurse will carry out the treatment plan or take some test samples if a laboratory test is ordered. After a period of time, some state variables of the patient will be changed or the patient unfortunately die. With the change of the patients' body condition, the doctor will decide what the patient need to do next: to go the hospital ward, go home or continue with diagnosis and treatment. In order to generalize the process of all the patients, the next state will be decided by probability distribution during simulating. The distribution model of the probability was based on the statistical data from the real EDs. Figure 3 indicates the general process-transfer strategy during the patients' stay in EDs.

In Figure 3, $P_1(\%)$, $P_2(\%)$, $P_3(\%)$ and $P_4(\%)$ represent the probability of the next state transition separately. $P_1'(\%)$ and $P_2'(\%)$ represent the decision of the doctor after reviewing the test results and body condition of the patients in probability. All of the probabilities follow some probability distributions. The probability density function of the distribution is decided by several key parameters based on the statistical analysis of doctors' decision and patients' behavior, the value of these parameters are estimated by a tuning process from real historical data of the specified ED. The uniform forms of the density functions are:

$$P_i = f(LoS, age, level) \tag{7}$$

$$\sum_{i=1}^{4} P_i = \quad 100\% \tag{8}$$

$$P' = f'(ToT, age, level) \tag{9}$$

$$\sum_{i=1}^{2} P'_i = \quad 100\% \tag{10}$$

where LoS is the patient's length of stay in the carebox. age is the age of the patient, which also has big influence to the probability of state transition. $level$ is the acuity level of the patient. And ToT is the type of test service or diagnosis by doctor.

The function f and f' are the probability density function. These functions will be implemented by analyzing real historical data in tuning process. This work can only be done in the tuning process because different EDs have different characteristics, it is a part of the simulator instead of the general model. Therefore, combined with (1) - (10), every patient will show different behavior during the execution of the model because of the probability distribution and their own differences in body condition. But the statistical property of agents will reflect their common behavior.

IV. CONCLUSION AND FUTURE WORK

Simulation methods have long been used to model elements of healthcare systems with a view to analyzing new system designs, retrofitting to existing systems and proposing changes to operating rules. The Emergency Department (ED) is a typical complex system. To perform intensively study, a realistic computational model is compulsory. An approach to modeling this kind of system is by using agent-based modeling and simulation, which is a kind of bottom-up modeling approach. This paper presents a generalized agent-based model of the emergency departments. It was designed based on the survey of different EDs and with the participation of sanitary staff in ED. This model has been implemented and verified in a Netlogo modeling environment. It is not dedicated with one

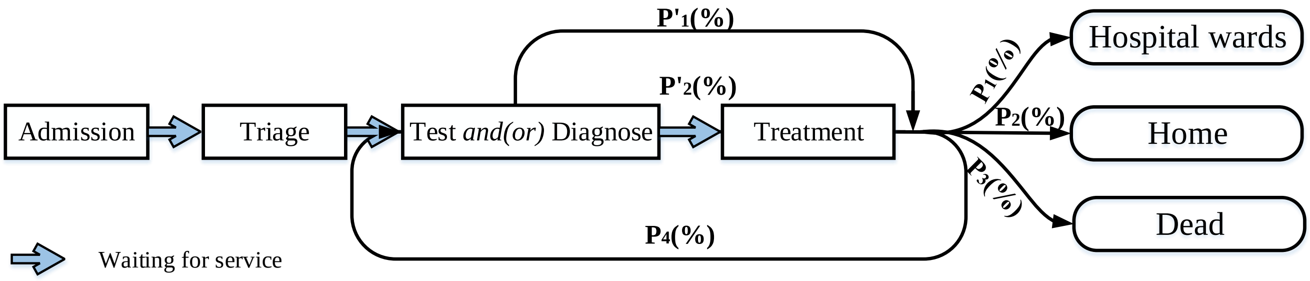


Figure 3. Main process in Emergency Departments.

specific ED, which can be used to simulate different EDs through tuning process.

This research is a progress of our previous work. To model the critical part of the ED (here, we call it area A), we defined some new agents also added some new state variables to extend the behavior of the previous agents. A new way to simulate the interaction between the agents and the state transition of the agent was provided, an easy tuning model was created for diagnosis and treatment phase in area A. In reality, most of the emergency department works like area A. For those big EDs, they have both area A and B, hence with both the model of A and B, we have the model of the whole EDs.

Creating the computational model of the object is the first step of simulation. Model verification is the task to ensure that the model behaves as intended, some basic experiments has been done to verify the functionality of the model. But, in order to validate the simulator, tuning for some real EDs is mandatory. Therefore, the first step of future work should be validation. Some real historical data of EDs will be asked to perform the tuning process. Moreover, during the tuning process, due to the great number of parameters for the model, and the large number of agents and interactions between them. To increase the number of studied scenarios and reduce execution time as well, the use of high performance computing will be mandatory.

In addition, the ED is the main entrance to the healthcare service; some problems of the healthcare service system are caused by the performance of ED. However, the ED is not independent, all the departments of healthcare system influence each another. Thus, our future work also include creating the simulator of other healthcare departments, for example the hospital wards to close the simulation loop of the whole healthcare service system. After that, the simulators of these departments will work as the sensor of the healthcare service system. The data generated from these sensors will be analyzed through data-mining and big-data techniques to find some unusual knowledge of the system to provide smarter service to patients.

ACKNOWLEDGEMENTS

This research has been supported by the MINECO (MICINN) Spain under contract TIN2011-24384 and partially supported by China Scholarship Council (CSC) under reference number: 201306290023.

REFERENCES

[1] R. S. Bermejo, C. C. Fadrique, B. R. Fraile, E. F. Centeno, S. P. Cueva, and E. De las Heras Castro, “El triaje en urgencias en los hospitales españoles,” Emergencias: Revista de la Sociedad Española de Medicina de Urgencias y Emergencias, vol. 25, no. 1, 2013, pp. 66–70.

[2] J. Tan, H. J. Wen, and N. Awad, “Health care and services delivery systems as complex adaptive systems,” Communications of the ACM, vol. 48, no. 5, 2005, pp. 36–44.

[3] M. Taboada, E. Cabrera, M. L. Iglesias, F. Epelde, and E. Luque, “An agent-based decision support system for hospitals emergency departments,” Procedia Computer Science, vol. 4, 2011, pp. 1870–1879.

[4] M. Taboada, E. Cabrera, F. Epelde, M. L. Iglesias, and E. Luque, “Using an agent-based simulation for predicting the effects of patients derivation policies in emergency departments,” Procedia Computer Science, vol. 18, 2013, pp. 641–650.

[5] E. J. Rising, R. Baron, and B. Averill, “A systems analysis of a university-health-service outpatient clinic,” Operations Research, vol. 21, no. 5, 1973, pp. 1030–1047.

[6] W. M. Hancock and P. F. Walter, “The use of computer simulation to develop hospital systems,” ACM SIGSIM Simulation Digest, vol. 10, no. 4, 1979, pp. 28–32.

[7] T. O. Paulussen, A. Zöller, A. Heinzl, L. Braubach, A. Pokahr, and W. Lamersdorf, “Patient scheduling under uncertainty,” in Proceedings of the 2004 ACM symposium on Applied computing. ACM, 2004, pp. 309–310.

[8] M. A. Badri and J. Hollingsworth, “A simulation model for scheduling in the emergency room,” International Journal of Operations & Production Management, vol. 13, no. 3, 1993, pp. 13–24.

[9] D. Gove and D. Hewett, “A hospital capacity planning model,” OR Insight, vol. 8, no. 2, 1995, pp. 12–15.

[10] M. Diefenbach and E. Kozan, “Hospital emergency department simulation for resource analysis,” Industrial Engineering & Management Systems, vol. 7, no. 2, 2008, pp. 133–142.

[11] J. Kuljis, R. J. Paul, and L. K. Stergioulas, “Can health care benefit from modeling and simulation methods in the same way as business and manufacturing has?” in Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come. IEEE Press, 2007, pp. 1449–1453.

[12] C. M. Macal and M. J. North, “Tutorial on agent-based modeling and simulation,” in Proceedings of the 37th conference on Winter simulation. Winter Simulation Conference, 2005, pp. 2–15.

[13] P. Escudero-Marin and M. Pidd, “Using abms to simulate emergency departments,” in Proceedings of the Winter Simulation Conference. Winter Simulation Conference, 2011, pp. 1239–1250.

[14] E. Cabrera, M. Taboada, M. L. Iglesias, F. Epelde, and E. Luque, “Simulation optimization for healthcare emergency departments,” Procedia Computer Science, vol. 9, 2012, pp. 1464–1473.

[15] L. Lamport, “Time, clocks, and the ordering of events in a distributed system,” Communications of the ACM, vol. 21, no. 7, 1978, pp. 558–565.

IoT Component Design and Implementation using Discrete Event Specification Simulations

Souhila Sehili
University of Corsica
SPE UMR CNRS 6134
Corte, France
Sehili@univ-corse.fr

Laurent Capocchi
University of Corsica
SPE UMR CNRS 6134
Corte, France
capocchi@univ-corse.fr

Jean-François Santucci
University of Corsica
SPE UMR CNRS 6134
Corte, France
santucci@univ-corse.fr

***Abstract*— The Internet of Things (IoT) approach enables rapid innovation in the area of internet connected devices and associated cloud services. An IoT node can be defined as a flexible platform for interacting with real world objects and making data about those objects accessible through the internet. Communication between nodes is discrete event-oriented and the simulation process play an important role in defining assembly of nodes. In this paper, we propose the definition of a modeling and simulation scheme based on a discrete-event formalism in order to specify at the very early phase of the design of an ambient system: (i) the behavior of the components involved in the ambient system to be implemented; (ii) the possibility to define a set of strategies which can be implemented in the execution machine. The DEVSimPy environment is then used to implement the example of a switchable on/off lamp.**

Keywords-DEVS; IoT; formalism; assembly; strategies.

I. Introduction

Technological advances in recent years around mobile communication and miniaturization of computer hardware have led to the emergence of ubiquitous computing. Computing tools are embedded in objects of everyday life. The user has at its disposal a range of small computing devices, such as Smartphone or PDA (Personal Digital Assistant), and their use is part of ordinary daily life. The definition of such complex systems involving sensors, smartphone, interconnected objects, computers, etc., results in what is called ambient systems. One of todays challenges in the framework of ubiquitous computing concerns the design ambient complex systems. One of the main problems is to propose a management adapted to the composition of applications in ubiquitous computing. The difficulty is to propose a compositional adaptation which aims to integrate new features that were not foreseen in the design, remove or exchange entities that are no longer available in a given context. Mechanisms to address this concern must then be proposed by middleware for ubiquitous computing. Several kinds of middleware tools have been proposed in the recent years [2]. We have being focused on the WComp environment. WComp is a prototyping and dynamic execution environment for Ambient Intelligence applications. WComp [2] is created by the Rainbow research team of the I3S laboratory, hosted by University of Nice - Sophia Antipolis and CNRS. It uses lightweight components to manage dynamic orchestrations of Web service for device, like UPnP (Universal Plug and Play), discovered in the software infrastructure. In the framework of the WComp, it has been defined a management mechanism allowing extensible interference between devices. In order to deal with the asynchronous nature of the real world, WComp has defined an execution machine for complex connections.

In this paper, we propose the definition a modeling and simulation scheme based on the DEVS formalism in order to specify at the very early phase of the design of an ambient system: (i) the behavior of the components involved in the ambient system to be implemented; (ii) the possibility to define a set of strategies which can be implemented in the execution machine. The interest of such an approach is twofold: (i) the behavior will be used to write the methods required in order to code the components using WComp environment; (ii) to check the different strategies (to be implemented in the execution machine) before implementation.

The rest of the paper is as follows: Section II concerns the background of the study by presenting the traditional approach for the design IoT systems. It briefly introduces a set of middleware framework before focusing on the WComp Framework. The DEVS formalism and the DEVSimPy environment are also presented. In Section III, the proposed approach based on the DEVS formalism is given. An overview of the approach as well as the interest in using DEVS simulation is detailed. Section IV deals with the validation of the approach through a case study The conclusion and future work are given in Section V.

II. Background

A. IoT Design and WComp.

The ubiquitous computing is a new form of computing that has inspired many works in various fields such as the embedded system, wireless communication, etc. Embedded systems offer computerized systems having sizes smaller and smaller and integrated into objects everyday life. An ambient system is a set of physical devices that interact with each other (e.g., a temperature sensor, a connecting lamp, etc.). The design of an ambient system should be based on a software infrastructure and any application to be executed in such an ambient environment must respect the constraints imposed by this software infrastructure.

Devices and software entities provided by the manufacturers are not provided to be changed: they are black boxes. This concept can limit the interactions to use the services they provide and prevents direct access to their implementation. The creation of an ambient system can not under any circumstances pass by a modification of the internal behavior of these entities but simply facilitate the principle reusability, since an entity chooses for its functionality and not its implementation. In the vision of ubiquitous computing, users and devices operate in an environment variable and potentially unpredictable in which the entities involved and appear conveniently disappear (a consequence of mobility, disconnections, breakdowns, etc.). It is not possible to anticipate what the design will all devices that will be available and when. As a result a set of tools have been interested in developing software infrastructure allowing the design of applications with the constraint unpredictability availability of component entities [2].

In this paper, we deal with the WComp framework, which is used in order to design ambient systems. The WComp architecture is organized around containers and designers. The purpose of containers is to take over the management of the dynamic structure such that instantiation, destruction of components and connections. An application is created by a WComp component assembly in a container, according to LCA (Lightweight Component Architecture) [2]. WComp allows to implement an application from an orchestration of services available in the platform and/or other off-the-shelves components.

Whatever the tool which may be used, the design of a IoT component leans on the definition of:

- A set of methods allowing to describe the behavior of the component;
- The execution machine associated with the considered component.

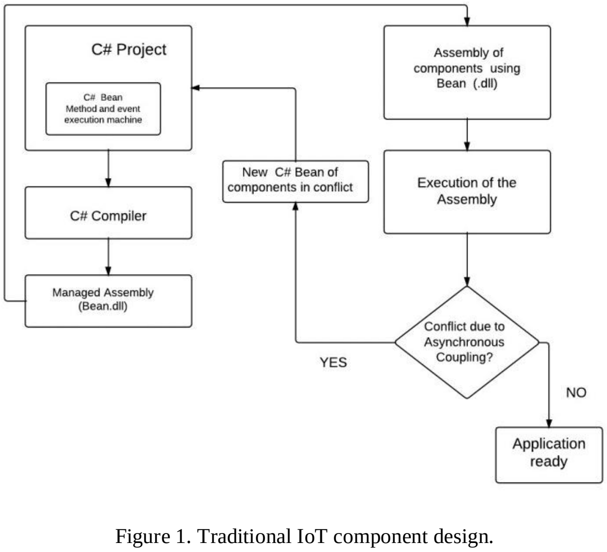


Figure 1. Traditional IoT component design.

The design of ambient computing systems involves a technique different from those used in conventional computing. Applications are designed dynamically by ”smart” devices (assembly components) of different nature. The construction of an ambient IoT system requires the definition Figure 1 of Methods and an execution engine.

The Designer runs the Container for instantiation and for the removal of components or connections between components in the Assembly which has to be created. A component belonging to the WComp platform is an instance of the Bean class implemented in the object language (C#). The description of a given execution machine has to be defined manually using methods for the management of events usually based on automata theory. Figure 1describes the traditional way to design an ambient system using WComp. The behavior and the components involved in the ambient system as well as the Bean classes describing the execution machine are coded using the C# language (C# rectangle in Figure 1).

The compilation allows to derive the corresponding binary files (dll) of the Bean classes involved in the resulting Assembly. The Assembly can then be executed. Conflicts are checked: if conflicts (generally due to asynchronous couplings) are detected the designer has to write a new behavior of the execution machine by recoding Bean classes in order to solve the coupling conflicts while if no conflict are detected the application is ready.

In this paper, we choose the proposed a new approach for a computer aided design of ambient systems using the DEVS formalism by developing DEVS simulation concepts and tools for the WComp platform. The goal is to use the DEVS formalism and the DEVSimPy framework in order to perform DEVS modeling and simulations: (i) to detect the potential conflicts without waiting to implementation and execution phases as in the traditional approach of Figure 1; (ii) to offer the designer to choose between different executions strategies and to test them using DEVs simulations; (iii) to propose a way to automatically generate the coded of the methods involved in the execution machine strategies. The DEVS formalism and the DEVSimPy environment are briefly introduced in the next two sub-sections while the proposed approach is introduced in Section III.

B. The DEVS formalism

Since the seventies, some formal works have been directed in order to develop the theoretical basements for the modeling and simulation of dynamical discrete event systems [8]. DEVS [9] has been introduced as an abstract formalism for the modeling of discrete event systems, and allows a complete independence from the simulator using the notion of abstract simulator.

DEVS defines two kinds of models: atomic models and coupled models. An atomic model is a basic model with specifications for the dynamics of the model. It describes the behavior of a component, which is indivisible, in a timed state transition level. Coupled models tell how to couple several component models together to form a new model. This kind of model can be employed as a component in a larger coupled model, thus giving rise to the construction of complex models in a hierarchical fashion. As in general systems theory, a DEVS model contains a set of states and transition functions that are triggered by the simulator.

A DEVS atomic model AM with the behavior is represented by the following structure:

$$AM =< X, Y, S, \delta int, \delta ext, \lambda, t_a > \qquad (1)$$

- X is the set of input values,
- Y is the set of output values,
- S is the set of sequential states,
- δint is the internal transition function dictating state transitions due to internal events,
- δext is the external transition function dictating state transitions due to external input events,
- λ is the output function generating external events at the output,
- t_a is the time-advance function which allows to associate a life time to a given state.

Connections between different atomic models can be performed by a coupled model. A coupled model, tells how to couple (connect) several component models together to form a new model. This latter model can itself be employed as a component in a larger coupled model, thus giving rise to hierarchical construction.

C. The DEVSimPy environment

DEVSimPy [1] is an open Source project (under GPL V.3 license) supported by the SPE team of the University of Corsica Pasquale Paoli. This aim is to provide a GUI for the modeling and simulation of PyDEVS [4] models. PyDEVS is an Application Programming Interface (API) allowing the implementation of the DEVS formalism in Python language. Python is known as an interpreted, very high-level, object-oriented programming language widely used to quickly implement algorithms without focusing on the code debugging [6]. The DEVSimPy environment has been developed in Python with the wxPython [7] graphical library without strong dependences other than the Scipy [3] and the Numpy [5] scientific python libraries. The basic idea behind DEVSimPy is to wrap the PyDEVS API with a GUI allowing significant simplification of handling PyDEVS models (like the coupling between models or their storage).

DEVSimPy capitalizes on the intrinsic qualities of DEVS formalism to simulate automatically the models. Simulation is carried out in pressing a simple button which invokes an error checker before the building of the simulation tree. The simulation algorithm can be selected among hierarchical simulator (default with the DEVS formalism) or direct coupling simulator (most efficient when the model is composed with DEVS coupled models).

III. Proposed approach

As pointed in sub-Section II-A, the traditional way to design ambient systems described in Figure 1 has the following drawback: the creation of Bean class components using the WComp platform is performed by the definition of methods (both implementing the behavior of a device and its execution machine) in the object oriented language C#. The compilation allows to obtain a set of library components which are used in a given Assembly (which corresponds to the designed ambient system). However, eventual conflicts due to the connections involved by the Assembly can be detected only after execution. This means that the Designer has to modify the execution machine of some components and restart the design at the beginning. We propose a quite different way to proceed, which is described in Figure 3. The idea is to use the DEVS formalism in order to help the Designer to:

- Validate different strategies for execution machines involved in an Assembly.
- Write the methods corresponding to the strategy of the execution machine he wants to implement.

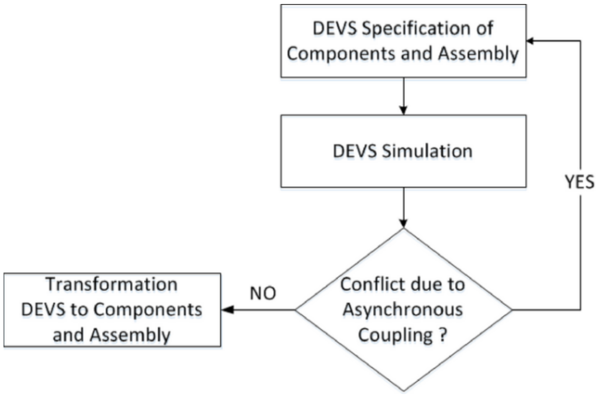

DEVS Specification of Components and Assembly
DEVS Simulation
YES
Conflict due to Asynchronous Coupling ?
NO
Transformation DEVS to Components and Assembly

Figure 2. IoT component design using DEVS.

For that, the Designer has first to write the specifications the components as well as the coupling involved in an Assembly (corresponding to an ambient system to implement) then simulations can be performed. According to the results of the simulation, conflicts can be highlighted: if some conflicts exists the DEVs specifications have to be modified if not the design process goes on with C# implementation, as in Figure 1. The DEVS specifications can be used to help the Designer to write the methods of the Bean classes in the C# language Figure 2 and then compile them and execute the resulting Assembly being assured that there will be no coupling conflict.

Section IV detailed the proposed approach using a pedagogical example. Two different execution machine strategies will be implemented using WComp and using the DEVS formalism. We will point out how DEVS can be used to simulate execution machines strategies before compilation and execution of the C# Bean classes. Furthermore, we also point out how the designer can use the DEVS specifications in order to write the methods involved in an execution machine strategy.

IV. CASE STUDY :SWITCHABLE ON/OFF LAMP

A. *Description*

We choose to validate the proposed approach on a pedagogical case study: realization of an application to control the lighting in a room. The case study involved three components to be assembled: a light component with an input (ON / OFF) and two switches components with an output (ON / OFF), as shown in Figure 3. Two different behaviors concerning the connections between the switch and the light component are envisioned (corresponding to the implementation of two different execution machines):

- First behavior: the light is controlled by toggles switches which rest in any of their positions.
- Second behavior: the light is controlled by pushbutton switches which have two-position devices actuated with a button that is pressed and released.

In this part, we will present first how we have implemented these two previous behaviors using the WComp platform. Then we will give the DEVS approach involving the DEVS specifications of the two behaviors of the case study and the way DEVS can be used for WComp design of the ambient components.

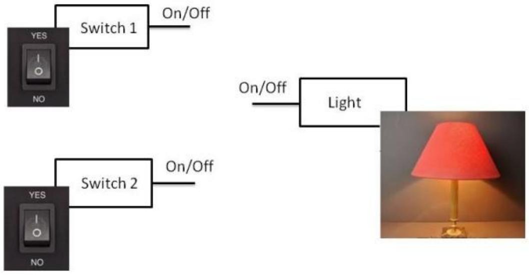


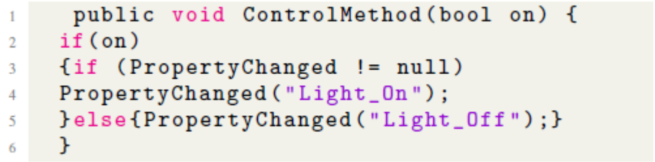
Figure 3. Assembly light and switches

B. *WComp implementation*

We implemented using the WComp platform the behavior corresponding to the toggle switch and the behavior of the push button switch. The two behaviors have been coded using two different Bean classes associated with the light component.

1) First behavior implementation.

The implementation corresponding to the toggle switch is described in the Figure 4. The line 2 is used to check the position of the toggle switch: if ON is true the line 3 ensures that there are subscribers before calling the event Property-Changed. In the lines 4 and 5, the event is raised and a resulting string is transmitted. The Bean class returns the String once the”ControlMethod” method is invoked.

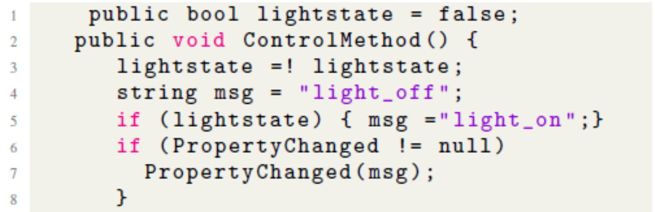


```
1   public void ControlMethod(bool on) {
2  if(on)
3  {if (PropertyChanged != null)
4  PropertyChanged("Light_On");
5  }else{PropertyChanged("Light_Off");}
6  }
```

Figure 4. First light method implementation in WComp.

2) Second behavior implementation.

The implementation corresponding to the pushbutton switch is described in the Figure 5. The initialization of the “lightstate” variable of component Light is performed through line 1. Line 3 allows to switch the value of the “lightstate” variable while line 4 allows to initialize the message to be returned. Line 5 is dedicated to check the “lightstate” variable and to eventually change to returned message. Lines 6 and 7 allow to ensure that there are subscribers before calling the event Property-Changed and transmit the returned message.



```
1    public bool lightstate = false;
2   public void ControlMethod() {
3      lightstate =! lightstate;
4      string msg = "light_off";
5      if (lightstate) { msg ="light_on";}
6      if (PropertyChanged != null)
7        PropertyChanged(msg);
8      }
```

Figure 5. Second light method implementation in WComp.

After the compilation of the two Bean classes, each bean class will be instantiated and connected with two checkbox representing the respective switches in order to realize the assembly in a WComp container, as shown in Figure 6.

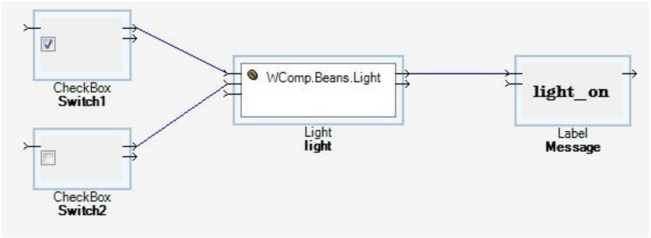


Figure 6. WComp assembly components.

C. *DEVS Specifications*

In order to highlight the interest of the DEVS formalism in the management of conflicts between WComp assembly components, we have written an atomic DEVS model for each kind of component ”light” and implement both of them using the DEVSimPy platform.

1) **First case:** DEVS implementation corresponding to the toggle switch behavior.

The DEVS specification of the corresponding WComp component of Section IV-B is achieved using the state automaton of Figure 7.

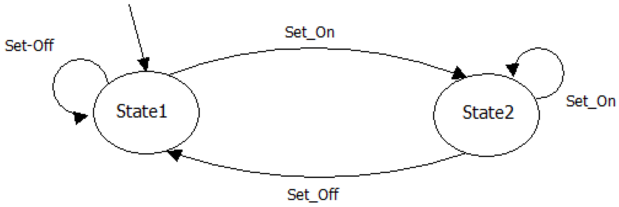
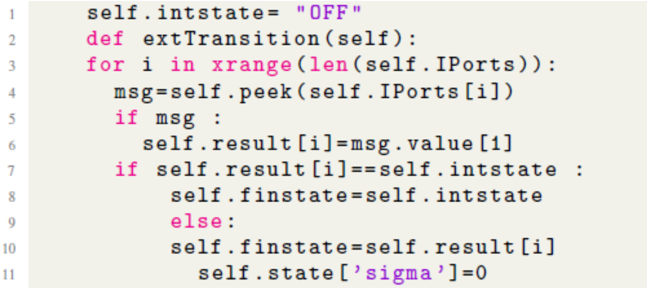


Figure 7. Automaton 1 of the light component.

The corresponding DEVSimPy implementation is given in Figure 8 (the behavior is expressed through the external transition of the Light component atomic model).



```
1   self.intstate= "OFF"
2   def extTransition(self):
3   for i in xrange(len(self.IPorts)):
4     msg=self.peek(self.IPorts[i])
5     if msg :
6       self.result[i]=msg.value[1]
7     if self.result[i]==self.intstate :
8         self.finstate=self.intstate
9         else:
10        self.finstate=self.result[i]
11          self.state['sigma']=0
```

Figure 8. External Transition of the light in DEVSimPy.

The initialization of the state variable “instate” is done in line 1 (initial value is OFF). Line 3 and line 4 allow to assign the variable msg with the value of the events on the input ports. From line 5 to line 10 the code allows to assign the value of the state variable “instate” according to the value of the variable “msg”: if the message on the port is equal to the initial state then the state variable remains on the same state else the value of the “instate” variable is changed. Line 11 by setting the variable sigma to 0 allows to activate the output function.

2) **Second case:** DEVS implementation corresponding to the pushbutton switch behavior.

The DEVS specification of the corresponding WComp component of Section IV-B is achieved using the state automaton of Figure 9.

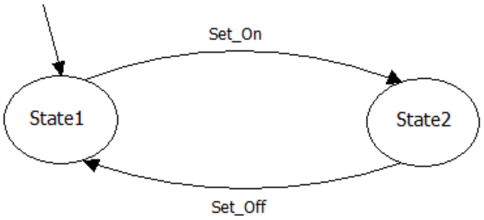
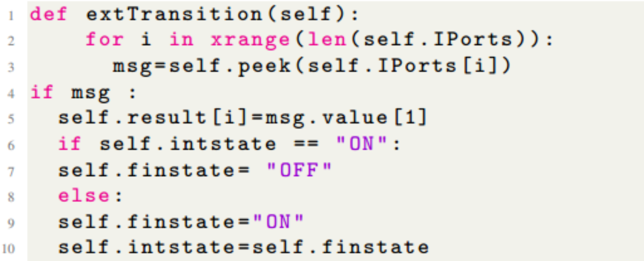


Figure 9. Automaton 2 of the light component.

The corresponding DEVSimPy implementation is given in Figure 10 (the behavior is expressed through the external transition of the Light component atomic model).



```
1  def extTransition(self):
2      for i in xrange(len(self.IPorts)):
3        msg=self.peek(self.IPorts[i])
4  if msg :
5    self.result[i]=msg.value[1]
6    if self.intstate == "ON":
7    self.finstate= "OFF"
8    else:
9    self.finstate="ON"
10   self.intstate=self.finstate
```

Figure 10. External Transition of the light in DEVSimPy.

Figure 10 gives the code of the external transition of the Light component atomic model. One can see from line 5 to 10 that in this second case the output message is switched from ON to OFF or OFF to ON according to the values of the input ports.

D. *Simulation results*

In both two cases, once the modeling scheme has be realized using the DEVSimPy environment we are able to perform simulations which correspond to the behavior of the ambient system under study according to the two different execution machines that have been defined.

The simulation results of the first case express the fact that the execution machine allows the ambient system under study remains in the initial position (ON or OFF) until we will actuate another position using one of the switches.

The simulation results of the second case express the fact that the execution machine allows the ambient system under study to alternately “ON” and “OFF “with every push of one of the switches.

E. *Interest of the presented approach*

As described in Sections IV-C and IV-D, the proposed approach allows to study the behavior of an ambient system using DEVS simulations before any WComp implementation. This will allow a Designer of ambient system to select the desired execution machine without having coding and compiling the C# classes under WComp platform.

Furthermore, in this sub-section, we briefly introduce how the DEVS specifications can be use by an ambient system Designer to write the code of execution machine. From the two previous cases one can note that the WComp method of Bean class of a given ambient component and the external transition of the corresponding DEVS atomic model present some similarities (in the one part, see Figure 4 and Figure 8; on the other part, see Figure 5 and Figure 10).

V. CONCLUSION AND FUTURE WORK

This paper dealt with an approach for the design and the implementation of IoT ambient systems based on Discrete

Event Modeling and Simulation. Instead of waiting the implementation phase to detect eventual conflicts, we propose an initial phase consisting in DEVS modeling and simulation of the behavior of components involved in an ambient system, as well as the behavior of execution machines. Once the DEVS simulations have brought successful results, the Designer can implement the behavior of the given ambient system using an IoT framework such as WComp. The presented approach has been applied on a pedagogical example which is described in detail in the paper: implementation of two different behaviors of a given ambient system, definition of the corresponding DEVS specification, implementation of the DEVS behavior using the DEVSimPy framework analysis of the simulation results. Furthermore, we have also pointed out that the DEVS specifications can be used in order to help the Designer to write the behavior of the IoT components. Our future work will consist in proposing an approach allowing to automatically write the behavior of the execution machines after their validation based on DEVS simulation.

In this paper, we used a pedagogical example to model “execution strategies” to prevent conflict.

REFERENCES

[1] L. Capocchi, J. F. Santucci, B. Poggi, and C. Nicolai, “DEVSimPy: A collaborative python software for modeling and simulation of DEVS systems,” Proc. IEEE Conf. WETICE, Eds, IEEE Computer Society, pp. 170–175, 27-29 June 2011, doi: 10.1109/WETICE.2011.31, Available from: http://code.google.com/p/devsimpy. [Retrieved: August, 2014].

[2] D. Cheung-Foo-Wo, “Adaptation dynamique par tissage d’aspects d’assemblage,” PhD thesis, University of Nice Sophia Antipolis, Nice (France), 2009.

[3] E. Jones, T. Oliphant, and P. Peterson, “Scipy: Open source scientific tools for python,” 2001, Available from: http://www.scipy.org. [Retrieved: February, 2014].

[4] X. Li, H. Vangheluwe, Y. Lei, H. Song, and W. Wang, “A testing framework for DEVS formalism implementations,” Proc. Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, Society for Computer Simulation International, San Diego, CA, USA, 2011, pp. 183–188.

[5] T. E. Oliphant. “Python for scientific computing,” Computing in Science and Engineering 9, 2007, pp. 10–20.

[6] F. Perez, B.E. Granger, and J. D. Hunter, “Python: An ecosystem for scientific computing,” Computing in Science and Engineering 13, 2011, pp. 13–21.

[7] N. Rappin, and R. Dunn, “WxPython in action,” Manning, 2006.

[8] B. P. Zeigler, “An introduction to set theory,” Tech. rep., ACIMS Laboratory, University of Arizona, 2003. Available from: http://www.acims.arizona.edu/EDUCATION. [Retrieved: April, 2014].

[9] B. P. Zeigler., H. Praehofer, and T. G. Kim, “Theory of Modeling and Simulation,” Second Edition. Academic Press, 2000.

Optimization of Resources to Improve Patient Experience in the New Emergency Department of Mater Hospital Dublin

Heba Habib, Waleed Abo-Hamad, Amr Arisha
3S Group, College of Business
Dublin Institute of Technology DIT
Dublin, Ireland
heba.habib@mydit.ie, waleed.abohamad@dit.ie, amr.arisha@dit.ie

***Abstract*— Healthcare systems globally are facing capacity issues due to the increased demand of health services, the high cost of resources and the level of quality anticipated of service providers. Emergency Departments (ED) are the most pressurized unit in healthcare systems due to uncertainty in demand and limited resources allocated. Mater Hospital (one of leading hospitals) in Dublin has built a new (state-of-the-art) unit for ED yet faced an issue in resourcing the unit to optimize performance. This paper presents an integrated solution to optimize the capacity of the new ED before opening to public and examine improvement interventions in the ED area. This solution provides ED management with a tool that can contribute significantly in enhancing patient experience by reducing the waiting time from 21 hours to 6 hours while achieving utilization below the 80% burn-out threshold. The model is recommended by Health Service Executives to be used national wide.**

Keywords-Health-care management; patient experience; discrete-event simulation; emergency department.

I. INTRODUCTION

Healthcare management, globally, is under constant pressures of increasing service costs, public demands, and quality expectations from patients. This drives the strategy into one direction, namely, continual improvement of strategies related to patients experience.

While hospitals represent an important part of healthcare service providers, Emergency Departments (EDs) are considered the front line defense in managing the flow of patients into hospitals. The problem faced by ED managers is related to the fact that number of patients who arrive at ED usually exceed the physical capacity of the waiting rooms [1]. Overcrowding can lead to dramatic consequences that may include higher mortality rates for patients [2]. Crowding involves the patients waiting for ED admission, being monitored in non-treatment areas (corridors) and those waiting to be admitted in the hospital (inpatient). Those patients utilize resources in non-treatment areas and their waiting times exceed reasonable periods [3] and the problem can get worse with higher arrival rate [4].

In an Irish context, the Health Service Executive (HSE) is the government entity responsible for the provision of health and social services. The HSE has always addressed in its strategies the urgency to bring real and sustained reforms to Irish healthcare services. In 2007, a scheme has been presented by HSE to reward hospitals that maintain high performance levels [5]. To support continual improvement that leads to reduce the pressure on EDs, HSE has set a target of less than 6 hours to overall Patient Experience Time (PET), i.e., length of stay, within the ED that has been adopted ever since [6].

The ED managers in Irish hospitals have developed a need, since then, for innovative solutions and applications to help them to achieve the target set by HSE and reduce the patient experience time in the emergency department to less than 6 hours. These solutions have to be capable of understanding their system dynamics and increase efficiency, while taking resources utilization and process rationalization into consideration. The challenge for these solutions would be in meeting the aforementioned pressures and managing the huge gap between the needs and costs of healthcare.

Simulation is a powerful tool used to capture the complexity and dynamic features of ED processes. Simulation models have been proven to be an excellent and flexible tool for modeling such kinds of complex environment. A simulation model is an effective tool for testing the effect of different resource allocation schemes, which is crucial for efficient utilization of resources within the ED [7]. A simulation model is also a flexible tool that can be used to simulate the effect of different possible ED settings on patient waiting time [8]. Moreover, multi-performance indicators can easily be measured using a simulation model, as stated by [9]. Simulation modeling used to examine staff scheduling impact on overall utilization and burnout issues related to over-utilized staff [10]. A number of studies in the literature used simulation to model the operation of ED using patient’s waiting time and throughput time as the main target service quality [11]. The impact of staff scheduling can also be investigated using simulation and modeling [12]. It can also be used to analyze the impact of the enhancements, made to the system after the relocation of the emergency department, on the patients flow [13].

Aforementioned studies show that modeling and simulation is currently seen as a competent tool for EDs performance analysis, which allows the effects of actions and changes to be understood and predicted more easily. Compared to change initiatives, tools such as discrete event simulation provide a low risk, lower cost method to develop improvement strategies, test assumptions, and observe potential outcomes of decisions prior to implementation. Numerous discrete event simulation applications are found in healthcare, but very few demonstrate a pre-/post-intervention comparison [14]. Frequently, healthcare decision makers use subjective information from frontline staff, providers, and other stakeholders to make strategic improvement decisions. Change attempts whether to structures (e.g., change in floor plan or layout) or to processes, may prove costly in terms of time and capital [15].

This paper presents a simulation based solution to the key healthcare service providers (i.e., Public Hospitals) in order to help them to make the optimum decision in such stochastic environment. The simulation-based model will offer a tool for the decision makers to examine different scenarios for the given variables of the system. This will enable them to envisage the impact of the decisions on patient throughput time and resource utilization. The interpretations of the model output also allow the decision makers to gain new insights into the complexity of the interrelated variables and the effect of changes on the overall performance of the ED units. This research is a continuity of research work published in WSC 2012 [16]. Their work contributed to the simulation application in healthcare services with particular interest in ED in Irish hospitals [17]. This paper presents a simulation-based solution to develop effective strategies to reallocate an existing ED to a new ED in one of the largest hospitals in Dublin. The impact of the additional capacity on patient experience time will be assessed prior to opening the ED to the public. The main objective is to evaluate the effectiveness of the new capacities and accordingly optimize nursing staff to cope with the increased demand of care.

Section II gives an overview of the project's background, highlighting the nature of the partner hospital. Section III describes the suggested scenarios proposed by the ED management. Section IV presents and discusses results from the different scenarios used and applied. Section V concludes the paper, presenting the best proposed scenario to decrease the PET.

II. Project Background

Mater University hospital in Dublin is an acute care public hospital in North Dublin. It provides a variety of healthcare services and has a total of 570 beds on premises, with a 24-hour ED that receives over 55,000 patients annually. The current ED of the hospital has 13 monitored trolley spaces, 3 of which are in a resuscitation area and are reserved for major trauma and critical care patients; an ambulatory care area (capacity 6 trolley spaces); two isolation rooms; a psychiatric assessment room; two rapid assessment triage bays; and two other triage rooms. The layout of the ED is shown in Figure 1, provided by the ED management.

Five distinct areas can be identified: a waiting room for walk-in patients waiting for triage, a diagnostics area (X-ray and CT scan), an ambulatory care unit (ACU) area, an ED resuscitation area (CPR), and an ED major assessment area. Patients arriving by ambulance – usually in critical condition– are routed directly to the resuscitation area, whereas patients whose conditions require monitoring stay in the major assessment area. The ambulatory care area is for walk-in patients, who may be suffering from abdominal pain, headache, limb problems, wounds, head injuries, and facial problems. As a 24-hours department, the ED has three consultants, two nursing managers, and eleven nurses during the day and nine nurses at night, divided into six types of nurse: Advanced Nurse Practitioners (ANPs), triage nurses, resuscitation nurses, respiratory nurses, majors/minors nurses, and health-care assistants. Physicians (excluding the three consultants who provide cover between 9 am and 5 pm (or 8 am and 8 pm) with 24/7 on-call provision) are divided into three types, registrar/specialist registrars; Senior House Officers (SHOs), and interns, and are distributed as follows when the roster allows: three registrars per day working 10-h shifts starting at 8am, 12pm and 10pm; two interns working daily 8 am – 5 pm shifts Monday to Friday; and 12 SHOs working fixed shifts during the day and night to keep the ED running.

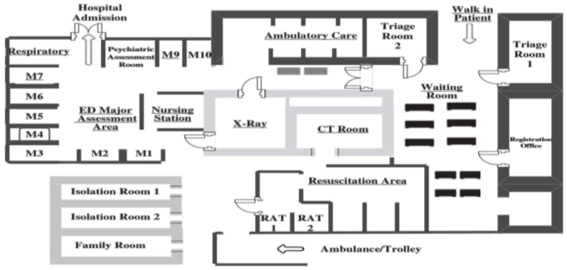


Figure 2 ED physical layout and main care areas.

Figure 1. ED physical layout and main care areas

According to the task force report in 2007 [18], the overall physical space of the ED and infrastructure were inadequate. The hospital – which was operating at approximately 99% occupancy – had difficulty in accommodating the increasing flows in ED admission numbers. Therefore, patients who required critical care (ICU/HDC) beds suffered from significant delays and the ED could not meet the national target of 6 hours average Length Of Stay (LOS) for patients. The ED figures showed a clear evidence of overcrowding with an average of 17% of its patients choosing to leave before being seen by the ED clinician. The report also indicated that the average time from ED registration to discharge was 9.16 hours, that is 3.16 hours over the 6 hours metric set by the HSE, and the average LOS from registration to acute admission was 21.3 hours with a standard deviation of 17.2 hours (i.e., 3.5 times higher than the same national metric). Obviously, patients who are admitted will usually experience longer LOS times than those who are discharged due to delays between admission referral by an ED doctor, the allocation of a bed, and time taken to transfer the patient to the bed.

To cope with these challenges, a simulation-based framework was developed by Abo-Hamad and Arisha [16], using ExtendSim v.8 [19], aiming to identify performance bottlenecks and explore improvement strategies to meet the HSE targets. To reduce the time of the model development cycle and to increase the confidence of the ED simulation model results, verification and validation were carried out throughout the development phases of the model. Furthermore, each model development phase was verified and validated against the previously completed phases. The final results of the simulation model were validated using face validation and comparison testing. Face validation was performed by interviewing ED senior managers and nursing staff to validate the final results of the simulation model. Comparison testing involved comparing the output of the simulation model with the real output of the system under identical input conditions According to the ED managers, the goal was to assess the performance of the ED if the average LOS of patients complies with the HSE 6 hours target. Therefore the framework was developed and used to assess the implications of a number of strategies. These strategies were the impact of variation in medical staffing, increasing clinical assessment space and finally assessing the impact of incorporating a ‘zero-tolerance’ policy regarding exceeding the national 6 hours LOS. The importance of this assumption was emphasized by the senior hospital decision makers to identify the real factors that contribute the unacceptable overcrowding status of the current ED; inappropriate physical space, insufficient staffing levels, or operational difficulties beyond the direct control of the ED. By using the model, the ED managers were able to reveal that enforcement of the national benchmark of 6-hour limit for EDs would have a significantly greater impact on reducing average LOS for all ED patients than increasing medical staff or assessment cubicles. Access block therefore, has been shown by the model to have the highest impact on prolonged average LOS for patients. Based on these recommendations, the ED management proposed the introduction of a new unit called Acute Medical Unit (AMU) that is co-located with the ED.

Meanwhile, a new campus for the hospital was completed. The hospital development was a €284 million redevelopment including 55,000m^2 of new acute hospital services including a new Emergency Department, Outpatients Department, GI unit, 12 new operating theatres, ICU and HDU, Radiology Department and 134 single en-suite bedrooms. However, the new campus planners and senior hospital management wanted to model the relocation of the present ED to the new ED. The purpose is to assess the capacity of the new ED before opening to the public and to check whether the new capacities will cope with the demand of care or other strategies will be required. The assessment of the impact of these changes will be discussed in the next sections.

III. Strategy Assessment

Three strategies were proposed by the ED management to be assessed by the proposed model. The first is the new changes of the ED layout. The objective of these changes was to provide better patient flow and more space to avoid overcrowding and allowing the management team to increase the number of beds in each medical assessment area (Table I).

The second scenario suggests that the hospital opens an AMU. The unit will be divided into two units; an acute medical assessment unit (AMAU) that opens from 8am to 8pm and a Short Stay Unit (SSU). The purpose of the AMAU is to facilitate the immediate medical assessment, diagnosis and treatment of medical patients who suffer from a wide range of medical conditions who present to, or from within, a hospital requiring urgent or emergency care. The required patients can be admitted to the SSU for a short period for acute treatment and/or observation where the estimated length of stay is less than 48 hours. The logic behind limiting the working hours in the AMAU to only 12-hours is to allow enough time for patients to be admitted to the hospital and make sure that this new pool of beds will always be available at the beginning of the next day. The unit will act then as a gateway between the ED and the wards of the hospital, and it will help increasing the throughput of the ED to achieve the national target of 6 hours.

TABLE I. Capacities of ED before and after changing the layout

Area	Current ED	New ED
CPR Area	03	05
ACU Area	10	15
ED Majors Area	06	11
Total	19	31

The third strategy scenario is related to the optimization of the staff levels with particular interest in nursing team due to the burn-out factor in two areas in the ED: Resuscitation Area and Majors Area. The ED managers were concerned that the nurses will be well over utilized due to the increase in the ED physical capacity. Therefore an optimization experiments were conducted to find the best combination of nurses for each shift which will enable the hospital to achieve the national 6-hour target without exceeding the threshold of staff burn-out. Flexibility in resource allocation and optimization of workload is a matter of urgency to the ED management team [10].

In summary, the strategy scenarios will model the impact of variation of layout, increasing the capacity in each assessment area, namely, CPR, majors area, and ACU area, open AMAU & SSU to avoid access blockage of patients to the 'upstream' hospital beds, and finally assessing the impact of optimizing the nursing staff level in each shift (Table II).

TABLE II. Simulation variables for base scenario and scenarios 1, 2, and 3

	Change Layout	CPR	Majors	ACU
Base line	No	3	10	6
Sc. 1	Yes	5	15	11
Sc. 2	Yes	5	15	11
Sc. 3	Yes	5	15	11

	Open AMU	Nurse - Day	Nurse - Night
Baseline	No	6	4
Sc. 1	No	6	4
Sc. 2	Yes	6	4
Sc. 3	Yes	6 - 11	4 - 9

ISBN: 978-1-61208-371-1

IV. RESULTS ANALYSIS

The first investigated scenario is to change the layout of the ED and increase the capacities of the three main assessment areas, and investigate the effect of that change on PET. Increasing the physical space by 65% (i.e., scenario 1) will decrease the number of patients in the waiting room, though the number of admitted patients will increase. The effect will be cascaded back through the ED progressively with more patients waiting on trolleys to be admitted to the hospital. Consequently the PET will increase for patients who are waiting to be admitted to the hospital (Table III). As a result, there will be no space left to meet the timely needs of the next patients who need emergency care.

TABLE III. COMPARISON BETWEEN SIMULATION RESULTS OF CURRENT ED AND NEW ED

Performance Indicators	Current ED	New ED
PET-Admitted *(hours)*	20.9	23.4
PET-Discharged *(hours)*	10.0	10.3
No. patients in waiting room	16.5	04.8

To prevent access blockage of patients who are required to be admitted to the hospital, the introduction of the acute medical assessment unit was the second suggested scenario. The question was how many beds are needed to unlock the access blockage, given the current demand of care. The unit was modeled as an additional assessment unit that deals with a wide variety of medical patients that present to the ED. A number of experiments were designed determine the optimal number of beds needed for the unit, as shown in Table IV.

TABLE IV. SIMULATION OUTPUT FOR DIFFERENT BED SETTINGS OF THE AMU

Performance Indicators	Number of beds in AMU				
	22	*23*	*24*	*25*	*26*
PET-Admitted *(hours)*	17.15	14.34	12.49	11.81	11.22
PET-Discharged *(hours)*	11.85	11.73	11.91	12.21	12.08
% Patients Treated	90%	91%	92%	93%	94%
No. patients in waiting room	17.30	16.40	15.80	14.30	16.30
No. patients waiting to be admitted	06.23	03.65	01.82	01.05	00.52

The capacity of the 25 beds was agreed to be divided between the two new units, namely AMAU and SSU, with 12 beds being assigned to the AMAU and 13 beds to the SSU. After that, the management decided to increase the capacity of the SSU with 6 more beds, leading to 19 beds overall in the SSU. Arguably, it was justified that the increase in the number of beds in the short stay unit was due to the utilization of the SSU in comparison to the AMAU. Integrating the new factors into the simulation model developed has showed that the national 6 hours admission target is still not met. However, the introduction of this intervention has provided a new decline in the PET (Table V).

ED Management demonstrated interests in the optimization of resources – with a particular emphasis on nurses. The question is "How many nurses should be availed in every shift to achieve the national target of 6-hours)?". The management team also emphasized on the importance of avoiding the burn-out of nurses due to long working hours. A full factorial design of experiment was constructed based on an orthogonal array (L36) [20]. Thirty six experiments were carried out with different levels of nursing staff, varying between 6-11 nurses during the day and 4-9 nurses during the night shift. The L36 design allows for testing the two factors at each of their levels to analyze their impact on the responses (i.e., outputs), namely the PET-All, PET-Admitted, PET-discharged and the utilization of the nurses, as recommended by the managers for each experiment. Table VII shows the impact of different levels of nursing staff for each shift on the performance indicators and on the nurses utilization.

TABLE V. COMPARISON BETWEEN SIMULATION RESULTS OF CURRENT ED WITH AMU AND NEW ED WITH AMU

Performance Indicators	Current ED + AMU	New ED + AMU
PET-All *(hours)*	09.85	6.52
PET-Admitted *(hours)*	13.91	6.88
PET-Discharged *(hours)*	07.66	6.31

Sensitivity analysis of number of nurses in each shift and different performance indicators (Figure 3) demonstrated that assigning 9 nurses on the day shift and 7 nurses on the night shift will improve performance. This resource schedule along with other recommendation can easily help the hospital to achieve the national target while maintaining the utilization level of the nurses at 80% (less than burn-out level) [10].

The results of the simulation model showed that the adoption of scenario 3 has the greatest impact on the patient experience time in the ED as shown in Figure 4. The recommendations of scenario 3 are changing the layout of the ED along with increasing the capacity of the assessment areas by 65%, open an AMAU and SSU units with 12 and 19 assessment areas respectively, and change the number of nurses on floor to 9 nurses during the day and 7 nurses at night. Scenarios 1 and 2 both showed a significant improvement from the baseline scenario, with patient experience times decreasing substantially, yet both failed to achieve the national target of less than 6 hour for the experience time of patients.

For verification of the simulation results, an overall confidence of interval of the simulation output was obtained with at least 95% confidence level that all KPIs lie in the dimensional box defined by the confidence intervals. Ten replicas of each scenario was generated to obtain the results of the confidence intervals for each KPI (Table VI).

To avoid the inflated error rate that resulted from using separate confidence intervals, a joint confidence region was constructed, based on Hotelling's T^2 distribution, which is a generalization of the univariate t-distribution [21]. The three KPIs were chosen for calculating the T^2 value, namely, the PET-All, PET-Admitted, and the PET-Discharged. The confidence regions were calculated for each of the three scenarios. For simplicity, Figure 2 shows confidence regions between each pair of the three main KPIs of the ED along with their individual confidence intervals.

TABLE VI. SIMULTANEOUS CONFIDENCE INTERVALS

	Simultaneous CIs (t = 4.0947)					
	New ED		*New ED + AMU*		*New ED + AMU + Staff Opt.*	
	LB	*UB*	*LB*	*UB*	*LB*	UB
PET-All	6.6	8.9	5.7	7.6	4.1	6.1
PET-Admitted	6.8	9.4	6.3	8.3	4.5	6.7
PET-Discharged	6.4	8.5	5.4	7.2	3.9	5.7

The following figures compare the confidence region of each pair of the performance indicators, namely the PET of all patients; PET of admitted patients and PET of discharged patients across the three scenarios. The confidence region on the upper left corner of each graph represents the region formed by the two indicators under the first scenario, i.e., changing the layout of the ED and increase the capacity of the three assessment areas by 65%. While the region formed in the center of each graph shows the confidence region of each pair of indicators for the second scenario. Finally the lower right corner of each graph is the confidence region for the third scenario; that combines the first and second scenarios with optimum number of nurses. Figure 2 shows that increasing the capacity of the ED will not solve the overcrowding problem unless other interventions are introduced. Thus, finding the appropriate staffing levels is a key to cope with such increase in capacity along with the incorporation of a block-free unit (i.e., AMU)

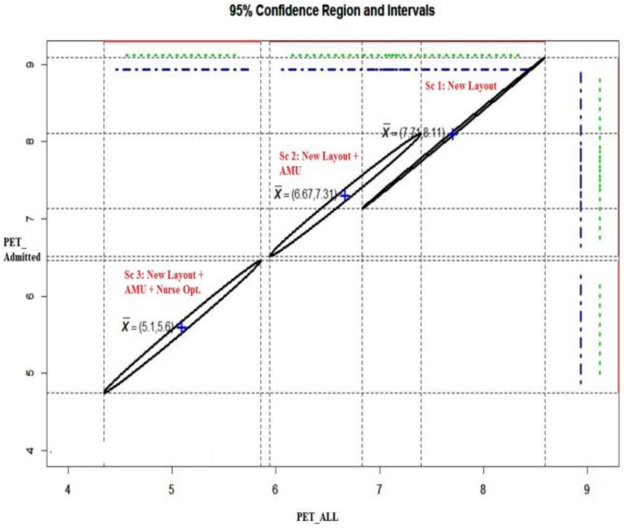


Figure 2. a. Confidence regions between PET of all patients, and PET of admitted patients for three scenarios

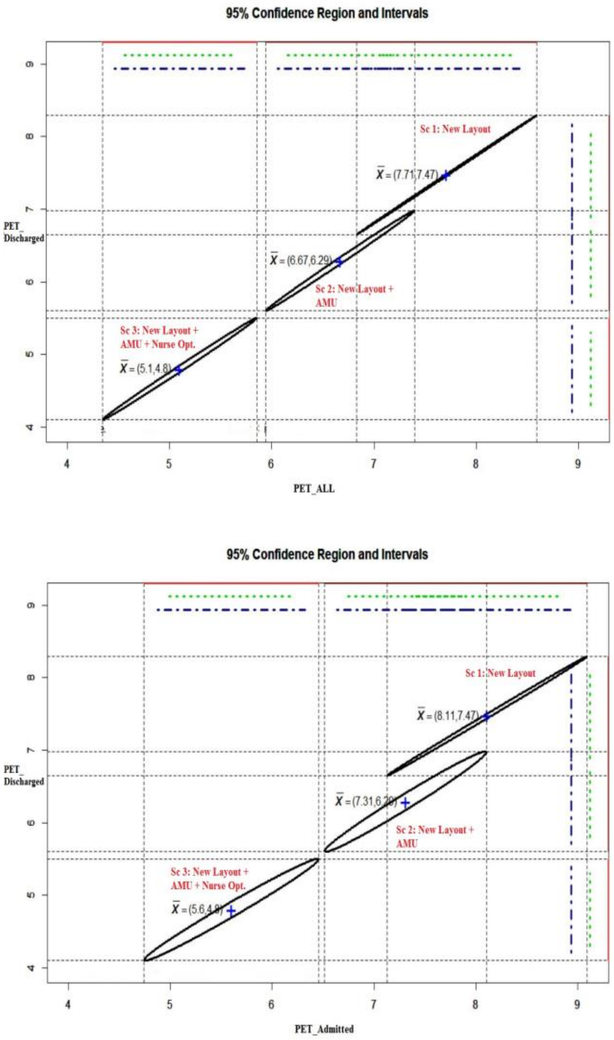


Figure 2. b. Confidence regions between PET of all patients, PET of admitted patients and PET of discharged patients for three scenarios

V. CONCLUSION

The increased demand for ED services in Irish Hospitals puts an immense pressure on healthcare systems to meet patients' expectations. It is evident that increasing one or two particular resources (e.g., layout) will not be adequate to significantly improve the patient experience time in the ED. The delays that the patients experience is strongly related to bed access 'upstream' in the hospital, thus whether a bed is available for the patient to be admitted or not is one of the main factors influencing the patient throughput. This paper presented a simulation-based solution that is used to develop effective strategies to reallocate the present ED to the new ED for Mater hospital in Dublin.

The business model of the present ED was projected into the layout of the new ED to assess the impact of the additional capacity on patient experience time. These changes were investigated before opening the ED to the public to check whether the new capacities will cope with the demand of care or other strategies will be required. The first intervention was the changes in the layout of the ED from 19 to 31 assessment areas that were proposed to enhance the patient flow. The simulation model showed that the new capacity will not reduce patient experience time. However, it will help to decrease the number of patients in the waiting room, though the number of admitted patients will increase. The second intervention that was assessed is the introduction of an acute medical unit (AMU) with access to a minimum of 25 assessment areas. This intervention was recommended based on previous research on the present ED. The unit is to be collocated with the ED and should have limited working hours allowing the pool of assessment areas to be vacant for the next following day. This unit contributed significantly in addressing the problem of upstream bed blockage in the hospital, leading to more than 50% reduction of the total patient experience time in the ED. Also, combining these interventions together led to more than 70% reduction in the total patient experience time. Finally the third scenario is to optimize the allocation of nurses in two areas of the ED: Resuscitation Area and Majors Area. The optimization process resulted into allocating 9 nurses in the day shift and 7 nurses in the night shift. Optimizing the resources in the ED will enhance the overall performance of the unit and also maintain the burn-out factor below 85%. The combination of these three interventions resulted in a significant reduction of PET from 21 hours to 6 hours and also to achieve 95% of patients to be seen by a doctor in less than 6 hours. ED management has engaged in the process of the solution and the confidence of the simulated results has increased significantly after the validation of the model.

Due to the flexibility and extendibility features of the framework, it provided the hospital management with a lower cost method to develop improvement strategies, test assumptions, and observe potential outcomes of decisions prior to implementation. The confidence level of the project output encouraged the HSE to adopt the framework to design and plan the introduction of the AMU in every hospital across the country. A pilot study has been initiated for the largest six hospitals in Dublin, which will be implemented at a later stage national wide.

REFERENCES

[1] L. F. McCaig and C. W. Burt, “National Hospital Ambulatory Medical Care Survey: 2002 emergency department summary.,” *Adv. Data*, no. 340, pp. 1–34, Mar. 2004.

[2] E. J. Carter, S. M. Pouch, and E. L. Larson, “The relationship between emergency department crowding and patient outcomes: a systematic review.,” *J. Nurs. Scholarsh.*, vol. 46, no. 2, pp. 106–15, Mar. 2014.

[3] J. H. Han, D. J. France, S. R. Levin, I. D. Jones, A. B. Storrow, and D. Aronsky, “The effect of physician triage on emergency department length of stay.,” *J. Emerg. Med.*, vol. 39, no. 2, pp. 227–33, Aug. 2010.

[4] A. Joshi and M. Rys, “Study on the Effect of Different Arrival Patterns on an Emergency Department’s Capacity Using Discrete Event Simulation.,” *Int. J. Ind. Eng. 2011*, vol. 18, no. 1, pp. 266–275, 2011.

[5] HSE, “Health Service Executive National Service Plan,” 2007.

[6] HSE, “Health Service National Service Plan 2014 Our Service Priorities,” 2014.

[7] M. A. Centeno, R. Giachetti, R. Linn, and A. M. Ismail, “A simulation-ILP based tool for scheduling ER staff,” in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*, 2003, vol. 2, pp. 1930–1938.

[8] D. Sinreich and Y. N. Marmor, “A Simple and Intuitve Simulation Tool for Analyzing the Performance of Emergency Departments,” in *Proceedings of the 2004 Winter Simulation Conference, 2004.*, 2004, vol. 2, pp. 908–916.

[9] M. E. Matta and S. S. Patterson, “Evaluating multiple performance measures across several dimensions at a multi-facility outpatient center,” *Health Care Manag. Sci.*, vol. 10, no. 2, pp. 173–194, May 2007.

[10] M. Thorwarth, P. Harper, and A. Arisha, “Simulation Model to Investigate Flexible Workload Management for Healthcare and Servicescape Environment,” in *Proceedings of the 2009 Winter Simulation Conference, 2009*, 2009, pp. 1946 – 1956.

[11] K. Ismail, W. Abo-Hamad, and A. Arisha, “Integrating balanced scorecard and simulation modeling to improve Emergency Department performance in Irish hospitals,” in *Proceedings of the 2010 Winter Simulation Conference*, 2010, pp. 2340–2351.

[12] W. Abo-hamad and A. Arisha, “Towards Operations Excellence: Optimising Staff Scheduling For New Emergency Department,” in *Proceedings of the 20th International Annual EurOMA Conference*, 2013, pp. 9–12.

[13] O. Rado, B. Lupia, J. M. Y. Leung, Y.-H. Kuo, and C. A. Graham, “Using Simulation to Analyze Patient Flows in a Hospital Emergency Department in Hong Kong,” in *Proceedings of the International Conference on Health Care Systems Engineering*, 2014, vol. 61, pp. 289–301.

[14] E. Hamrock, K. Paige, J. Parks, J. Scheulen, and S. Levin, “Discrete event simulation for healthcare organizations: a tool for decision making.,” *J. Healthc. Manag.*, vol. 58, no. 2, pp. 110–124, 2013.

[15] M. Jahangirian, S. J. E. Taylor, and T. Young, “Economics of Modeling and Simulation: Reflections and Implications for Healthcare,” in *Proceedings of the 2010 Winter Simulation Conference*, 2010, pp. 2283–2292.

[16] W. Abo-Hamad and A. Arisha, “Multi-Criteria Framework for an Emergency Department in an Irish Hospital,” in *Proceedings of the 2012 Winter Simulation Conference*, 2012, pp. 1–12.

[17] W. Abo-Hamad and A. Arisha, “Multi-criteria approach using simulation-based balanced scorecard for supporting decisions in health-care facilities: an emergency department case study,” *Heal. Syst.*, vol. 3, no. 1, pp. 43–59, Nov. 2014.

[18] HSE, “Chapter 4 - Task Force findings and recommendations by hospital,” 2007.

[19] I. T. Inc., “ExtendSim Simulation Software,” 2014. [Online]. Available: https://www.extendsim.com.

[20] G. Taguchi, S. Chowdhury, and Y. Wu, “Orthogonal Arrays and Linear Graphs : Tools for Quality Engineering,” in *Taguchi’s Quality Engineering Handbook*, 2005, pp. 35–38.

[21] C. Alexopoulos, “Statistical analysis of simulation output: state of the art,” in *Proceedings of the 2007 Winter Simulation Conference*, 2007, vol. 100, pp. 150–161.

TABLE VII. RESULTS OF DOE FOR NURSE ALLOCATION IN DAY AND NIGHT SHIFTS

No.	Nurse-Day	Nurse-Night	*PET-All*	*PET-Ad.*	*PET-Dis.*	*Utilization*
1	6	4	*7.8*	*8.4*	*7.4*	*100%*
2	7	4	*6.8*	*7.5*	*6.4*	*100%*
3	8	4	*6.5*	*7.2*	*6.1*	*98%*
4	9	4	*6.5*	*7.1*	*6.1*	*93%*
5	10	4	*6.2*	*6.8*	*5.9*	*88%*
6	11	4	*6.3*	*6.8*	*6.0*	*83%*
7	6	5	*6.5*	*7.2*	*6.1*	*100%*
8	7	5	*5.8*	*6.3*	*5.5*	*100%*
9	8	5	*5.7*	*6.3*	*5.3*	*94%*
10	9	5	*5.4*	*5.9*	*5.1*	*89%*
11	10	5	*5.6*	*6.3*	*5.2*	*84%*
12	11	5	*5.5*	*6.1*	*5.1*	*79%*
13	6	6	*6.0*	*6.5*	*5.6*	*100%*
14	7	6	*5.5*	*6.1*	*5.2*	*95%*
15	8	6	*5.2*	*5.6*	*4.9*	*90%*
16	9	6	*5.3*	*5.8*	*4.9*	*84%*
17	10	6	*5.0*	*5.4*	*4.8*	*80%*
18	11	6	*5.1*	*5.6*	*4.9*	*77%*
19	6	7	*5.7*	*6.2*	*5.4*	*96%*
20	7	7	*5.3*	*5.7*	*5.0*	*90%*
21	8	7	*5.1*	*5.5*	*4.8*	*85%*
22	9	7	*5.1*	*5.6*	*4.8*	*81%*
23	10	7	*5.2*	*5.8*	*4.8*	*77%*
24	11	7	*5.1*	*5.5*	*4.8*	*74%*
25	6	8	*5.6*	*6.1*	*5.3*	*91%*
26	7	8	*5.2*	*5.8*	*4.9*	*86%*
27	8	8	*5.3*	*5.8*	*4.9*	*82%*
28	9	8	*4.9*	*5.3*	*4.7*	*78%*
29	10	8	*5.1*	*5.7*	*4.8*	*74%*
30	11	8	*4.9*	*5.3*	*4.7*	*71%*
31	6	9	*5.7*	*6.2*	*5.3*	*87%*
32	7	9	*5.3*	*5.8*	*4.9*	*82%*
33	8	9	*5.1*	*5.6*	*4.8*	*78%*
34	9	9	*5.0*	*5.4*	*4.7*	*75%*
35	10	9	*5.0*	*5.5*	*4.7*	*71%*
36	11	9	*5.1*	*5.6*	*4.7*	*69%*

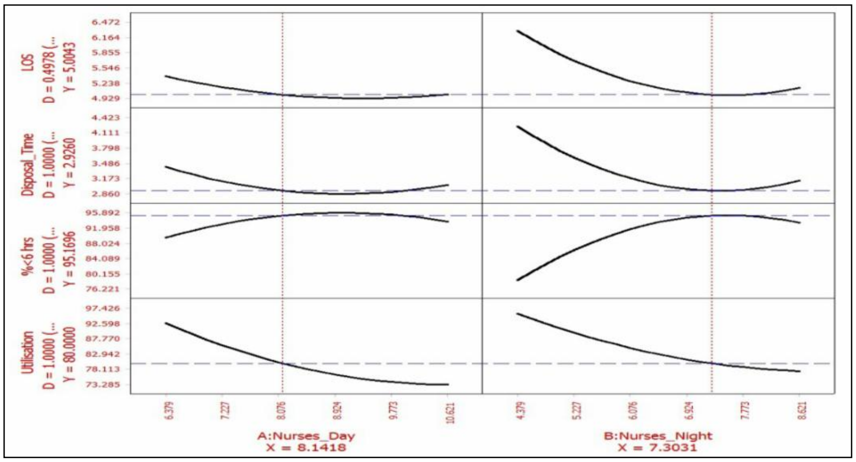
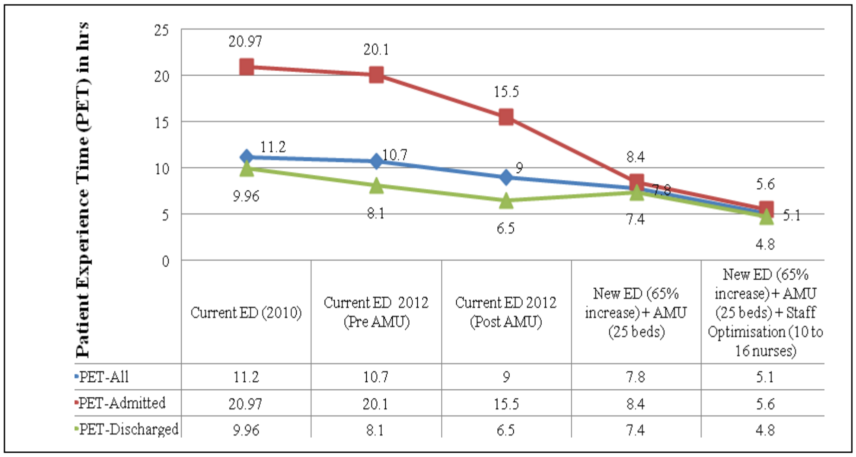


Figure 3. Sensitivity analysis of number of nurses in each shift against performance indicators



	Current ED (2010)	Current ED 2012 (Pre AMU)	Current ED 2012 (Post AMU)	New ED (65% increase) + AMU (25 beds)	New ED (65% increase) + AMU (25 beds) + Staff Optimisation (10 to 16 nurses)
◆PET-All	11.2	10.7	9	7.8	5.1
■PET-Admitted	20.97	20.1	15.5	8.4	5.6
▲PET-Discharged	9.96	8.1	6.5	7.4	4.8

Figure 4. Comparison of simulation results of different scenarios

Modeling the Contact Propagation of Nosocomial Infection in Hospital Emergency Departments

Cecilia Jaramillo, Dolores Rexachs
and Emilio Luque

Computer Architecture and
Operating Systems Department
Autonomous University of Barcelona
Bellaterra/Barcelona, Spain
Email: cjaramillo@caos.uab.es,
dolores.rexachs@uab.es,
emilio.luque@uab.es

Francisco Epelde

Medicine Department,
Hospital Universitari Parc Taulí
Autonomous University of Barcelona
Sabadell/Barcelona, Spain
Email: fepelde@tauli.cat

Manel Taboada

Tomas Cerda Computer Science School
Autonomous University of Barcelona
Sant Cugat/Barcelona, Spain.
Email: manel.taboada@eug.es

***Abstract*—The nosocomial infection is a special kind of infection that is caused by microorganisms acquired inside a hospital. In the daily care process of an emergency department, the interactions between patients and sanitary staff create the environment for the transmission of such microorganisms. Rates of morbility and mortality due to nosocomial infections are important indicators of the quality of hospital work. In this research, we use Agent Based Modeling and Simulation techniques to build a model of Methicillin-resistant Staphylococcus Aureus propagation based on an Emergency Department Simulator which has been tested and validated previously. The model obtained will allow us to build a contact propagation simulator that enables the construction of virtual environments with the aim of analyzing how the prevention policies affect the rate of propagation of nosocomial infection.**

***Index Terms*—Agent Based Modeling and Simulation; Nosocomial Infection; Hospital Emergency Departments.**

I. Introduction

The nosocomial infection is a kind of infection that is caused by microorganisms acquired inside a health care environment. It is the most common type of complication affecting hospitalized patients. Inside a health care environment we can find several microorganism that can be causative of a nosocomial infection, but our work has focused in the propagation of the Methicillin-resistant Staphylococcus Aureus (MRSA), one of the most common and dangerous microorganisms in this environment. The presence of these bacteria could mean serious health problems for a patient. It is a common cause of skin, wound and, most seriously, blood stream infections as it may be responsible for a greater hospital length of stay, expensive treatments and an increased mortality [1]. MRSA is a bacteria usually resistant to conventional antibiotics that makes it very difficult to treat. These bacteria live in the skin of some patients and could be transmitted to another patient by physical contact through the interaction between patients, healthcare staff and environment. The most common transmission vias are: healthcare staff's hands, contaminated medical equipment and objects in the hospital room environment. The risk of MRSA acquisition is particularly high in elderly patients, severe underlying disease, prolonged hospitalization (especially in ICUs, surgery and burns units), use of invasive medical devices, previous antibiotic treatment and exposure to infected or colonized patients [2].

An emergency department is undoubtedly one of the most complex and dynamic areas in a hospital. Its operation is not linear and depends on several factors. This way, we can conclude that an emergency department can be classified as a complex system. When we work with complex systems, one of the strongest problems is how we can represent human behaviors that cannot be predicted using conventional methods such as qualitative or statistical analysis. Modeling techniques using agents can bring the most benefit when applied to human systems, where agents exhibit complex and stochastic behavior and the interaction between agents are heterogeneous and complex [3]. In our research, we make use of an Agent Based Model and Simulation (ABMS) to create a contact propagation model of MRSA inside a emergency department. For this purpose, we defined all actors involved in the emergency departament process and their specific function and behavior: patients, doctors, nurses, admission staff, laboratory technicians, auxiliary personnel and cleaning staff. Every person who has a role in the emergency department is defined as an active agent in our model. It is very important to take into account the environmental objects that healthcare staff use in the process of attention patient because our model is focused on the transmission by contact and this transmission could be between active agents (direct transmission) or between active and passive agents (indirect transmission). Each one of the agents who is interacting in a emergency department can be a multiplier of MRSA bacteria, which depends on the condition of *colonized* or *infected*, in the case of active agents, or the condition *contaminated* in the case of passive agents. In all cases, we apply the term ”transmission vector” to any agent capable of transmitting MRSA bacteria and ”susceptible” to any agent that has risk to adquire the infection.

Some health services have implemented concrete actions that attempt to control the rate of propagation of nosocomial infections. These actions are called ”prevention policies” and are represented in our model through the definition of behaviors performed by members of the sanitary staff.

The remainder of this article is organized as follows: the previous emergency department model is detailed in Section 2. Section 3 describes the related and previous works. Section 4 the proposed contact propagation model of nosocomial infection. Finally, Section 5 closes this paper with future works and conclusions.

II. Emergency Department Simulator. Previous Model

We based our model in a previous emergency department model (ED-Simulator) that had been developed as part of a previous research work [4][5][6] in our research group.

This model considers the emergency department divided into two zones, A and B. The patients are divided in 5 acuity levels according to the Spanish Triage System [7], very similar to the Canadian Triage System. Acuity level I means that the state of patient is very serious and a patient with acuity level V is the patient with the least severity. Patients with acuity level I, II and III are located in zone A, and patients IV and V are located in zone B. The agents are divided into active agents and passive agents. The active agents represent people who act upon their own initiative and passive agents represent systems that are solely reactive, such as patient information system and diagnostic services (radiology service and laboratories). Each agent has variables and behavior. The behavior depends of the kind of agent and the interactions between agents allow a system behavior to emerge. The behavior of agents has been modeled using Moore State Machines [8]. In each state, the agent has a set of probable outputs. Each state of the state machine for a specific agent has been defined on basis of the values of state variables of each agent at a certain time. Moreover, each one of the state variables could have more than one possible value. The agent will remain in a specific state until, through interaction with other agents, they receive an ”input” (an output generated by other agents), which could cause a change in the state of such agent and generating an ”output” sent to the agent with whom they are interacting. The agents state machine will move to the next state following the transition, which may be another state or the same one in which the agent was before the transition.

With this background, our proposal is obtain an efficient model of the propagation of MRSA by adding new features to the actual model through the use of ABMS technics.

Propagation of nosocomial infections has already been widely studied using different techniques. In this section, we will refer to some relevant papers relating to the modeling of the transmission of MRSA with ABMS techniques. These investigations focus on contact transmission of MRSA mainly through the interaction between patients, doctors and nurses, other members of healthcare staff are not included. The interaction between patients or healthcare staff with the environmental objects or medical equipment have only been dealt with in three research papers. In general, the patients are divided into colonized or infected patients and the healthcare staff are divided into colonized and non-colonized or transiently colonized. Neither considers the severity level of the patients and only two members of healthcare staff are included in the simulation, doctors and nurses. All of these features are resumed in the Table I.

TABLE I: COMPARISON BETWEEN OUR RESEARCH AND RELATED RESEARCHES.

Characteristic / Researches		8	9	10	11	12	Our work
Interaction between patients and...	Doctors	x	x	x	x	x	x
	Nurses	x	x	x	x	x	x
	Others healthcare staff						x
	Equipment and objects			x			x
Prevention policies		x	x		x	x	x
Patient risk adquired NI		x		x	x		x
Transmission probability			x		x	x	x

In Barnes et al. [9], MRSA transmission reduction using agent-based modeling and simulation is presented. The environment of this simulation is a hospital ward. Two types of interactions are modeled: patients-healthcare staff, and patients-visitors. The interaction between the members of the healthcare staff is considered unnecessary for the model. Patients are generated continually and are housed in a waiting room until replacing the discharged patient, so hospital wards are always full. Admitted patients can take one of two states: susceptible or colonized. It is not taken into account the possibility that the patients come as infected but they can develop infection during their stay. Members of the healthcare staff are created at the beginning of the simulation and are all considered as being in an uncolonized state. During the simulation the healthcare staff could be susceptible or colonized. This research includes visitors as agents and they have a colonized status (non-colonized, colonized); therefore, they can transmit the infection. The visitors only have interaction with the patients. The transmission of MRSA between agents is based on the risk level of the patient to becoming infected and the behavior of the healthcare staff members who visit the patient.

In Milazzo et al. [10], the following factors are taken into account: the status and the movement of each agent, the contacts between individuals during a ward round, the hand hygiene compliance for each agent, and the control measures applied. The environment of the simulation is a hospital ward. The main parameters that characterize each agent, patient and healthcare staff, are: the colonization status, the transmission probability, and the compliance factor. The transmission dynamics of the infection is based only on transmission from patient to patient via healthcare staff. The model does not take into account the direct transmission from patient to patient or the indirect transmission from patient to patient via environmental surfaces. The healthcare staff may become transiently colonized and carry MRSA on their hands if their compliance with hand hygiene is poor.

The model proposed by Meng et al. [11] is a propagation model in a single hospital ward divided into bays, with some isolation rooms. Transmission is modeled by pairwise interaction between colonized and non-colonized patients, patient and healthcare staff (nurse and doctor) transiently or permanently colonised, patient-to-patient contacts and transmission from a contaminated environment. This model takes into account the susceptibility of the patient to colonisation. Some possible states are defined for the patient: colonisation, detection, decolonisation treatment and location status. This model includes the time required to treat colonization or infection carried by the patient and a metric that assesses the patient's susceptibility to acquire a nosocomial infection. This model therefore assumes that a susceptible patient may acquire MRSA due to the presence of colonised patients in the vicinity, regardless of the mode of transmission.

According to Raboud et al. [12], a Monte Carlo simulation was used to model MRSA transmission in a hospital ward. They used the simulation to study the impact of different components of infection control programs on the propagation of MRSA on a hospital ward representative of a general hospital ward. Visits from healthcare staff to patients were simulated and MRSA was assumed to be transmitted from patient-to-patient via healthcare staff. Once colonized, healthcare staff remained colonized until they next washed their hands. The model did not address transmission related to healthcare staff who chronically carry the organism or to contaminated environment surfaces or equipment.

In Barnes et al. [13], a dynamic patient network model is defined through ABMS techniques. The environment is a hospital unit. They explicitly define only patients as agents. Patients have a single boolean state that indicates whether or not a patient is infected or colonized with some type of pathogen. In the simulation, two healthcare staff types are defined, nurses and doctors that they are not modeled as agents but they are implicit through the transmission mechanism. Each patient has a primary nurse and a primary doctor who provide care a patient during the hospital stay. There is an underlying network that connects patients who share nurses and a separate network for patients who share doctors. A patient could be infected only if there is a source patient who shares a nurse or doctor. Patients who are connected by a nurse and a doctor have an increased probability of transmission if one of them becomes infected. A single parameter called virulence defines the probability of an infected agent transmitting the microorganism to a susceptible agent.

III. Model of the propagation by contact of nosocomial infection

As explained above, the model proposed in our research has the advantage of being developed based on a previous emergency department model and previous emergency simulator, both of which have been developed as part of previous research works [4][5][6] carried out with the collaboration of healthcare staff at the Emergency Department of Hospital Universitari Parc Taulí. This previous model used ABMS techniques to define the full attention process of an emergency department. Some agents defined in this model can be used as part of the model of propagation by contact, but we need to add more passive and active agents. In the same way, it is necessary to add new variables and behaviors in all agents, in order to represent MRSA transmission between a transmission vector and a susceptible host.

A. Transmition Forms

There are two forms of contact transmission, direct transmission and indirect transmission. In both cases we need a transmission vector.

1) Direct transmission: When MRSA bacteria is transmitted from an active agent (transmission vector) to another active agent (susceptible agent). For instance, when an infected/contaminated patient is touched by a member of healthcare staff without a physical barrier.

2) Indirect Transmission: When an active agent (transmission vector) touches medical equipment or objects in the hospital environment and MRSA bacteria is transmitted to the object, later, a susceptible agent (patient or healthcare staff) has contact with the same object and acquires the microorganism.

In the real process, for the first time to diagnose an MRSA infection or colonized , it is necessary to apply a laboratory test. Many laboratory test options exist but we mention two: culture testing and polymerase chain reaction (PCR). The culture testing is a conventional laboratory test, which has a low cost but the results are available in 72 or 96 hours (3 or 4 days). In contrast, PCR has a high cost but the results are available in a few hours (2-6 hours) and it may be more sensitive [14]. The time required to obtain results is an important factor in the attention process because the length of stay (LoS) of the patient in an emergency department is usually short. On the other hand, the long wait for the result of the laboratory test could contribute to increasing the rate of propagation of nosocomial infection.

To make a model about the indirect transmission, it is necessary to bear in mind the lifetime of MRSA on dead surfaces. There is scientific evidence that suggest that bacteria can live more than 90 days on different surfaces [15]. If we consider that the time of the attention process of the ED changes in function of the illness severity of the patient and availability of the healthcare services, in all cases it is less than 90 days, then we can assume in our model that the lifetime of MRSA bacteria is unlimited on dead surfaces (objects and medical equipment) but can be eliminated through a disinfection process carried out by cleaning staff.

B. Agents and Behaviors

Frequent interaction between patients and healthcare staff is the principal way to MRSA propagation. To combat MRSA, some healthcare services implement prevention polices to control transmission. These prevention policies are a set of concrete actions and behaviors that healthcare staff perform to control the rate of propagation. Some of these actions are: handwashing, use of hydroalcoholic solution and use of isolation material. Healthcare staff are required to practice the handwashing and use of hydroalcoholic solution every time

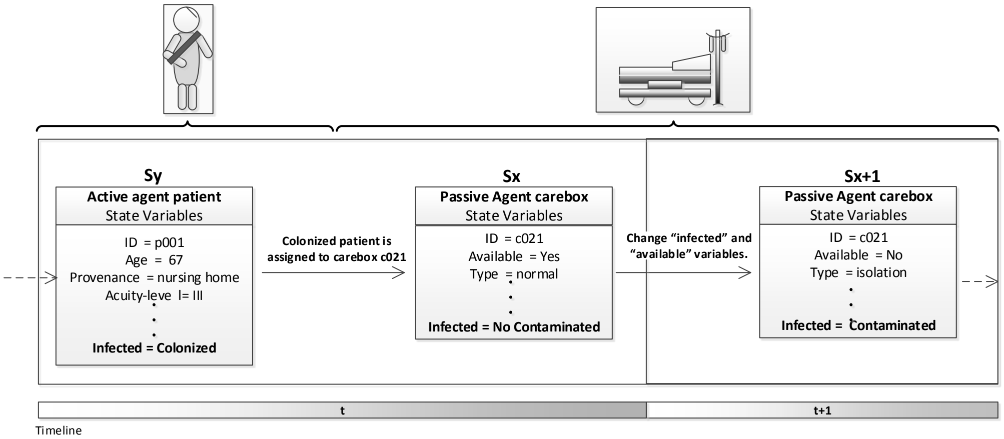


Fig. 1: Interaction between colonized patient and carebox.

that they attend a patient, but the use of isolation material is required only when they attend an isolated patient.

1) Passive Agent: Passive agents are agents that do not have their own initiative, they are solely reactive. All passive agents react to an action of an active agent or time. The actual model of ED has some passive agents such as information system. For our purpose, we added the passive agent "Carebox" in order to represent the interaction between active agents with the environmental objects.

Carebox: The carebox is the physical space where the patient is accommodated during the treatment and diagnosis process. In the model, once it has been confirmed that a patient is a transmission vector we have the possibility to isolate the patient in a carebox. All careboxes could be transformed in isolated careboxes. We use the infected variable to reflect whether a carebox is contaminated with MRSA or not. The possible value for infected are: *non-contaminated* or *contaminated*. When an MRSA patient is assigned to a carebox, the infected variable takes the value *contaminated* (see Figure 1), and it will remain at such a value until a disinfection process has been carried out. This process can only be executed when the MRSA patient leaves the emergency department and releases the carebox. The disinfection process is carried out by cleaning staff. The isolated carebox contains the entire equipment needed for patient care, which means that we have a lot of objects likely to be a vector of transmission, though all these objects are usually inside the same carebox, therefore we consider them as a single object. In extreme cases, where all carebox are busy, it is possible to use "a virtual carebox", which are additional spaces located in the ED. The main difference between a virtual box and a carebox is that the virtual one does not have the same level of isolation, which is why the probability of MRSA propagation is higher.

2) Active Agents: Any actor who has the ability to act by himself/herself is an active agent. The propagation model includes the interaction between patients and some agents of healthcare staff, such as doctors, triage nurses, nurses, admission staff, auxiliary personnel and cleaning staff (see Figure 2). For the purpose of our propagation model we can devide all these active agents in two cathegories: Patient and Healthcare Staff.

Patient: In order to properly define the propagation model of MRSA, it is important to consider that the patient on arriving to an emergency department, regardless of their acuity level, has a probability of being a transmission vector (infected or colonized) of MRSA. It means that the patient has a probability of being a carrier a microorganism causative to nosocomial infection. MRSA bacteria could live in the skin of some patients without their knowledge it and without any symptoms. However these patients could transmit the bacteria to another patients by physical contact.

In our model, we use the infected variable to show if the patient is *non-colonized*, *colonized* or *infected*. All non-colonized patients are susceptible to acquiring a nosocomial infection. We can classify patients in two groups: 1) patients that do not have a clear possibility of being colonized or infected with MRSA bacteria. In this case the healthcare staff assumes that this patient is non-colonized; 2) patients that are known or can be assumed to be colonized or infected with MRSA. If a patient has had a previous admission, the ED has their clinical history and knows if they had or have MRSA (during the admission process). In such cases the patient is classified as colonized, because a patient who has previously been colonized has a higher risk to a new colonization [16]. However, if it is the first time that the patient attends the emergency department, healthcare staff evaluate the MRSA Risk Factor (RF) of the patient.

In some health services, when it has been confirmed that a patient has MRSA (on the basis of the doctor's exploration and laboratory test results), this patient is immediately isolated in a carebox.

The MRSA risk factor is a metric that allows us to give the patient a level of probability (P(RF)) of being a transmission vector. With the purpose of calculating the risk factor, we can ask about several factors associated with a higher risk of acquisition of MRSA [17], but our research focused on three

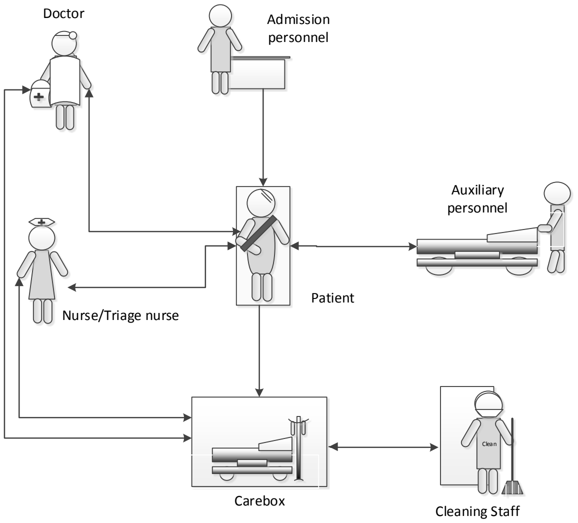


Fig. 2: Diagram of physical contacts of the patient with other actives and passives agents.

aspects that we consider require special attention:

- Patient age: patients over 65 years old have more probability of being an MRSA patient carrier because their immunological systems are usually weakened due to several illnesses or extended treatments.
- Place of residence: when patient is an institutionalized patient, sharing their residence with many people, such as a nursing home, prison, etc.
- Acuity level: usually patients with acuity level I, have more probability of acquiring a nosocomial infection because their immunological system is weakened.

$$\text{P(RF)}= f(\text{Age,PlaceResidence,AcuityLevel})$$

When the risk factor is high then we assume that there is a high probability that the patient be colonized. The patient can be considered infected only when the diagnostic can be supported by the results of laboratory tests. It is very important to ensure the results because if one positive case escapes, we have a transmission vector together with susceptible patients, and on the other hand, if one negative case is assigned as positive, the patient will receive an unnecessary treatment that will increase the cost.

It is very important to differentiate between infected and colonized patients. Colonized patients carry MRSA on their skin, but they are asymptomatic and therefore require laboratory tests for it to be detected. If MRSA bacteria get inside the body, the patient will develop an infection, becoming a symptomatic carrier, which is called an *infected patient*. However, the doctor needs to apply a physical examination and laboratory test to support the diagnosis. Both colonized and infected patients may transmit MRSA bacteria to susceptible patients and this can lead to infections with serious consequences.

Healthcare Staff: In this model we study the interactions with some active agents of healthcare staff: admission staff, triage nurses, patients, doctors, nurses, laboratory technicians, auxiliary personnel and cleaning staff. Each of these agents has a specific function in the attention process. The healthcare staff agent has the infected variable with three possible values: 1) *non-carrier*; 2) *carrier*; 3) and *colonized*. A member of the healthcare staff could be a *carrier* when they acquire temporarily bacteria. This happens when they have physical contact with a transmission vector and acquire the bacteria. Later, if the healthcare staff complies with the prevention policies, the bacteria will be eliminated and the healthcare agent will return to non-carrier. The accomplishment level of the healthcare staff agents with the prevention policies is measured by the *accomplishment factor* (AF).

This metric measures the probability that a member of the healthcare staff transmits the bacteria to a susceptible agent. This probability (P(AF)) will be calculated on the basis of the accomplishment level of prevention policies/actions. The three prevention actions that are evaluated in this research are:

- Handwashing: the most important of the actions. The healthcare staff is required to do it, before and after they have a physical contact with a patient.
- Sanitizing hand: with a hidroalcoholic solution. It is necessary to do it before and after they have physical contact with a patient. There is no substitute for handwashing and it is better if both actions are done.
- Using isolated material: this is required only when they attend an isolated patient.

$$\text{P(AF)}= f(\text{Handwashing,SanitizeHand,IsolatedMaterial})$$

C. Propagation of Nosocomial Infection by Contact Model

The model takes into account the parts of the overall process in which contact propagation can take place. For this it is necessary to clear the entire care process. This section is dedicated to describing the emergency attending process and identifying in which parts of this process a transmission vector can be in contact with a susceptible patient.

Attention Process: When a patient arrives in the emergency department they approach the admissions zone. Here the admissions staff ask for their health card and registers their arrival. Then the patient waits in the waiting room for the triage process. When the triage nurse is available, they call the patient and takes their vital signs and asking for some additional information in order to identify the acuity level of the patient. The acuity scale applied in Spanish ED considers five different values, from I until V, with I being the highest level, and V the lowest. If acuity level assigned to the patient is IV or V, they will wait for diagnosis and treatment process in a waiting room, but patients with acuity level I, II or III will be assigned immediately to a carebox and diagnosis and treatment phases should be done inside such a carebox, with the exception of some specific tests. The diagnosis and treatment process is divided in 3 phases: 1) laboratory test; 2) treatment; 3) Exit from ED. When a doctor is available, they call the patient and decide what the next step is. Laboratory test and treatment can be carried out several times.

TABLE II: POSSIBLE WAYS OF TRANSMISSION BETWEEN ACTIVE AGENTS.

Agent1	Agent2
Carrier healthcare staff	Susceptible patient
Carrier healthcare staff	Passive agent
Infected/colonized patient	Passive agent
Infected/colonized patient	Healthcare staff
Passive agent	Healthcare staff
Passive agent	Susceptible patient

When the treatment has finished and an additional laboratory test is not necessary, the doctor will prescribe that patient leaves the ED. In the case of patients IV and V, the interaction with the doctor will be carried out in attention boxes, and the patient will remain in a specific waiting room while there is no interaction (between each one of these phases). We focus on the stages in which agents have physical contact with other agents, that is, when patients have interaction with healthcare staff. The MRSA transmission is possible in one of several ways (Table II). If patient is a transmission vector and has contact with a susceptible agent, transmission is probable [17], in which case the susceptible agent will change their state. The variable ”infected” is used to reflect changes in the state of agents. To decide if one agent acquires the MRSA bacteria, we will use a probabilistic distribution based on the likelihood of the transmission vector transmitting the microorganism, and the probability of a susceptible patient acquiring the microorganism.

IV. Future Work and Conclusions

As a result of our research, we proposed an agents-based model of the contact propagation of MRSA in emergency departments. This model has been designed based on an emergency department simulator, ED-Simulator, developed in previous research and which has the advantage of having been verified and validated in several cycles or iterations, taking into account a wide variety of data and configurations, and with the participation of ED staff at the Hospital of Sabadell (Spain). Based on such an ED-Simulator and after a careful analysis of the care process, we have established in which parts of the process there is risk of infection and which agents have to be added in order to complete the propagation model. We have enhanced the model by including new agents, and by adding new variables and new behaviors in all the agents that participate in the transmission process. In this way, we have obtained a model of contact propagation of MRSA. Our future work is to implement the contact propagation model of MRSA in the ED-Simulator in order to obtain the computational model of the contact propagation. The next step will be the execution, validation and verification stages for improving the model. The computational model will allow the ED managers to analyze and evaluate potential solutions (the possible effects of applying the control policies' actions) for propagation of nosocomial infection in a virtual environment and to evaluate the effectiveness of different combinations of laboratory tests, isolation, and other control policies, with the purpose of identifying the best infection control policy.

Acknowledgment

This research has been supported by : MINECO (MICINN) Spain under contract TIN2011-24384. Ecuador government. SENESCYT, under contract 2013-AR7L335.

References

[1] N. Graves *et al.*, “Effect of HealthcareAcquired Infection on Length of Hospital Stay and Cost,” *Infection Control and Hospital Epidemiology*, vol. 28, no. 3, pp. 280–292, 2007.

[2] W. Hryniewicz, “Epidemiology of MRSA,” *Infection*, vol. 27, no. 2, pp. S13–S16, 1999.

[3] E. Bonabeau, “Agent-based modeling: Methods and techniques for simulating human systems,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. Suppl 3, pp. 7280–7287, 2002.

[4] M. Taboada, E. Cabrera, M. Iglesias, F. Epelde, and E. Luque, “An agent-based decision support system for hospitals emergency departments,” *Procedia Computer Science*, vol. 4, pp. 1870–1879, 2011.

[5] E. Cabrera, M. Taboada, M. L. Iglesias, F. Epelde, and E. Luque, “Simulation optimization for healthcare emergency departments,” *Procedia Computer Science*, vol. 9, pp. 1464–1473, 2012.

[6] M. Taboada, E. Cabrera, F. Epelde, M. Iglesias, and E. Luque, “Using an Agent-Based Simulation for predicting the effects of patients derivation policies in Emergency Departments,” *Procedia Computer Science*, vol. 18, pp. 641–650, 2013.

[7] R. Sanchez *et al.*, “El triaje en urgencias en los hospitales españoles,” *Emerging Infectious Diseases*, vol. 25, no. 1, pp. 74–76, 2013.

[8] R. Katz and G. Borriello, *Contemporary logic design*. Pearson Prentice Hall., New York, 2014, chapter 7, pp. 321–334, ISBN: 0201308576.

[9] S. Barnes and B. Golden, “Methicillin resistant Staphylococcus aureus transmission reduction using Agent-Based Modeling and Simulation,” *INFORMS Journal on Computing*, vol. 22, no. 4, pp. 635–646, 2010.

[10] L. Milazzo, J. Bown, A. Eberst, G. Phillips, and J. Crawford, “Modelling of Healthcare Associated Infections: A study on the dynamics of pathogen transmission by using an individual-based approach,” *Computer methods and programs in biomedicine*, vol. 104, no. 2, pp. 260–265, 2011.

[11] Y. Meng, R. Davies, K. Hardy, and P. Hawkey, “An application of agent-based simulation to the management of hospital-acquired infection,” *Journal of Simulation*, vol. 4, no. 1, pp. 60–67, 2010.

[12] J. Raboud *et al.*, “Modeling Transmission of MethicillinResistant Staphylococcus aureus Among Patients Admitted to a Hospital,” *Infection Control and Hospital Epidemiology*, vol. 26, no. 7, pp. 607–615, 2005.

[13] S. Barnes, B. Golden, and E. Wasil, “A dynamic patient network model of hospital-acquired infections,” in *Simulation Conference (WSC), Proceedings of the 2010 Winter*. IEEE, 2010, pp. 2249–2260.

[14] S. Van Hal, D. Stark, B. Lockwood, D. Marriott, and J. Harkness, “Methicillin-resistant Staphylococcus aureus (MRSA) detection: comparison of two molecular methods (IDI-MRSA PCR assay and GenoType MRSA Direct PCR assay) with three selective MRSA agars (MRSA ID, MRSASelect, and CHROMagar MRSA) for use with infection-control swabs,” *Journal of clinical microbiology*, vol. 45, no. 8, pp. 2486–2490, 2007.

[15] A. Neely and M. Maley, “Survival of enterococci and staphylococci on hospital fabrics and plastic,” *Journal of clinical microbiology*, vol. 38, no. 2, pp. 724–726, 2000.

[16] L. Coia, J. Alistair and J. Reilly, “Screening for meticillin resistant Staphylococcus aureus (MRSA): who, when, and how?” *British Medical Journal*, p. 348:g1697, 2014.

[17] C. Salgado, B. Farr, and D. Calfee, “Community-acquired methicillin-resistant Staphylococcus aureus: a meta-analysis of prevalence and risk factors,” *Clinical Infectious Diseases*, vol. 36, no. 2, pp. 131–139, 2003.

Impedance-based Higher Order Sliding Mode Control for Grasping and Manipulation

Rakibul Hasan, Ranjan Vepa, Hasan Shaheed
School of Engineering and Materials Science
Queen Mary, University of London, London, UK
{m.r.hasan,r.vepa,m.h.shaheed}@qmul.ac.uk

Abstract—**Grasping and manipulation by multifingered robot hand is challenging due to the dynamic uncertainties of the hand and the object. The purpose of the grasping and manipulation research is to improve the control performance based on the hand, object and contact model. In this paper, a multifingered hand called The Barrett Hand is chosen for investigation. A mathematical model of the hand is developed using Lagrangian method. Contact model between the hand and object is developed as compliance system assuming that the object stiffness properties are known. Grasping and manipulation control problem is stated as: 1. Object motion and 2. Contact force tracking. Higher Order Sliding Mode (HOSM) control is proposed for motion tracking of the hand joints and object. 2nd order sliding mode law is used to minimise the tracking error considering nonlinear dynamics and uncertainties. Adaptive Force-based Impedance Mode (AFIM) control is proposed to regulate the contact force which drives the desired impedance of the hand-object compliance system. This impedance also generates online reference trajectory for motion tracking according to the contact force. Both control methods are simulated in Simulink/SimMechanics environment. The joint, object and force tracking results are presented and discussed.**

Keywords–Grasping; Sliding; Impedance; Adaptive; AFIM; HOSM.

I. Introduction

Conventional robot manipulators used in industries are combined form of single arm attached to a gripper. The gripper is effective in performing tasks where the large motion of the payload required with fewer Degrees of Freedom (DOF). Traditional grippers are unable to show the efficiency where precise movement is needed for the object manipulation [2]. Grippers are being replaced now by the human hand alike multifingered hand to ease the grasping and manipulation activities by industrial manipulators.

Human hand grasps and manipulates an object by several steps: 1. The human eye works as sensor to locate the object location. 2. Hand moves to the object location with previous grasp planning knowledge. 3. When grasp is planned, fingers make contact with assumed forces, but change the force depending on the object's mass, stiffness and material properties. These properties are sensed by the brain to send the signals to the arm and hand to apply the optimized forces for grasping and manipulation of the object. The above procedures are not simple in robot hand grasping and manipulation. In an attempt to attain the identical abilities of the human hand, research has been going on for last two decades. In the early stage of the grasping research, Okada discussed the computer control method of multijoint finger system for precise object handling [11]. Mason and Salisbury formulated the grasp problem and introduced the grasp jacobian [10]. Kerr and Roth worked on grasp kinematics [8]. Bicchi and Okamura focused on grasping and manipulation of unknown objects [2][12]. Currently, several multifingered hands are available in the industries and also for the research. The robot hand is a complex system due to the presence of additional DOF for dexterity. In addition, this is not similar to the custom gripper which is efficient in generating a specific rigid grasp of an object. The same grasp with multifingered hand requires intelligent control and results in a decrease in grasp rigidity.

In this paper, two advanced robust control methods are applied for grasping and manipulation of an object by a robot hand. Higher Order Sliding Mode (HOSM) is implemented for position accuracy of the robot containing uncertainties and Adaptive Force based Impedance Method (AFIM) is applied to minimize the contact forces during manipulation. Both methods are advanced and powerful in achieving position and force control respectively in individual robotic tasks. Therefore, those are applied together in grasping and manipulation of the hand which is related to combined position and force problem. To implement and see the performances, a three fingered robot hand is modeled, contact model between the hand and the object is derived, control problem of grasping and manipulation is defined and robust adaptive controllers are designed according to the problem requirements. Contact locations and types are assumed to be known for simplifying grasp planning. The other assumptions made in studying the kinematics are that the fingers do not slip on the object, the object and finger links are rigid body, accurate models of the hand and object are known.

In next four subsections, grasp properties, robot and object kinematics and dynamics are derived. Section 2 explains the control theories behind HOSM and AFIM. The control laws implemented for the robot hand are also included here. Simulation of the proposed control and the results are presented in Section 3. The last section provided the summary of the work and future aspects.

A. Grasp and manipulation properties

A grasp is defined as a combination of hand posture and position that generates a contact force set between the hand and the grasped object. The grasp jacobian maps the relation between the object and the hand through this contact force. It is first proposed by Mason and Salisbury [10]. The popular Barrett hand is considered for analysing grasping and manipulation properties in this paper. It is used in industries for different tasks and available for research activities. Consider this three fingered hand grasping an object in Figure 1 [15]. Each finger has two DOF available for the grasping and manipulation tasks. Lets define the object coordinate frame

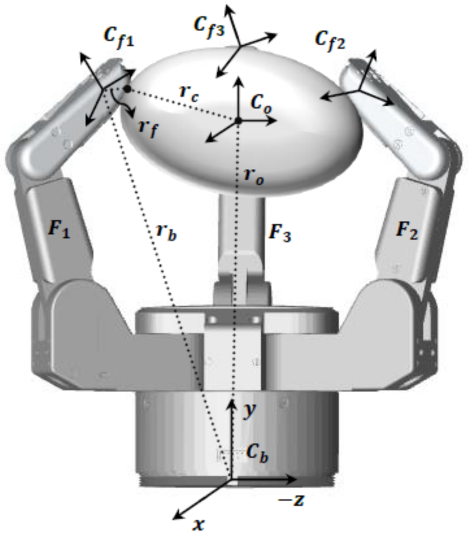


Figure 1. The Barrett Hand model grasping an object [15].

as C_o where center of mass of the object is chosen as the object reference point. C_b is the palm frame and $C_{f,i}$ is the contact coordinate (i= contact number). The term $r_o = [p_o, o_o]$ is the position vector of the origin of C_o with respect to C_b where p_o is the position and o_o is the orientation vector of the object origin. The term $r_b = f(q_i)$ is the position vector of the origin of $C_{f,i}$ with respect to C_b. The term r_c is the position vector from the origin of C_o to contact coordinate $C_{f,i}$ with respect to C_b . All contacts between the object and hand are considered as point contact without slip. If each finger applies a force $f_{c,i}$ at the contact point $C_{f,i}$, the applied object wrench is given as,

$$\sum_{i=1}^{n} G_i F_{c,i} \tag{1}$$

where, $F_o = [f_o^T \quad m_o^T]^T$ is the external force exerted on the object and $G_i \in \Re^{6\times n}$ is called the grasp Jacobian matrix which transforms the contact force $f_{c,i}$ to the object coordinate frame (n is the number of contacts) and is given by,

$$G_i = \begin{bmatrix} I_3 \\ r_c\times \end{bmatrix} \tag{2}$$

where, I_3 is the 3×3 identity matrix and $[r_c\times] \in \Re^{3\times 3}$ is a skew symmetric matrix determined by the cross product of r_c. The full grasp matrix can be shown as $G = [\ G_1 \quad G_2 \quad \dots \quad G_n\ ]$ and the contact force set,$f_c = [\ f_{c1}^T \quad f_{c2}^T \quad \dots \quad f_{cn}^T]^T$. The contact force $f_{c,i} \in FC$ where, $FC = \{f \in \Re^3 \left\{ f \in \Re^3 : \sqrt{f_1^2 + f_2^2} \leq \mu f_3 \right\}$ is called the friction cone, which is the angle of the cone with respect to contact normal [10][13]. Similarly, the matrix which transforms the contact force $f_{c,i}$ to the finger coordinate frame is given by,

$$F_i = A_i f_{c,i} \tag{3}$$

where, $A_i \in \Re^{6\times n}$ and have the following matrix form,

$$A_i = \begin{bmatrix} I_3 \\ r_f\times \end{bmatrix} \tag{4}$$

the term, $[r_f\times] \in \Re^{3\times 3}$ is the cross product of r_f. Now, if the i-th finger of the hand manipulates an object; the velocity relation $\dot{r}_{o,i} = \dot{r}_{f,i}$ is valid where, $\dot{r}_{o,i}$ is the object and $\dot{r}_{f,i}$ is the fingertip velocity at i-th contact point and both are equal, assuming there is no slip during contact. Considering the transformation with respect to base frame, the velocity relation can be expressed as,

$$G_i{}^T \dot{x}_o = J_{hi}\dot{q}_i \tag{5}$$

Equation (5) is the fundamental grasp constraint which governs the grasp and manipulation of an object by contact force related to the hand and the object velocity. The term $\dot{x}_o = \begin{bmatrix} \dot{r}_o^T & \omega_o^T \end{bmatrix}^T$ is the velocity vectors of the object and ω_o is the angular velocity of the object with respect to the base frame C_b. Now define the jacobian form in the finger space as, $J_{hi} \in \Re^{6\times l} = A_i{}^T J_i$ as the kinematic jacobian at the i-th finger; where l_i are the number of links for fingers and J_i is the kinematics jacobian of the i-th finger. The orientation vector, $\zeta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ of the object is represented by *Z-Y-Z* Euler angles. Based on Euler transformation, the relation between object velocity $\dot{x}_o$ and $\dot{r}_o$ can be shown as,

$$\dot{x}_o = T_o(\xi)\dot{r}_o \tag{6}$$

where, $T_o(\xi) \in \Re^{6\times 6}$ is a transformation matrix between two velocity vectors given as,

$$T_o(\zeta) = \begin{bmatrix} I_3 & 0 \\ 0 & \prod(\xi) \end{bmatrix} \tag{7}$$

and $\prod(\xi)$ is represented by the following transformation matrix,

$$\prod(\xi) = \begin{bmatrix} -\sin\theta\cos\psi & \sin\psi & 0 \\ \sin\theta\sin\psi & \cos\psi & 0 \\ \cos\theta & 0 & 1 \end{bmatrix} \tag{8}$$

B. Object dynamics

Using (8) - (10), the object dynamics can be expressed by Newton-Euler equation in following form:

$$\begin{bmatrix} m_o I & 0 \\ 0 & I_b \end{bmatrix} \begin{bmatrix} \ddot{x}_o \\ \dot{\omega}_o \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_o \times I_b\omega_o \end{bmatrix} = \begin{bmatrix} f_o \\ \tau_o \end{bmatrix} \tag{9}$$

where, m_o is the mass of the object and $I_b = \Pi(\xi)$. Let X_o be the local parametrization of x_o to transform (9) in local coordinate as,

$$M_o\ddot{X} + C_o(X, \dot{X})\dot{X} = F_o \tag{10}$$

where, $M_o \in \Re^{6\times 6}$ is the object mass matrix, and the object coriolis terms are expressed as $C_o(X, \dot{X})$. The resultant force exerted on the object is $F_o = [f_o^T \quad \tau_o^T]^T$. Now, applying (1), (10) is rearranged as,

$$M_o\ddot{X} + C_o(X, \dot{X})\dot{X} = GF_c + f_e \tag{11}$$

Equation (11) is the object dynamics where contact force F_c is related to the object motion and f_e occurs due to external forces. Recall (5) and rewrite as follows:

$$\dot{q} = (A^T J)^{-1} G^T \dot{X}_o \tag{12}$$

Differentiating (12) yields,

$$\ddot{q} = (A^T J)^{-1} G^T \ddot{X}_o + \frac{d}{dt}((A^T J)^{-1} G^T)\dot{X}_o \tag{13}$$

where, $\ddot{q}$ is the joint acceleration related to the object motion.

1) Robot hand Kinematics and Dynamics: The robot hand dynamics is derived from the hand kinematics. According to the Figure 1, the Barrett hand has 3 fingers, each finger has 2 links. All fingers are identical in shape and dimension, but finger F_1 and F_2 have extra DOF. Let $(x, y, z)_{F,i}$ be the coordinates of the fingers and the forward kinematics of the fingers $F_i\,(i = 1, 2)$ are,

$$\begin{gathered} x_{F,i} = (l_1 \cos(q_1 + q_{i_1}) + l_2 \cos(q_1 + q_2 + q_{i_1} + q_{i_2})) \sin(\phi) \\ y_{F,i} = l_1 \sin(q_1 + q_{i_1}) + l_2 \sin(q_1 + q_2 + q_{i_1} + q_{i_2}) \\ z_{F,i} = (l_1 \cos(q_1 + q_{i_1}) + l_2 \cos(q_1 + q_2 + q_{i_1} + q_{i_2})) \cos(\phi) \end{gathered} \tag{14}$$

and the forward kinematics of the finger F_3 are,

$$\begin{gathered} y_{F,3} = l_1 \sin(q_1 + q_{i_1}) + l_2 \sin(q_1 + q_2 + q_{i_1} + q_{i_2}) \\ z_{F,3} = (l_1 \cos(q_1 + q_{i_1}) + l_2 \cos(q_1 + q_2 + q_{i_1} + q_{i_2})) \end{gathered} \tag{15}$$

where, $q_{i1} = 2.46^\circ$ and $q_{i2} = 40^\circ$. For Cartesian coordinate position (x_c, y_c, z_c), the joint angles of the fingers can be calculated with inverse kinematics using simple algebra and trigonometry [10].

Robot dynamics is determined by the Lagrangian method. The Lagrangian L is determined as the difference of kinetic (T) and potential (V) energy of the system [10][16]. Now, differentiate the Lagrangian L to derive the equations of motion of the system from the following term,

$$T_i = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \tag{16}$$

where, i=1,2,3.....n is considered as DOF of the system. Robot finger dynamics are derived using (16) as,

$$M_{f_i}\ddot{q} + C_{f_i}(q, \dot{q})\dot{q} + N_{f_i}(q) = \tau_{f_i} \tag{17}$$

where, $M_{f_i} \in \Re^{3\times3}$ is the hand mass matrix, $C_{f_i}(q, \dot{q}) \in \Re^{3\times3}$ is the vector of the coriolis and centrifugal terms and $N_{f_i} \in \Re^{3\times1}$ are the gravity terms for the fingers F_1 and F_2. For the finger F_3, $M_{f_3} \in \Re^{2\times2}, C_{f_3}(q, \dot{q}) \in \Re^{2\times2}$ and $N_{f_3} \in \Re^{2\times1}$. The finger joint acceleration $\ddot{q}$ is calculated from (17) as,

$$\ddot{q} = {M_f}^{-1}\left(\tau_f - C_f(q, \dot{q}) - N_f(q)\right) \tag{18}$$

Substitute, (18) into (13), the robot-object dynamics at object frame is written in the following form,

$$M_{of}\ddot{q} + C_{of}(q, \dot{q}) + G_{of}(q) = \tau_{OF} \tag{19}$$

where, M_{of}, C_{of} and G_{of} are combined robot-object terms and have the following expression after rearranging,

$$\begin{gathered} M_{of} = M_o + GJ_h^{-T}M_f(q)J_h^{-1}G^T \\ C_{of} = C_o + GJ_h^{-T}\left(C_f(q, \dot{q})J_h^{-1}G^T + M_f(q)\right)\tfrac{d}{dt}(J_h^{-1}G^T) \\ G_{of} = GJ_h^{-T}N_f(q) \end{gathered} \tag{20}$$

2) Grasp Problem: Control problems of grasping and manipulation by a robot hand is defined following (19) as,

Object Tracking: Given a desired object trajectory $X_{do} \in \Re^6$ the center of mass of the object $X_o \in \Re^6$ should track it while the object is grasped by three fingered robot hand considering grasp constraint. The first control objective is,

$$\lim_{t\to\infty} X_{do} - X_o = 0 \tag{21}$$

Contact Force Tracking: Given a desired contact force $F_d \in \Re^9$ the actual force $F_c \in \Re^9$ should follow the desired force. The second objective is,

$$\lim_{t\to\infty} F_d - F_c = 0 \tag{22}$$

Object tracking problem is also associated with the robot finger tracking problem. Robot fingers are tracked for desired contact location before the object is grasped. Following the properties of the friction cone, it can be shown that for an arbitrary set of contact force F_c, an internal force F_I is found such that $F_c + F_I$ lies in the friction cone. This internal force does not have any net effect on the motion of the hand or object.

II. CONTROLLER DESIGN

Different control methods are available for solving the control problem defined above. Classical and adaptive sliding based controllers were used in the past for object tracking solution [1][16]. Recently Higher Order Sliding Methods (HOSM) showed robustness for perfect tracking considering uncertainties [4][5]. For force tracking problem, impedance controllers by Hogan have been used in robotics for last two decades [6]. Based on Hogan, Seraji and Jungs work, Adaptive Force based Impedance Method (AFIM) is used for force tracking [6][7][13]. In this paper, the HOSM is used for tracking control. The AFIM is used for optimizing the contact force. The control solution will be divided into three steps:

1. Given a desired contact location, HOSM is used for tracking the current position of the finger before grasp. 2. Considering grasp constraint, AFIM is implemented to optimize the contact force 3. The object trajectory is tracked with HOSM based on the contact force.

A. Higher Order Sliding Mode Control (HOSM)

The robot hand contains dynamic nonlinearities and uncertainties. Joint tracking is not simple considering uncertain dynamics due to unmodeled mechanics, external disturbances etc. The controller robustness is important to overcome this dynamic uncertainties and achieve good tracking. HOSM is a robust controller, which deals greatly with disturbances and uncertainties. It is an extended version of Sliding Mode control (SMC) [9]. Lets introduce SMC applied by a sliding surface with high frequency switching control. The goal of SMC is to converge the actual system to the desired state. In general, SMC has chattering problem due to the discontinuous nature of control. It excites unexpected high frequency dynamics which may cause damage to the system. HOSM is effective in reducing chattering problem that also guarantees the convergence of system state in finite time. In this method, the sliding variable is extended to accelerate the system state towards the selected sliding surface. Considering the tracking control problem, the general SMC is defined as,

$$s(q, t) = \dot{e} + \lambda e \tag{23}$$

where, $e_q = q - q_d$ is the finger tracking error and $e_o = X_o - X_d$ is the object tracking error and λ is a constant whose eigenvalues are always positive. The control goal is achieved by choosing the control input such that the sliding surface fulfills the following condition:

$$\frac{1}{2}\frac{d}{dt}s^2 \leq \eta\,|s| \tag{24}$$

where, η is a positive constant. The above condition states that the system energy decays until the system state reaches the sliding surface. In HOSM, the discontinuous control input is applied to a higher time derivative of the sliding surface. The applications of HOSM in non-linear systems are found in [4][5]. Levant proposed a method of arbitrary order sliding mode controller design is discussed [9]. Defoort developed a robust HOSM controller for finite time stability [4]. For specifying HOSM, consider the non-linear system as,

$$\begin{aligned} \dot{x} &= f(x,t) + g(x,t) \\ y &= s(x,t) \end{aligned} \tag{25}$$

The relative degree r of system (24) is assumed to be known with respect to the sliding variable s. The control goal is to reach s=0 in finite time and hold by discontinuous function of s and its derivatives $s, \dot{s}, \ddot{s} \cdots\cdots\cdots s^{r-1}$. The r-th order sliding mode is expressed as,

$$S^r = \{u \parallel s, \dot{s}, \ddot{s} \cdots\cdots\cdots s^{r-1} = 0\} \tag{26}$$

The mode $s = 0$ is achieved after finite time transient. The control input appears at the *r-th* order sliding set as,

$$S^r = h(x,t) + g(x,t) \tag{27}$$

where $h(x,t) = s^r|_{u=0}$, $g(x,t) = \frac{\partial}{\partial u}s^r \neq 0$. From HOSM structure of (26) and (27), the controller u_h is chosen as,

$$u_h = \alpha sign\left(\frac{\dot{s}_h + \sqrt{|\,s_h\,|}sign(s_h)}{\dot{s}_h + \sqrt{|\,s_h\,|}}\right) \tag{28}$$

where, $h = f/o$, (f for finger and o for object). Equation (28) is the HOSM law proposed for finger and object tracking. The sliding variable s_h is expressed from tracking error e_q and e_o as,

$$s_h = \dot{e}_h + \lambda_h e_h \tag{29}$$

The control input torque for trajectory tracking is expressed as a combination of SMC and HOSM below,

$$u_{hosm} = u_c + u_h \tag{30}$$

where, input u_c is expressed as,

$$u_c = \overline{u} - Ks_c = M\ddot{q}_r + C\dot{q}_r + G - Ks_c \tag{31}$$

The term K is the sliding gain and the sliding function s_c is considered from (23).

B. Adaptive Force based Impedance Control

For grasping and manipulation, the important task is to maintain the contact force when the object is grasped. Position and force cannot be maintained at the same time, but the contact force can be regulated by controlling the impedance of the hand. Impedance methods have high impact over position and contact force control.The impedance problem is defined as proposing a single controller with estimated contact force exerted by the object on the robot hand that regulates motion trajectory and contact force with or without knowledge of object stiffness. It was first introduced by Hogan [3]. The general impedance law is defined as,

$$F_d - F_c = M_I(\ddot{X}_d - \ddot{X}_r) + B_I(\dot{X}_d - \dot{X}_r) + K_I(X_d - X_r) \tag{32}$$

In (32), force error $e_f = F_d - F_c$ drives the mechanical impedance properties of the object where,X_r is the regulated object position trajectory driven by $e_f = F_d - F_c$. To calculate the resultant force exerted by object, substitute (32) into object dynamics of (10) yields,

$$F_o = (M_o - M_I)\ddot{e} + B_I\dot{e} + K_I e + C_o(X_o, \dot{X}_o) \tag{33}$$

The impedance error should tend to zero and can be expressed mathematically,

$$\lim_{t\to\infty} X_d - X_r \approx 0 \tag{34}$$

The resultant force found from (33) is to realize the desired coupled compliance of (32). The contact force F_c can be calculated from (11) as,

$$F_c = G^{+}F_o + F_I \tag{35}$$

where, F_I is the internal force. Reference trajectory is then regulated from (33) for tracking control by HOSM. The integration of adaptation mainly focuses on generating desired trajectory online as a function of force tracking error e_f. When contact force is generated, the online based desired object trajectory is defined as,

$$X_d(t) = f(t) + k_p(t)e_F(t) + k_d(t)\dot{e}_F(t) \tag{36}$$

where,$f(t)$ is the auxiliary input, generated by the adaptation scheme. $k_p(t)$ and $k_v(t)$ are adaptive proportional gains driving force error and error rate. These are determined by the P-I adaptation laws below:

$$\begin{aligned} a(t) &= a(0) + \alpha_1\int_0^t q(t)dt + \alpha_2 q(t) \\ k_p(t) &= k_p(0) + \beta_1\int_0^t q(t)e_F(t)dt + \beta_2 q(t)e_F(t) \\ k_d(t) &= k_d(0) + \gamma_1\int_0^t q(t)\dot{e}_F(t)dt + \gamma_2 q(t)\dot{e}_F(t) \\ q(t) &= w_p e_F(t) + w_d\dot{e}_F(t) \end{aligned} \tag{37}$$

where, w_p and w_d are weighting factors object position and velocity. The terms $(\alpha_1, \beta_1, \gamma_1)$ are integral adaptation gains, and $(\alpha_2, \beta_2, \gamma_2)$ are proportional adaptation gains. The online based trajectory X_d is calculated from (36) and (37) where $X_d(0) = a(0)$. This online based X_d is fed by the impedance filter (32) to extract regulated trajectory X_r. The regulated X_r is then tracked by HOSM control method. Two control methods applied to the multifingered hand is shown in Figure 2.

III. SIMULATION AND RESULTS

For the Barrett hand simulation, link lengths, $l_1 = 0.05m, l_2 = 0.07m, l_3 = 0.07m$, link mass, $m_1 = 0.03kg, m_2 = 0.05kg, m_3 = 0.03kg$, base height of the hand, $h = 0.05m$, gravity, $g = 9.81m/s^2$. The HOSM parameters before grasp: sliding constant, $\lambda = 0.04$, switching gain, $K_h = 18$, gain, $\alpha = 1.8$. After grasp, HOSM control parameters in object space are: Sliding constant, $\lambda = 0.01$, Switching gain, $K_h = 12$, gain, $\alpha_h = 1.1$. AFIM mode is dependent on the contact force. Desired force F_d is chosen as $10\,N$. Impedance parameters are $M_I = 23, B_I = 70$ and

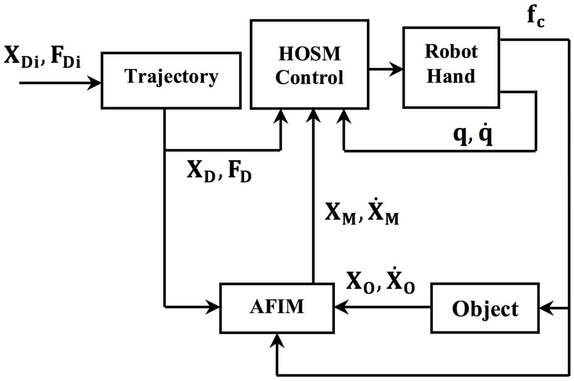


Figure 2. Proposed control architecture of HOSM-AFIM.

$K_I = 450$. The adaptive parameters are: k_p, k_d, w_p and w_d are selected as $1.4, 0.041, 0.02$ and 0.001 respectively.

At first, fingers are simulated in joint space to see the tracking performance with HOSM. Desired trajectories of the fingers are given for both mode as $[sp, q_2, q_3] = [0.7854, 0.6109, 0.4363]$ rad. In Figure 3 and 4, joint motion of fingers are compared between HOSM and general computed torque method (CTC). It appears that the overshoots are less and settling down is quick with HOSM considering uncertainties. Trajectories are well tracked with disturbances given as additional input into the system. In Figure 5, control torques of finger F_3 are shown with SMC (Top) and HOSM (Bottom). The input signal chattering is visible with SMC, common in first order sliding mode. The chatterings are completely eliminated with HOSM which indicates that HOSM is not only able to achieve precise tracking performance but also provides smooth torque to drive the joints. Then, the system is simulated to assess the performance of object manipulation after grasp. The object position and orientation errors are shown in Figure 6 and 7 respectively. The HOSM based manipulation errors are compared with CTC. The error are large with CTC method and not converging to zero for few motion vectors. The motion errors are improved and converged to zero with HOSM. Figure 8 shows the tracked normal contact forces of all fingers for desired force input of $10\ N$. The results are compared with natural impedance filter. The steady state errors are found with impedance law. Overshoots are reduced and steady state errors are minimized with AFIM mode.

IV. CONCLUSION

Two adaptive and robust control methods are proposed for grasping and manipulation of an object by multifingered robot hand. Sliding mode based HOSM control is proposed due to the high precision tracking requirements for performing manipulation tasks. Joint and object motion tracking results indicate the authenticity of the HOSM control algorithm. The AFIM process is efficient in generating a grasp of an object with a desired force estimated by object stiffness properties. Addition of adaptation helped to achieve the desired force. This method ensures not to cause any damage to the object as the fingers only exert the desired force to grasp and manipulate the object. Control objectives are successfully achieved by the proposed methods and the simulated results.

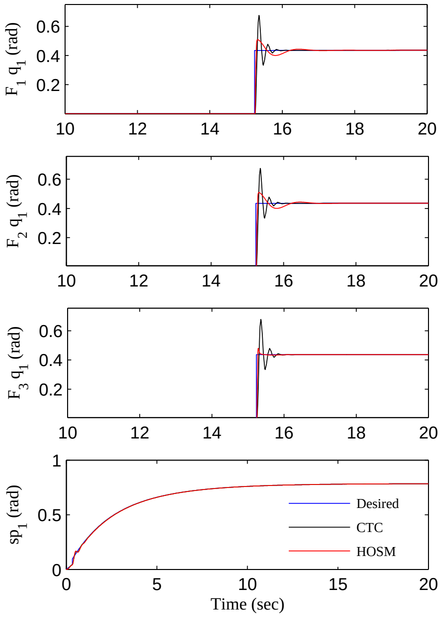


Figure 3. First revolute joint q_2 and sp_1 results of all fingers.

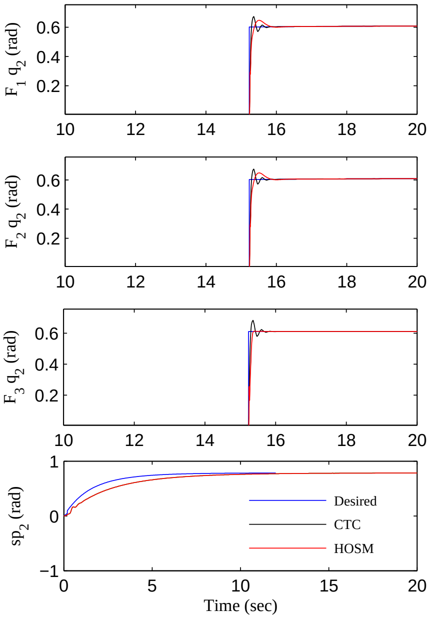


Figure 4. Second revolute joint q_3 and sp_2 results of all fingers.

REFERENCES

[1] S. Arimoto, R. Ozawa, and M. Yoshida, “Two-dimensional stable blind grasping under the gravity effect,” in Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE, 2005, pp. 1196–1202. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1570278

[2] A. Bicchi and V. Kumar, “Robotic grasping and contact: A review,” in ICRA. Citeseer, 2000, pp. 348–353.

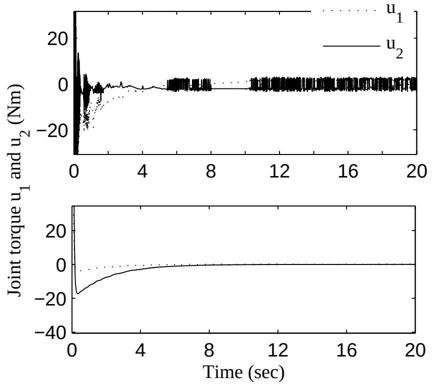


Figure 5. Chattering elimination of joint torque τ with HOSM for finger F_3; SMC (Top) HOSM (Bottom).

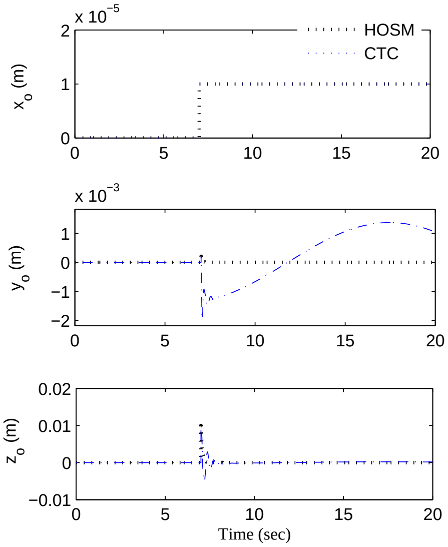


Figure 6. Object position error comparison with HOSM and CTC law.

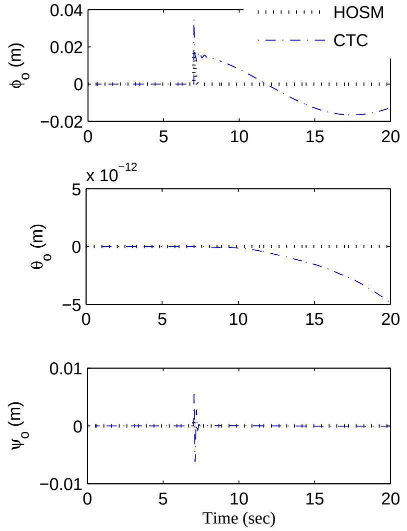


Figure 7. Object orientation error comparison with HOSM and CTC law.

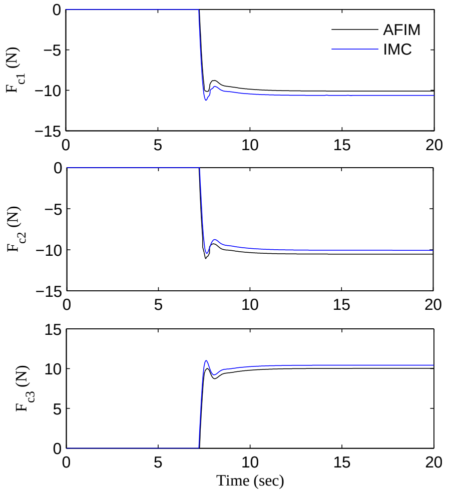


Figure 8. Contact force comparison with IMC and AFIM law.

[3] A. B. Cole, J. E. Hauser, and S. S. Sastry, “Kinematics and control of multifingered hands with rolling contact,” Automatic Control, IEEE Transactions on, vol. 34, no. 4, 1989, pp. 398–404. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=28014

[4] M. Defoort, T. Floquet, A. Kokosy, and W. Perruquetti, “A novel higher order sliding mode control scheme,” Systems & Control Letters, vol. 58, no. 2, 2009, pp. 102–108. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167691108001588

[5] F. Dinuzzo and A. Ferrara, “Higher order sliding mode controllers with optimal reaching,” Automatic Control, IEEE Transactions on, vol. 54, no. 9, 2009, pp. 2126–2136. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5208186

[6] N. Hogan, “Impedance control: An approach to manipulation: Part ithe-ory, part ii-implementation, part iii-applications,” Journal of Dynamic Systems, Measurement, and Control, 1985, p. 1.

[7] S. Jung, T. C. Hsia, and R. G. Bonitz, “Force tracking impedance control for robot manipulators with an unknown environment: theory, simulation, and experiment,” The International Journal of Robotics Research, vol. 20, no. 9, 2001, pp. 765–774. [Online]. Available: http://ijr.sagepub.com/content/20/9/765.short

[8] J. Kerr and B. Roth, “Analysis of multifingered hands,” The International Journal of Robotics Research, vol. 4, no. 4, 1986, pp. 3–17. [Online]. Available: http://ijr.sagepub.com/content/4/4/3.short

[9] A. Levant, “Higher-order sliding modes, differentiation and output-feedback control,” International Journal of control, vol. 76, no. 9-10, 2003, pp. 924–941. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/0020717031000099029

[10] M. T. Mason and J. K. Salisbury Jr, Robot hands and the mechanics of manipulation. MIT press, 1985.

[11] T. Okada, “Computer control of multijointed finger system for precise object-handling,” Systems, Man and Cybernetics, IEEE Transactions on, vol. 12, no. 3, 1982, pp. 289–299. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4308818

[12] A. M. Okamura, N. Smaby, and M. R. Cutkosky, “An overview of dexterous manipulation,” in Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on, vol. 1. IEEE, 2000, pp. 255–262. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=844067

[13] H. Seraji and R. Colbaugh, “Force tracking in impedance control,” The

International Journal of Robotics Research, vol. 16, no. 1, 1997, pp. 97–117. [Online]. Available: http://ijr.sagepub.com/content/16/1/97.short

[14] C.-Y. Su and Y. Stepanenko, “Adaptive sliding mode coordinated control of multiple robot arms attached to a constrained object,” Systems, Man and Cybernetics, IEEE Transactions on, vol. 25, no. 5, 1995, pp. 871–878. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=376500

[15] W. Townsend, “The barretthand grasper–programmably flexible part handling and assembly,” Industrial Robot: An International Journal, vol. 27, no. 3, 2000, pp. 181–188. [Online]. Available: http://www.emeraldinsight.com/journals.htm?articleid=875164&amp;show=abstract

[16] T. Yoshikawa and K. Nagai, “Manipulating and grasping forces in manipulation by multifingered robot hands,” Robotics and Automation, IEEE Transactions on, vol. 7, no. 1, 1991, pp. 67–77. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=68071

Towards a New Alternative to Assess the Validity of Driving Simulators: The Concept of Presence

Christophe Deniaud, Vincent Honnet, Benoit Jeanne
DRD/DRIA
PSA Peugeot Citroën
Vélizy-Villacoublay, France
christophe.deniaud@mpsa.com;
vincent.honnet@mpsa.com;
benoit.jeanne@mpsa.com

Daniel Mestre
CNRS, ISM, UMR 7287
Aix-Marseille University
Marseille, France
daniel.mestre@univ-amu.fr

***Abstract*— In this paper, we propose a new approach of the behavioral validity's assessment of driving simulators. Our ambition is to find a way of measuring "presence" to use it as a measure for ecological validity in driving simulators. The underlying assumption is that a person experiencing a strong sense of presence in the virtual environment will react in this environment as if it would be a real one. We propose to measure "presence" by measuring "attention" toward the driving task". Our objective is to demonstrate that the higher the subject's attention required by the primary driving task will be, the more the spatial presence will be felt. In the experiment, we tried to vary "attention" by adding a dual task and by adding traffic and measure driving performance and subjective "presence" (MEC-SPQ: Measurement, Effects, Conditions-Spatial Presence Questionnaire). The main result is a lack of congruence between subjective and behavioral measures.**

Keywords-Simulation; Attention; Presence; Driving Simulator.

I. INTRODUCTION

In the 1960s, driving simulation was mainly used to train specific target audiences such as novice drivers, law enforcement officers and truck drivers [3]. Since then, many advances have been achieved in terms of computing, visual display and vehicle dynamics rendering [2]. Driving simulation was originally developed to avoid cost of field studies, allowing more control over circumstances and measurements, and ensuring safety in hazardous conditions [1]. In the second half of the twentieth century, simulation was being successfully applied to aeronautical, rail and maritime operations. In spite of significant differences, it is interesting to note that the development of driving simulation was based on the development of flight simulation. Driving is a dynamic task with a set of rapid control maneuvers involving critical feedback for avoiding obstacles and preventing crashes [10]. Compared to the activity performed by air line pilots, driving involves higher amplitude, and higher frequency cues. The motion feedback does not play a key role for the major part of slow maneuvers performed by civilian pilots. There is no evidence that motion base-simulators are more efficient than fixed base simulators for training of commercial pilots [6]. Thus, there is a stark contrast between driving simulation and flight simulation (Civil aviation only). Compared to an airline pilot, a driver needs a higher degree of motion simulation. That is probably the reason why the use of flight simulators is more than commonplace for pilot training and, conversely driving simulators are not widely used for driver training due to the inherent higher complexity of the driving task.

Nowadays, driving simulators are usually designed for two purposes: research and training. The simulator is essentially used to place constraints on driver behavior in order to study driver distraction and workload or used as test beds for highway design [15]. The use of a modern advanced driving simulator for human factors research has many advantages such as experimental control, efficiency, expense, safety, and ease of data collection [20]. However, the literature describes some possible disadvantages, i.e., simulator sickness, accurate replication of physical sensations, and most importantly, validity.

In this paper, we will first look at a brief overview about the ecological validity to identify which specific issues still need to be addressed in driving simulation. Then a methodological alternative will be exposed through the concept of Presence. The remainder of the report basically consists of exposing results about this new experimental approach.

II. ECOLOGICAL VALIDITY

In spite of significant advancements in the physical fidelity of the driving simulation, a lack of realism seems to be always observed in the major part of driving simulator studies [4]. The most important question is to know in which extent measures from simulation are similar to those obtained in the real world. This multidimensional problem is called simulation validation [5]. This question has been a concern for at least 25 years. Blaauw [5] defined two types of validity. The first one is the absolute validity; it deals with the extent to which a manipulation of a variable in the real world produces the same or equivalent change in the same measure when manipulated in a driving simulator. The second one is called the relative validity and refers to the extent to which the direction of change of a variable is in the same direction as a corresponding manipulation and measure in the real world [16]. If absolute validity is obviously desired by researchers, regarding the variability of driver performance, it seems highly unlikely to have an exact correspondence of on-road and simulation measures. Furthermore, there is no bad or good simulator from a methodological point of view. The simulation validation seems to be arbitrated between the research issue and how simulators are used to investigate this question.

Each simulator must be validated for a specific use. In addition, the question of simulation validation has followed the perpetual development of a significant number of simulator components as computers and various display technologies. That is the reason why, since four decades, simulators have been designed to deliver more and more perceptual cues to the driver in order to reproduce as accurately as possible the experience of driving an automobile. Thus, simulator validity is often addressed in the extent to which a physical variable in a simulator corresponds to its operationally equivalent component in the real world is called Physical fidelity [19]. As previously discussed, simulation validity is multidimensional and can be related to behavioral and physical dimensions [12], but also to the perceived sensation of the subjective experience and objective performance [30]. Indeed, despite significant advancements in the fidelity of the driving experience driving simulator studies continue to be criticized for lack of realism [9]. More specifically, the physical fidelity of the driving experience appears insufficient to overcome criticisms concerning the lack of psychological fidelity [8], defined as the extent to which the risks and rewards of participation in the experiment correspond to real-world risks and rewards [21]. The main problem is that driving experimental studies failed to provide a non-artificial trip purpose which could be able to reproduce drivers' motives inherent to the real driving activity.

Generally, it appears that the assessment of the validity of the virtual environment involves the comparison of results obtained from studies conducted in real situations and in virtual environment. However, this comparison is expensive (instrumentation) and complex (strict control of all the events occurring in a real situation). This is probably the reason why questions about the validity of simulators are most often pending [23] and why only few studies on this question can be found.

III. A METHODOLOGICAL ALTERNATIVE: THE CONCEPT OF PRESENCE

Driving simulation is a historical component of virtual reality, the purpose of which being to enable one person (or more) to develop sensori-motor and cognitive activity in an artificial world [17]. The interaction of a person with the virtual world is a transposition of the perception-cognition-action loop of human behaviour in the real world. Immersion in a virtual world cannot be the same as in the real world [11], since the user has learned to act naturally in a real and physical world (without, for instance, any delay and/or sensorimotor bias). Thus, immersion, depending on the sensorimotor contingencies permitted by the simulator, is a necessary but not sufficient condition for the expression, within the virtual environment, of a performance that is representative of the actual situation [18]. Facing this problem, a concept emerged in the '80s from the early steps of research about virtual reality. This concept addresses the issue of "ecological" validity of the behaviour observed in virtual environments. It is the concept of presence. This multidimensional concept is considered as the ability of individuals to adopt behavioural patterns similar to that observed in everyday life and therefore as their propensity to respond to various stimuli by a realistic way [26].

Kaptein et al. [14] and Tichon [28] already proposed this concept as a tool for assessing driving or railway simulators but only in driving situations generating a state of stress. In studies about presence, finding a consensus about presence conceptualization in order to enhance its operationalization and its assessment [27] seems to be the main challenge. Various attempts have been made to describe this concept. Despite divergences, the major part of publications considers that presence rests on an attentional basis [22][24][25][31].

Thus, we decided to modulate experimentally the cognitive load induced during the simulated driving task, in order to generate different attentional states and finally to induce different levels of spatial presence. We crossed the two following independent variables: 1) A secondary task (dual-task paradigm), supposed to distract the driver from the primary task (driving) and 2) the presence of traffic on the roadway, supposed to focus the driver's attention toward the primary task. Our objective was to develop a sensitive measure of presence in order to assess the simulator validity. Our main hypotheses were:

-H1: The vehicle traffic in the virtual world is a positive predictor of the different sections of the MEC Spatial Presence Questionnaire.

-H2: A dual task performed during the driving is a negative predictor of the different sections of the MEC Spatial Presence Questionnaire and of the driving performance.

IV. METHOD

A. Participants

Twenty experienced car drivers, with at least five years of experience (14 men and 6 women), were divided into four groups, by crossing two independent variables, i.e., a dual task to be performed or not during the experiment and the presence or not of other vehicles' traffic on the road. The age of the participants ranged from 22 to 45 (M=32.8 years, SD=6.45). All were tested on a voluntary basis, having signed an informed consent form.

B. Apparatus

The experiment was carried out using the SIM^2-IFSTTAR (French Institute of Science and Technology for Transport) fixed-base driving simulator equipped with a multi-actor parallel architecture for traffic simulation (ArchiSim) and an object database SIM^2-IFSTTAR (simulation software). The "ArchiSim" architecture was built on the DR2 traffic simulation model (management of "autonomous" and "enslaved" vehicle with a behavior defined by the scripts for each scenario, simulation generated by captors of punctual and space traffic) and on the 3D SIM2 loop of visualization [7]. The interactive driving station comprised one quarter of a vehicle including a seat, a dashboard, and controls equipped with captors, i.e. pedals and steering wheel. The projected display (at 30 Hz) presented a field of view of 150° horizontally and 40° vertically.

C. Procedure

In our experiment, we used a digital model of the Versailles Satory runway, which is a closed loop of 3.7km, with long straights and corners with different radii of curvature. The first factor was the level of attention induced by the virtual environment with two levels: automated bidirectional traffic or not. The second factor was the level of cognitive involvement induced by the real world with two levels: presence of a dual task or not. The secondary task consisted in launching every minute a digital hourglass by double clicking the mouse of a laptop positioned so that the person had to deport his gaze from the main visual scene. Half of the subjects had to perform the dual task, either in condition "traffic", or in condition "no traffic" on the Satory circuit. The dual task might be considered as a derivative of the time-production task [33]. It is important to note that this dual task was used as a manipulation device and not as a performance measurement. That is the reason why reaction times were not presented in this paper. Whatever the experimental conditions, each participant had to perform 10 laps with a maximum speed of 110 km/h by respecting the Highway Code. We applied a 2 X 2 factorial design. Four experimental groups of five subjects were thus created (see Table I).

TABLE I. EXPERIMENTAL DESIGN

	Dual task	**Traffic**
Group1	No	Yes
Group2	Yes	Yes
Group3	No	No
Group4	Yes	No

D. MEC Spatial Presence Questionnaire

After each session, an adapted version of the MEC Spatial Presence Questionnaire (MEC-SPQ) [11][29] was used. As suggested for each scale, we used a 5-point Likert scale ranging from 1 ('I do not agree at all') to 5 ('I fully agree'). We used 4-item scales for the tested dimensions. On the first level, Visual Spatial Imagery (VSI) was assessed with items such as " When someone describes a space to me, it's usually very easy for me to imagine it clearly"; for allocation of attention "I dedicated myself completely to the medium"; for the Spatial Situational Model (SSM) " I was able to imagine the arrangement of the spaces presented in the medium very well". On the second level, higher Cognitive Involvement (CogInv) was assessed with items such as " I thought most about things having to do with the medium"; for Suspension of Disbelief (Sod)" I didn't really pay attention to the existence of errors or inconsistencies in the medium". Finally, spatial presence was measured and analysed by the self-location dimension (e.g., «I felt as though I was physically present in the environment of the presentation").

E. Driving performance

We analyzed two behavioral variables reflecting the driving performance, i.e. means and Standard Deviations of speed and of Lateral Position (SDLP).

V. RESULTS

A. MEC Spatial Presence Questionnaire

A mean score was computed for each group for the various dimensions of the MEC SPQ (see Table II). Overall, Whatever the section of the questionnaire, participants reported rather high scores. Specifically, "attention" scale (M= 4.2; SD=0.70) had the highest score while "cognitive involvement" scale (M=3.33; SD=0.95) had the lowest.

The generalized linear model was used in order to test our hypotheses, with a 2x2 factorial design and independent groups. Then multivariate analysis of variance (MANOVA) was considered for assessing interaction effects between independent variables. From the MANOVA analysis, no interaction effect was observed between the traffic condition and the dual task condition.

TABLE II. MEC SPQ RESULTS

	Group 1	Group 2	Group 3	Group 4

Attention	M=4.07
	SD=0.72
		M=4.27
		SD=0.28
			M=4.20
			SD=0.93
				M=4.33
				SD=0.91

SSM	M=3.6
	SD= 0.28
		M=3.53
		SD=0.73
			M=4.27
			SD=0.64
				M=3.93
				SD=0.60

CogInv	M=3.07
	SD=1.19
		M=3.80
		SD=0.80
			M=3.33
			SD=1.10
				M=3.13
				SD=0.80

Sod	M=3.80
	SD=0.65
		M=3.67
		SD=0.75
			M=3.80
			SD=0.90
				M=3.93
				SD=0.55

VSI	M=3.05
	SD=0.74
		M=3.85
		SD=1.29
			M=3.55
			SD=0.51
				M=4.30
				SD=0.55

Self location	M=3.50
	SD=0.71
		M=4
		SD=0.18
			M=3.30
			SD=1.31
				M=3.7
				SD=0.69

Whatever the experimental condition no significant effect was observed (p<0.05) on the various sections of the MEC Spatial Presence Questionnaire. Contrary to what had been hoped the dual task and the traffic had no impact on the questionnaire results.

B. *Driving performance*

1) Lateral Position: An interaction effect was first observed between dual task and traffic (F(1, 360)=28.827;p<0.01). Lateral positions of Groups 1 and 2 submitted to traffic were higher than those of groups 3 and 4 not submitted to traffic (see Table III). A dual task effect is also observed in the absence of traffic, (F(1,360)=35.27, p<0.01). Indeed, subjects not submitted to the dual task (Group3) had a higher mean lateral position compared to subjects submitted to the dual task (Group4).

TABLE III. LATERAL POSITION (M)

Groups	Means	SD
G1	1.42	0.80
G2	1.39	0.72
G3	0.98	0.99
G4	0.33	1.11

2) Speed: A dual task effect was observed (F(9,360)=2.33;p=0.012) on 10 laps performed between groups not submitted to the dual task (group 1 and group 3) and groups submitted to the dual task (group 2 and group 4).

TABLE IV. SPEED (km/h)

Groups	Means	SD
G1 & G3	112.14	21.25
G2 & G4	103.02	23.99

Subjects not performing the dual task drove faster than others (see Table IV).

3) SD of lateral position (SDLP): There was no interaction effect between group and lap variables (F(24,1754)=0.60, p=0.94). However, the group significantly influenced the SDLP (F(3, 1754)=156,39, p<0.01). Over the 10 laps, the group 4 submitted to the dual task without traffic had the higher mean contrary to groups 1, 2, 3.

4) SD of speed: There was no interaction effect between group and lap variables (F(24,1754)=0.60, p=0.93). However, the group significantly influenced the SD of speed (F(3, 1754)=41,018, p<0.01). Over the 10 laps, the group 2 submitted to the dual task with traffic had the higher mean contrary to groups 1, 3, 4.

V. DISCUSSION

Results globally showed that whatever the experimental condition, no significant difference was clearly observed for the different sections of our presence questionnaire. Indeed, the vehicle traffic in the virtual world was not a positive predictor of the different sections of the MEC Spatial Presence Questionnaire. The dual task was not a negative predictor of these different sections either. Although the dual task did not have a strong effect on the subjective measures of presence, it affected the behavioral measures. As described in the literature [13][32], low values in driving performance (SD of speed and SDLP) indicated a good steering control and a stable and consistent driving. Indeed, the SD of speed in traffic condition and the SDLP in no traffic condition were higher in dual task than in single task. Drivers submitted to the dual task without traffic drove in the middle of the road (the smallest mean lateral position), which could be interpreted as an efficient strategy but pretty inconsistent with the Highway Code. Similarly, participants submitted to the dual task in traffic condition have tried to reduce their speed as a compensatory strategy to deal with the dual task. Unfortunately, it appeared that they also failed to maintain a stable driving with a lack of speed control (the highest SD of speed). Thus, driving performance seemed to be globally impaired by the dual task.

In conclusion, the main outcome was that behavioral measures revealed significant effects of the manipulated variables (traffic and dual task) on driving performance. These effects showed that participants took into account these variables. For instance, they indicate that traffic clearly influenced the lateral position in the lane, which can be considered as a positive (although rough) indicator of presence and reactivity to the simulator's scenario. However, these behavioral effects were not confirmed by subjective reports. One explanation could be that, whatever the current experimental conditions, driving activity did not involve high-level, conscious, cognitive processes. Despite high scores reported through the MEC Spatial Presence Questionnaire, driving was probably more based on a set of procedures or routines. Our experimental conditions might have been insufficient to induce several distinct levels of attention, involvement and suspension of disbelief, leading to several distinct levels of presence. Thus, high levels of self-reported presence (positively correlated with behavioral measures) might require to develop more challenging scenarios, in terms of controlled attention, cognitive involvement and more specifically, in terms of emotions induced by the media.

ACKNOWLEDGMENT

The authors would like to thank Thelma Coyle and Vincent Perrot for assistance in data entry and analysis.

REFERENCES

[1] R. W. Allen and D. T. McRuer, The effect of adverse visibility on driver steering performance in an automobile simulator, SAE Transactions, Vol. 86, Feb. 1977, pp. 1081-1092.

[2] R. W. Allen, A. C. Stein, B. L. Aponso, T. J. Rosenthal, and J. R. Hogu, A low cost, part task driving simulator based on microcomputer technology. Transportation Research Record, 1270, Sep. 1990, pp. 107-113.

[3] R. W. Allen, and H. R. Jex, Driving simulation – Requirements, mechanization and application, SAE Technical Paper Series Paper number 800448, Feb. 1980, pp. 1-12.

[4] R. W. Allen, D. G. Mitchell, A. C. Stein, and J. R. Hogue, Validation of real-time man-in-the-loop simulation. VTI Rapport, 372A, Part 4, Sep. 1991, pp. 18-31.

[5] G. J. Blaauw, Driving experiments and task demands in simulator and instrumented car. Human Factors, Aug. 1982, pp. 473-486.

[6] J. Bürki-Cohen and T. H. Go, The effect of simulator motion cues on initial training of airlines pilots. AIAA-2005-6109. Proceedings of the American Institute of Aeronautics and Astronautics Modeling and Simulation Technologies Conference. San Francisco, CA, Aug. 2005.

[7] S. Espié, P. Gauriat, P, and M. Duraz, Driving simulators validation: The issue of transferability of results acquired on simulator. In National Advanced Driving Simulator, University of Iowa (Eds) Proc. Driving Simulation Conference DSC North-America'2005, november 30th - october 2nd, Orlando, FL., Center for Advanced Transportation Systems Simulation (2005), pp. 149-156.

[8] S. T. Godley, T. J. Triggs, and B. N. Fildes, Driving simulator validation for speed research.Accident Analysis and Prevention, vol. 34, May. 2002, pp. 589–600.

[9] M. Goodman, F. D. Bents, L. Tijerina, W. W. Wierwille, N. Lerner, and D. Benel, An investigation of the safey implications of wireless communications in vehicles (DOT HS 808 635). Washington, DC: National highway Traffic Safety Administration, US Department of Transportation, Jun 1997.

[10] P. A. Hancock, The future of simulation. In D. Vicenzi, J. Wise, M. Mouloua, & P. A. Hancock (Eds.), Human factors in simulation and training Boca Raton, FL: CRC Press, Dec. 2008, pp. 169-188.

[11] W. A. Ijsselsteijn, H. de Ridder, J. Freeman, and S. Avons, S. E. Presence: Concept, determinants and measurement. Paper presented at Photonics West-Human Vision and Electronic Imaging, San Jose, CA, Jan. 2000.

[12] A. H. Jamson, Curve negociation in the Leeds Driving Simulator: The role of driver experience. In D. Harris (Ed.), Engineering psychology and cognitive ergonomics. Ashgate, vol. 3, Dec. 1999, pp. 351-358.

[13] S. L. Jamson and A. H. Jamson, The validity of a low-cost simulator for the assessment of the effects of in-vehicle information systems, SAFETY SCI, vol. 48, Aug. 2010, pp. 1477-1483.

[14] M. J. Johnson, T. Chahal, A. Stinchcombe, N. Mullen, B. Weaver, and M. Bedard, Physiological responses to simulated and on-road driving. International Journal of Psychophysiology, vol. 81, Mar. 2011, pp. 203–208.

[15] B. H. Kantowitz, Using microworlds to design intelligent interfaces that minimize driver distraction. Proceedings of the first international driving symposium on human factors in driver assessment, training and vehicle design, Aspen, CO, Aug. 2001, pp. 42-57.

[16] N. A Kaptein, J. Theeuwes, and R. Van der Horst, R, Driving simulator Validity: Some considerations. Transportation Research Record, 1550, Jan. 1996, pp. 30-36.

[17] A. Kemeny and F. Panerai, Evaluating perception in driving simulation experiments, Trends in Cognitive Sciences, vol. 7, Jan. 2003, pp. 31–37.

[18] K. M. Lee, Presence, explicated. Communication Theory, vol. 14, Jan. 2004, pp. 27–50.

[19] A. T. Lee, Flight simulation. Aldershot, United Kingdom: Ashgate, 2005.

[20] L. Nilsson, Behavioural research in an advanced driving simulator-experiences of the VTI system, Proceedings of the Human Factors and Ergonomics Society 37th Annual, Oct. 1993.

[21] T. A. Ranney, L. A. Simmons, Z. Boulos, and M. M. Macchi, Effect of an afternoon nap on nighttime performance in a driving simulator, Transportation Research Record, 1686, Nov. 2000, pp. 49-56.

[22] H. Regenbrecht and T. Schubert, Real and illusory interaction enhance presence in virtual environments.

Presence: Teleoperators and Virtual Environments, 11, Aug. 2002, pp. 425-434.

[23] B. Reimer, L. A D'Ambrosio, J. E. Coughlin, M. E. Kafrissen, and J. Biederman, Using selfreported data to assess the validity of driving simulation data, Behavior Research Methods, vol. 38, Feb. 2006, pp. 314–324.

[24] T. W. Schubert, A new conceptualization of spatial presence: Once again, with feeling. Communication Theory, vol. 19, May. 2009, pp. 161-187

[25] T. W. Schubert, The sense of presence in virtual environments: A three-component scale measuring spatial presence, involvement, and realness. Zeitschrift für Medienpsychologie, vol. 15, Feb. 2003, pp. 69-71

[26] M. Slater, B. Lotto, M. M. Arnold, and M. V. Sanchez-Vives, How we experience immersive virtual environments : the concept of presence and its measurement. Anuario de Psicologia, vol. 40, Feb. 2009, pp. 193–210.

[27] M. Slater, Measuring presence: A response to the Witmer and Singer presence questionnaire. Presence: Teleoperators and Virtual Environments, Vol. 8, Mar. 1999, pp. 560-565.

[28] J. G. Tichon, Using presence to improve a virtual training environment. CyberPsychology &Behavior, vol. 10, Jun. 2007, pp. 781–787.

[29] P. Vorderer, W. Wirth, F. R. F. Gouveia, T. Saari, F. Jäncke, S. Böcking, H. Schramm, A. Gysbers, T. Hartmann, C. Klimmt, J. Laarni, N. Ravaja, A. Sacau, T. Baumgartner, and P. Jäncke, MEC Spatial Presence Questionnaire (MEC-SPQ):Short Documentation and Instructions for Application. Report to the European Community, Project Presence : MEC (IST-2001-37661), Jun. 2004.

[30] M. G. Wade and C. Hammond, Simulation validation: Evaluating driver performance in simulation and the real world. Final Report. (MN/RC-1998-28). Minneapolis, MN: University of Minnesota, Jul. 1998.

[31] W. Wirth, T. Hartmann, S. Bocking, P. Vorderer, C. Klimmt, H. Schramm, T. Saari, J. Laarni, N. Ravaja, and F. R. Gouveia, A process model of the formation of spatial presence experiences. Media Psychology, vol. 9, Mar. 2007, pp. 493–525.

[32] M. Wittmann, M. Kiss, P. Gugg, A. Steffen, M. Fink, E. Poppel, and H. Kamiya, Effects of display location of a visual in-vehicle task on simulated driving. Appl.Ergon. vol. 37, Nov. 2006, pp. 187–199.

[33] D. Zakay, J. Shub, Concurrent duration production as a workload measure. Ergonomics, vol. 41, Aug. 1998, pp. 1115–1128.

Consistency of the Stochastic Mesh Method

Yuri Kashtanov

Faculty of Mathematics and Mechanics
Saint Petersburg State University
Russia
Email: y.kashtanov@spbu.ru

***Abstract*—A Monte Carlo method for pricing high-dimensional American options is considered. The consistency of the stochastic mesh method is studied. Some ”natural” estimators of this method have infinite variance. A modification which gives consistent estimators for a diffusion model is proposed. It is shown that the variance of estimators is inverse proportional to the number of points in each layer of the mesh.**

Keywords–Optimal stopping; American option; Stochastic mesh.

I. Introduction

Pricing of American options may be formulated as a problem of the theory of optimal stopping: if S_n is the price process of the underlying asset and f_n are the option payoffs then the option price C has the expession [1] $\mathcal{C} = \sup_{\tau<T} \mathbf{E} f_\tau(S_\tau)$, T being the expiry date, τ - an arbitrary Markov moment. In the case when S_n is a high-dimensional process, numerical methods with regular mesh are very time-consuming. Mark Broadie and Paul Glasserman [2] suggested a stochastic mesh method which does not depend on the dimension. In parallel, Athanassios N. Avramidis and Heinrich Matzinger [3] formulated general conditions under which estimators of the method are statistically consistent. Using Malliavin calculus approach, Vlad Bally and others [4] suggested analogous estimators for a generalized Black-Scholes model; these estimators may be transformed [5] to the formula (20) of the present paper. This slight modification of formula (5) allows to convert the estimator with infinite variance to the one with a finite variance. The issue of the method consistency applying to the general diffusion model was not investigated till now and is the goal of the paper. Section 2 describes the stochastic mesh method; in Section 3, theoretical assertions, concerning consistency, are formulated; in Section 4, numerical examples, which illustrate these results, are given; proofs are gathered in the Appendix.

II. The Stochastic Mesh Method

Let prices of d securities $S_n = (S_{n1}, \dots, S_{nd})$, $S_n \in R^d = X$, be given at discrete moments $t_n = n\Delta t$, where $\Delta t = T/N$. Assume also, that discounted prices are martingales that means the following equalities are fulfilled

$$\mathbf{E}(S_{n+1,i} \mid S_1, \dots, S_n) = e^{r\Delta t} S_{ni}, \tag{1}$$

r being an interest rate, which is assumed to be constant. Onwards, it will be more convenient to deal with sequences $\xi_{ni} = \ln S_{ni} - rt_n$. Assume that $\xi_n = (\xi_{n1}, \dots, \xi_{nd})$ is a Markov chain with values in X and $p_n(x, dy)$ are transitional probabilities.

Let an American option be given with an expiration T and discounted payoffs of the form $f_n = f_n(\xi_n)$. Define successively functions $Y_n(x)$:

$$Y_N(x) = f_N(x), \quad Y_n(x) = \max\left(f_n(x), \mathbf{E}_{n,x} Y_{n+1}(\xi_{n+1})\right). \tag{2}$$

It is known [1] that the option price $\mathcal{C}$ satisfies $\mathcal{C} = Y_0(\xi_0)$.

For every time step n, a set of random points $\bar{x}_n = \left\{x_n^i\right\}_{i=1}^M$ (”mesh”) is constructed as a Markov chain with transitional probabilities

$$\bar{q}_n(\bar{x}, d\bar{y}) = q_{n,1}(\bar{x}, dy_1)...q_{n,M}(\bar{x}, dy_M). \tag{3}$$

Make an assumption on probabilities $q_{n,j}(\bar{x}, dy)$ that densities

$$\rho_{n,j}(\bar{x}, x, y) = p_n(x, dy)/q_{n,j}(\bar{x}, dy) \tag{4}$$

exist. For a shortage of further notations, introduce random variables defined on the mesh: $Y_n(j) = Y_n(x_n^j)$, $\rho_n(x, j) = \rho_{n,j}(\bar{x}_{n-1}, x, x_n^j)$, $\rho_n(i, j) = \rho_n(x_{n-1}^i, j)$.

Recursively construct random sequence $\check{Y}_n(x)$: first, set $\check{Y}_N(x) = f_N(x)$, then define

$$\check{Y}_n(x) = \max\left(f_n(x), \frac{1}{M}\sum_{j=1}^{M} \rho_{n+1}(x, j)\check{Y}_{n+1}(j)\right), \tag{5}$$

where using analogous notations $\check{Y}_{n+1}(j) = \check{Y}_{n+1}(x_{n+1}^j)$.

Introduce $\mathcal{F}_n$ – a σ-algebra generated by values $\bar{x}_1, \dots, \bar{x}_n$. Denote a conditional expectation with respect to $\mathcal{F}_n$ as $\mathbf{E}_{\mathcal{F}_n}$. Assume that for every n, random variables j_n taking values $1, \dots, M$ with equal probabilities, are defined and which are independent in total and with respect to $\mathcal{F}_N$, then the last equality can be rewritten in the form

$$\check{Y}_n(x) = \max\left(f_n(x), \mathbf{E}_{\mathcal{F}_N} \rho_{n+1}(x, j_{n+1})\check{Y}_{n+1}(j_{n+1})\right). \tag{6}$$

It follows from (2) and (6) that

$$\left|\check{Y}_0(x_0) - Y_0(x_0)\right| \leqslant \sum_{n=0}^{N-1} \mathbf{E}_{\mathcal{F}_N} \prod_{k=1}^{n} \rho_k(j_{k-1}, j_k)|\Delta_n(j_n)|, \tag{7}$$

where

$$\Delta_n(x) = \mathbf{E}_{\mathcal{F}_N} \rho_{n+1}(x, j_{n+1}) Y_{n+1}(j_{n+1}) - \mathbf{E}_{n,x} Y_{n+1}(\xi_{n+1}).$$

In virtue of (3), values $x_{n+1}^1, \dots, x_{n+1}^M$ are independent with regard to $\mathcal{F}_n$, therefore

$$\mathbf{E}_{\mathcal{F}_n} \Delta_n^2(x) \leqslant \frac{1}{M} \mathbf{E} \rho_{n+1}^2(x, j_{n+1}) Y^2(j_{n+1}). \tag{8}$$

It follows in turn from (7) and (8) that under the condition

$$\mathbf{E}[\rho_1(x_0, j_)...\rho_n(j_{n-1}, j_n)Y_n(j_n)]^2 < \infty, \quad n = 1, \dots, N; \tag{9}$$

the inequality

$$\mathbf{E}(\check{Y}_0 - \mathcal{C})^2 \leqslant C/M \tag{10}$$

is fulfilled ant thereby $\check{Y}_0$ is a consistent estimator for $\mathcal{C}$.

In [2] some substantiation is given for using the average distribution, which is defined by the expression

$$q_{n,j}(\bar{x}, dy) = M^{-1} \sum_i p_n(x^i, dy). \tag{11}$$

In some models (e.g. Black-Scholes), transitional probabilities for several steps $p_{k,n}(x, dy)$ are known, in this case one can use [2]

$$q_{n,j}(\bar{x}, dy) = p_{0,n}(x_0, dy). \tag{12}$$

The objective of the paper is to study the statistical consistency of these and other estimators for the general diffusion model.

III. Consistent estimators

First, consider a case when x_n are normal vectors in R^d defined by the sequence

$$x_{n+1} = x_n + a_n \varepsilon_{n+1} + b_n, \tag{13}$$

a_n being $d \times m$ - matrices, ε_n – independent standard normal vectors in R^m, $A_n = a_n a_n^*$, $b_n(i) = -0.5 A_n(i,i)$. Denote $\psi(A_n, z) = (A_n z, z)$ and suppose that

$$\sup_{||z||=1} \psi(A_n, z) \geqslant \bar{A}_0 > 0, \tag{14}$$

then densities $p_{k,n+1}(x, y)$ exist for transition from point x at the moment k to point y at the moment $n+1$:

$$p_{k,n+1}(x, y) = c_{kn} \exp(-0.5\psi(\Sigma_{k,n}^{-1},\ x + B_{k,n} - y)),$$

where $c_{kn} = [(2\pi)^d \det \Sigma_{kn}]^{-1/2}$, $\Sigma_{k,n} = \sum_{i=k}^n A_i$ and $B_{k,n} = \sum_{i=k}^n b_i$. Suppose payoffs imply that

$$Y_n(x) \leqslant \sum_i c_i (e^{k_i x_i} + 1). \tag{15}$$

Theorem 1. *The inequality (10) for the estimator with average densities (11) is valid under conditions (14), (15).*

Proofs are given in the Appendix.

Now, consider the mesh generated according to the formula (12). Note that it is approximately 2 times less time-consuming for calculation than (11) but the variance of $\check{Y}_0$ may be infinite if N is big enough. Really, let $N \geqslant 3$ then

$$\mathbf{E}\check{Y}_0^2 \geqslant \mathbf{E}\left[\rho_2(j_1, j_2)\rho_3(j_2, j_3)f_3(j_3)\right]^2$$
$$= \frac{1}{M^3} \int\limits_{X^3} p(x_0, y_1) \frac{p^2(y_1, y_2)}{p_{0,2}(y_0, y_2)} \frac{p^2(y_2, z_3)}{p_{0,3}(y_0, y_3)} f_3^2(y_3) dy_1 dy_2 dy_3.$$

Consider the case when $d = 1$ and a_n are constant, then after integrating by y_1, y_2 receive

$$\mathbf{E}\check{Y}_0^2 \geqslant c \int_R \exp\left(\frac{7y^2}{78a^2}\right) f_3^2(ya\sqrt{\Delta t} + 3b\Delta t) dy. \tag{16}$$

If $f_3(x) > \varepsilon$ for $x > K$ or $x < -K$ with some $\varepsilon, K > 0$ then the integral does not converge and thus the variance is infinite. Below, a modification of this estimator with finite variance will be constructed for a more general diffusion model.

Now, consider a discretization of a diffusion process according to the Euler scheme:

$$x_{n+1} = x_n + a_n(x_n)\varepsilon_{n+1} + b_n(x_n). \tag{17}$$

Suppose that for some $\bar{A}_0$, $\bar{A}_1$ the inequalities are fulfilled

$$\bar{A}_0 |z|^2 \leqslant (A_n(x)z, z) \leqslant \bar{A}_1 |z|^2; \tag{18}$$

then $|b_n(x)| \leqslant \bar{b}$ for some $\bar{b}$.

Let the mesh be produced by transition probabilities

$$q_{n,j}(\bar{x}, dy) = q_n(y)dy = c_n \exp(-|y - x_0|^2/(2s^2 n))dy. \tag{19}$$

Consider a following modification of the scheme (5) :

$$\check{Y}_n(i) = \max\left(f_n(i), \frac{\sum_j \rho_{n+1}(i,j)\ \check{Y}_{n+1}(j)}{\sum_j \rho_{n+1}(i,j)}\right). \tag{20}$$

Note that unlike the scheme (5), the estimator of the expectation (6) is biased here.

Theorem 2. *Suppose that $f_n(x) \leqslant F$ and $s^2 > 0.5\bar{A}_1$ then the inequality (10) is valid for the scheme (20).*

Note. Consider a more general model for which transitional densities may be estimated by a mixture

$$p_n(x, y) \leqslant C \sum_{k=1}^K p_n^{(k)}(x, y), \tag{21}$$

$p_n^{(k)}(x, y)$ being normal densities with correlation matrices $A_n^{(k)}(x)$. Suppose these matrices satisfy condition (18) with common constants A_0, A_1, then, as follows from the proof, the assertion of the theorem still holds true.

Now, suppose that instead of the condition $f_n(x) \leqslant F$ the the following condition is fulfilled: for some $g_n(x)$, G and F

$$f_n(x) \leqslant F g_n(x), \qquad \int_X dy\, p_n(x, y) g_n(y) \leqslant G g_{n-1}(x). \tag{22}$$

Denote $\widetilde{g}_n(x) = \int_X dy\, p_n(x, y) g_n(y)$ and consider the scheme

$$\check{Y}_n(i) = \max\left(f_n(i), \widetilde{g}_n(i) \frac{\sum_j \rho_{n+1}(i,j)\ \check{Y}_{n+1}(j)}{\sum_j \rho_{n+1}(i,j) g_{n+1}(j)}\right). \tag{23}$$

Theorem 3. *Suppose conditions (22) are fulfilled, densities $\widetilde{p}_n(x, y) = p_n(x, y) g_n(y)/\widetilde{g}_n(x)$ satisfy (21) and $s^2 > 0.5A_1$ then inequality (10) is valid for the scheme (23).*

Note 1. Though the variance decreases when M tends to infinity, it may be still quite big for a chosen M. To reduce it one can use a method of control variates [2], which implies the substitution of the term $\check{Y}_{n+1}(j)$ in formulas (5), (23) by the term $\check{Y}_{n+1}(j) + \nu_{n+1}(j)$ with appropriate functions $\nu_{n+1}(x)$, which satisfy the condition $\mathbf{E}_{n,x} \nu_{n+1}(\xi_{n+1}) = 0$.

Note 2. Give an example when conditions of the theorem are fulfilled. For many multi-dimensional options (basket, geometrical average, maximum [2]), functions $g_n(x)$ may be chosen in the form $g_n(x) = \sum_{i=1}^d \exp(x_i)$. Since $\exp(\xi_{ni})$

are martingales then $\widetilde{g}_n(x) = g_n(x)$. It is easy to verify that the representation (21) takes place under $K = d$, correlation matrices $A_n^{(k)}(x) = A_n(x)$ and $b_n^{(k)}(x) = b_n(x) + A_n(x)e_k$.

IV. Numerical results

Consider a geometrical average option with payoff functions $f_n = ((S_{n1}..S_{nd})^{1/d} - K)^+$. In the case of Black-Scholes model, the calculation may be reduced to a one-dimensional American call with dividends and the solution may be obtained with high accuracy either by a regular mesh or by the analytical approximation [6]. Under parameters $d = 5$, $a = 0.3$, $r = 0.02$, $S = 100$, $K = 100$, $T = 1$, the price of American option in the continuous model is $\mathcal{C}_a = 4.63$, which is strictly greater than the price of the correspondent European option $\mathcal{C}_e = 4.46$. In discrete models the price of American option increases from $\mathcal{C}_e$ to $\mathcal{C}_a$ with increasing N. As a control variate, take the corresponding European call option.

TABLE I. Comparison of estimators

Estimator, density			(5), (11)		(23), (19)	
N	True	M	Av	Err	Av	Err
6	4.59	300	4.61	0.008	4.59	0.006
		1200	4.60	0.004	4.59	0.0026
12	4.61	300	4.7	0.017	4.61	0.0055
		1200	4.66	0.006	4.61	0.0023
24	4.62	300	4.86	0.04	4.63	0.005
		1200	4.81	0.015	4.62	0.0025

The table includes the true price, estimates and statistical errors, corresponding to 99% confidence level by 50 realizations. It illustrates the assertion that the variance is inverse proportional to M. It may be observed from the table (and is known from the theory) that the estimator (5), (11) is biased high and this bias increases with the number of time steps; to compensate it one should increase the mesh size. Note also that the first estimator is approximately 2 times more time-consuming than the second one.

Consider the same option in the following diffusion model $\Delta x_{ni} = \sqrt{\Delta t}\sigma(x_{n-1,i})[\alpha\varepsilon_{ni} + \rho\varepsilon_{n,d+1}] - 0.5\sigma^2(x_{n-1,i})\Delta t$, where $\alpha^2 + \rho^2 = 1$, $\rho = 0.5$, $\sigma(x)$ being a decreasing function, which varies from 0.6 to 0.1, $\sigma(x_0) = 0.3$.

TABLE II. Case of the general diffusion model

s	0.2		0.43		0.5	
M	Av	Err	Av	Err	Av	Err
300	7.13	0.012	7.19	0.009	7.21	0.012
600	7.13	0.008	7.18	0.007	7.19	0.009
1200	7.14	0.007	7.16	0.005	7.17	0.007

Rough 1-dimensional approximations give prices from 7.1 to 7.2. The simulation results for the estimator (19), (23) with $N = 12$ and different values of the parameter s are presented in the Table II. The value 0.43 is slightly greater than $A_1/\sqrt{2}$, which provides the variance finiteness.

V. Appendix

Proof of theorem 1. First, note that $\rho_1(x_0, j) = 1$ for every j; further, $\mathbf{E}_{\mathcal{F}_N}\rho_2(j_1, j) = 1$ also for every j and by induction we receive that the right part in (7) is equal to $\sum_{n=0}^{N-1} |\Delta_n(j_n)|$. Therefore, to prove consistency it is sufficient to show that for every $n < N$ the following integrals are finite:

$$\mathbf{E}\rho_{n+1}^2(j_n, 1)Y_{n+1}^2(1) = \int_X dy\; Y_{n+1}^2(y)\mathbf{E}\frac{\sum_j p_{n+1}^2(x_n^j, y)}{\sum_j p_{n+1}(x_n^j, y)}. \tag{24}$$

To estimate the expectation under the sign of the integral, prove the following lemma:

Lemma. *Let η_i, $i = 1, \dots, M$, be positive, independent, similar distributed random variables, $\zeta_i = c\eta_i^{1+\varepsilon}$ $\varepsilon > 0$; denote $\bar\eta = M^{-1}\sum_j \eta_j$, $\bar\zeta = M^{-1}\sum_j \zeta_j$, then*

$$\mathbf{E}\bar\zeta/\bar\eta \leqslant 3\mathbf{E}\zeta/\mathbf{E}\eta.$$

Proof. From the equality

$$\frac{\bar\zeta}{\bar\eta} = \frac{\bar\zeta}{\mathbf{E}\eta} - \frac{1}{\mathbf{E}\eta}\frac{\bar\zeta}{\bar\eta}(\bar\eta - \mathbf{E}\eta) \tag{25}$$

the estimate follows:

$$\mathbf{E}\frac{\bar\zeta}{\bar\eta} \leqslant \frac{\mathbf{E}\zeta}{\mathbf{E}\eta} + \frac{1}{\mathbf{E}\eta}\mathbf{E}\frac{\bar\zeta}{\bar\eta}\left|\bar\eta - \mathbf{E}\eta\right|. \tag{26}$$

Further, from inequality $\sum_j \eta_j^{1+\varepsilon} \leqslant (\sum_j \eta_j)^{1+\varepsilon}$ it follows that

$$\bar\zeta/\bar\eta \leqslant \Big(\sum_j \eta_j^{1+\varepsilon}\Big)^{\frac{\varepsilon}{1+\varepsilon}}. \tag{27}$$

Using the last estimate and then Hölder's inequality receive

$$\begin{aligned} \mathbf{E}\frac{\bar\zeta}{\bar\eta}\left|\bar\eta - \mathbf{E}\eta\right| &\leqslant \mathbf{E}\Big(\sum_j \eta_j^{1+\varepsilon}\Big)^{\frac{\varepsilon}{1+\varepsilon}}\left|\bar\eta - \mathbf{E}\eta\right| \qquad (28)\\ &\leqslant \mathbf{E}^{\frac{\varepsilon}{1+\varepsilon}}\Big(\sum_j \eta_j^{1+\varepsilon}\Big)\,\mathbf{E}^{\frac{1}{1+\varepsilon}}\left|\bar\eta - \mathbf{E}\eta\right|^{1+\varepsilon}. \end{aligned}$$

Now use the following theorem ([7], c.79):

Let $X_1, \dots, X_n$ be independent random variables with zero mean and finite absolute moments of order p ($1 \leqslant p \leqslant 2$); then

$$\mathbf{E}\left|\sum_{k=1}^n X_k\right|^p \leqslant \Big(2 - \frac{1}{n}\Big)\sum_{k=1}^n \mathbf{E}|X_k|^p.$$

Using this theorem one can estimate (28) by the value

$$M^{\frac{\varepsilon}{1+\varepsilon}}\mathbf{E}^{\frac{\varepsilon}{1+\varepsilon}}\eta^{1+\varepsilon}\;\frac{2}{M^{\frac{\varepsilon}{1+\varepsilon}}}\mathbf{E}^{\frac{1}{1+\varepsilon}}\eta^{1+\varepsilon} = 2\mathbf{E}\zeta.$$

Substituting the obtained estimates in (26), one receive the assertion of the lemma.

Now denote $\eta_{k,n+1}(j) = p_{k,n+1}(x_{k,j}, y)$; then applying lemma (with $\varepsilon = 1$) to expectation in (24), receive

$$\mathbf{E}\frac{\sum_j \eta_{n,n+1}^2(j)}{\sum_j \eta_{n,n+1}(j)} \leqslant 3\mathbf{E}\frac{\mathbf{E}_{\mathcal{F}_{n-1}}\eta_{n,n+1}^2(1)}{\mathbf{E}_{\mathcal{F}_{n-1}}\eta_{n,n+1}(1)}. \tag{29}$$

According to Chapman-Kolmogorov equations, the denominator is equal to $\sum_j \eta_{n-1,n+1}(j)$. Calculate $\mathbf{E}_{\mathcal{F}_{k-1}}\eta_{k,n+1}^m(1)$ with $m > 1$:

$$\mathbf{E}_{\mathcal{F}_{k-1}}\eta_{k,n+1}^m(1) = \widetilde{c}_{kn}\sum_j e^{-\psi(\widetilde\Sigma_{k-1,n}^{-1},\, x_{k-1}^j + B_{k-1,n} - y)/2},$$

where

$$\widetilde{\Sigma}_{k-1,n} = A_{k-1} + \frac{1}{m}\Sigma_{k,n} = \Sigma_{k-1,n} - \frac{m-1}{m}\,\Sigma_{k,n}. \quad (30)$$

It follows from (30) that $\widetilde{\Sigma}^{-1}_{k-1,n} = \Sigma^{-1}_{k-1,n}(I - D_{k,n})^{-1}$, where $D_{k,n} = (m-1)/m\ \Sigma_{k,n}\Sigma^{-1}_{k-1,n}$, I being a unit $d \times d$ matrix. Since $||D_{k,n}|| < 1$, one can estimate the inverse matrix $\widetilde{\Sigma}^{-1}_{k-1,n} \geqslant \Sigma^{-1}_{k-1,n}(I + D_{k,n})$. On the other hand, $D_{k,n} \geqslant \varepsilon I$ with some value $\varepsilon = \varepsilon_{k,n,m} > 0$; therefore, an estimation from below is $\widetilde{\Sigma}^{-1}_{k-1,n} \geqslant (1+\varepsilon)\Sigma^{-1}_{k-1,n}$. Particularly, it follows from this estimate that

$$\frac{\mathbf{E}_{\mathcal{F}_{n-1}}\eta^2_{n,n+1}(1)}{\mathbf{E}_{\mathcal{F}_{n-1}}\eta_{n,n+1}(1)} \leqslant C_1\frac{\sum_j \eta^{1+\varepsilon}_{n-1,n+1}(j)}{\sum_j \eta_{n-1,n+1}(j)}$$

with $\varepsilon = \varepsilon_{n-1,n,2}$. The lemma may be applied to the obtained expression as well, and after n iterations receive

$$\mathbf{E}\frac{\sum_j p^2_{n+1}(x^j_n, y)}{\sum_j p_{n+1}(x^j_n, y)} \leqslant C_n p^{\varepsilon}_{0,n+1}(x_0, y).$$

Under condition (15) the integral in (24) is finite. Thus, the proof is complete.

Proof of theorem 2. Introduce an additional notation

$$\bar{\rho}_n(i,j) = \rho_n(i,j)/\sum_k \rho_n(i,k)$$

and in the same way as in (7) obtain

$$\left|\check{Y}_0 - Y_0\right| \leqslant \sum_{n=0}^{N-1}\sum_{i_1,\dots,i_n}\bar{\rho}_1(0,i_1)\dots\bar{\rho}_n(i_{n-1},i_n)|\Delta_n(i_n)|,$$

where $\Delta_n(i) = \sum_j \bar{\rho}_{n+1}(i,j)Y_{n+1}(j) - \mathbf{E}_{n,x_{ni}}Y_{n+1}(\xi_{n+1})$. Squaring both parts and taking into account that $\bar{\rho}_n(i,j)$ is a distribution by j, due to Hölder inequality obtain

$$\mathbf{E}\left(\check{Y}_0 - Y_0\right)^2 \leqslant N\sum_{n=0}^{N-1}\sum_{i_1,\dots,i_n}\mathbf{E}\prod_{k=1}^{n}\bar{\rho}_k(i_{k-1},i_k)\Delta^2_n(i_n). \quad (31)$$

Define values $d_{kn}(i)$:

$$\begin{aligned} d_{nn}(i) &= \mathbf{E}_{\mathcal{F}_n}\Delta^2_n(i), \\ d_{k-1,n}(i) &= \mathbf{E}_{\mathcal{F}_{k-1}}\bar{\rho}_k(i,j_k)d_{k,n}(j_k). \quad k < n, \end{aligned}$$

and show that the following estimate takes place

$$d_{kn}(i) \leqslant \min\left(F^2, \frac{C}{M}\sum_{m=k}^{n}\phi_{m+1,m+1-k}(x_{ki})\right), \quad (32)$$

where $\phi_{rl}(x) = e^{\frac{(|x-x_0|+\beta_l)^2}{s^2r-0.5\bar{A}_1 l}}$, $\beta_l = \bar{b}\sum_{i=1}^{l} d^{(i-1)/2}$.

Fix the index i and represent $\Delta_n(i)$ in a form $\Delta_n(i) = \bar{\zeta}/\bar{\eta}$, where $\bar{\zeta} = \sum_j \zeta_j/M$, $\bar{\eta} = \sum_j \eta_j/M$,

$$\begin{aligned} &\zeta_j = \rho_{n+1}(i,j)\left[Y_{n+1}(j) - \mathbf{E}_{n,x_{ni}}\widetilde{Y}_{n+1}(S_{n+1})\right], \\ &\eta_j = \rho_{n+1}(i,j),\ \mathbf{E}_{\mathcal{F}_n}\zeta_j = 0,\ \mathbf{E}_{\mathcal{F}_n}\eta_j = 1. \end{aligned}$$

It follows from (26) that

$$\mathbf{E}_{\mathcal{F}_n}\left(\frac{\bar{\zeta}}{\bar{\eta}}\right)^2 \leqslant 2\mathbf{E}_{\mathcal{F}_n}\bar{\zeta}^2 + 2\mathbf{E}_{\mathcal{F}_n}\left(\frac{\bar{\zeta}}{\bar{\eta}}\right)^2(\bar{\eta}-1)^2.$$

First, note that $|\zeta_j| \leqslant F|\eta_j|$, and second that with respect to $\mathcal{F}_n$ values $\bar{\eta} - 1$ and $\bar{\zeta}$ are sample averages of independent random variables with zero mean, therefore

$$\mathbf{E}_{\mathcal{F}_n}\Delta^2_n(i) \leqslant \frac{C}{M^2}\sum_j \mathbf{E}_{\mathcal{F}_n}\eta^2_j.$$

Now calculate $\mathbf{E}_{\mathcal{F}_n}\eta^2_j$; denote $z^i_n = x^i_n + b_n(x^i_n) - x_0$ then

$$\begin{aligned} \mathbf{E}_{\mathcal{F}_n}\eta^2_j &= c_{ni}\exp(\psi([(n+1)s^2I - 0.5A_n(x^i_n)]^{-1}, z^i_n)/2) \\ &\leqslant C\phi_{n+1,1}(x^i_n). \end{aligned} \quad (33)$$

Thus, (32) is fulfilled for $k = n$. Now suppose that the inequality is fulfilled for the index $k+1$; estimate $d_{kn}(i)$. Redenote $\zeta_j = \rho_{k+1}(i,j)d_{k+1,n}(j)$, $\eta_j = \rho_{k+1}(i,j)$, then $d_{kn}(i) = \mathbf{E}_{\mathcal{F}_k}\bar{\zeta}/\bar{\eta}$. Split the region of integration and use the induction assumption that $d_{k+1,n}(i) \leqslant F^2$:

$$\begin{aligned} \mathbf{E}_{\mathcal{F}_k}\bar{\zeta}/\bar{\eta} &= \mathbf{E}_{\mathcal{F}_k}\bar{\zeta}/\bar{\eta}\,1_{\{\bar{\eta}\geqslant 1/2\}} + \mathbf{E}_{\mathcal{F}_k}\bar{\zeta}/\bar{\eta}\,1_{\{\bar{\eta}<1/2\}} \\ &\leqslant 2\mathbf{E}_{\mathcal{F}_k}\bar{\zeta} + F^2\mathbf{E}_{\mathcal{F}_k}1_{\{|\bar{\eta}-1|>1/2\}}. \end{aligned} \quad (34)$$

For the first term use the induction assumption

$$\mathbf{E}_{\mathcal{F}_k}\bar{\zeta} \leqslant \frac{C}{M}\sum_{m=k+1}^{n}\int_X dy\ p_{k+1}(x^i_k, y)\phi_{m+1,m-k}(y),$$

and then the estimate $\phi_{r,l}(y) \leqslant \sum_{i=1}^{2d} e^{\frac{(x_0-x+e_i\beta_l\sqrt{d})^2}{s^2r-0.5\bar{A}_1 l}}$, e_i being the basis vectors in R^d and opposite ones by sign. The integration leads to

$$\int_X dy\ p_{k+1}(x^i_k, y)\phi_{m+1,m-k}(y) \leqslant C\phi_{m+1,m-k+1}(x^i_k).$$

For the second term in (34), apply the Chebyshev inequality and enequality (33):

$$\mathbf{E}_{\mathcal{F}_k}1_{\{|\bar{\eta}-1|>1/2\}} \leqslant 4\mathbf{E}_{\mathcal{F}_k}(\bar{\eta}-1)^2 \leqslant \frac{C}{M}\phi_{k+1,1}(x^i_k).$$

Thus, (32) is proven, which implies $d_{0n}(x_0) \leqslant C/M$. Finally, from (31) receive the assertion of the theorem:

$$\mathbf{E}\left(\check{Y}_0 - Y_0\right)^2 \leqslant N\sum_{n=0}^{N-1} d_{n0}(x_0) \leqslant C/M. \quad (35)$$

Proof of theorem 3. Introduce auxiliary payoff functions $\widetilde{f}_n(x) = f_n(x)/g_n(x)$ and consider the sequence

$$\widetilde{Y}_n(x) = \max\left(\widetilde{f}_n(x),\ \alpha_n(x)\int_X dy\ \widetilde{p}_{n+1}(x,y)\widetilde{Y}_{n+1}(y)\right), \quad (36)$$

where $\alpha_n(x) = \widetilde{g}_{n+1}(x)/g_n(x)$. One can construct the scheme analogous to (20) for the sequence (36):

$$Y'_n(i) = \max\left(\widetilde{f}_n(i), \alpha_n(i)\frac{\sum_j \widetilde{\rho}_{n+1}(i,j)\ Y'_{n+1}(j)}{\sum_j \widetilde{\rho}_{n+1}(i,j)}\right), \quad (37)$$

where $\widetilde{\rho}_n$ is defined by (4) with substitution $\widetilde{p}$ for p. Since $\widetilde{f}_n(x) \leqslant F$ and $\alpha_n(x) \leqslant G$, the theorem 2 may be applied to the scheme (37). Thus $\mathbf{E}(Y'_0(\xi_0) - \widetilde{Y}_0(\xi_0))^2 \leqslant CM^{-1}$. On the other hand, it is easy to verify that $\widetilde{Y}_n(x) = Y_n(x)/g_n(x)$ and $Y'_n(i) = \check{Y}_n(i)/g_n(i)$. The proof is complete.

VI. Conclusion and Future Work

It is shown in the paper that a simple modification from scheme (5) to schemes (20), (37) allows to prove the finiteness of the variance and, thus, consistency of the estimators in the general diffusion model. Besides, unpleasant bias vanishes, which allows to reduce the number of nodes in the mesh. The next step may be the extension of results to the jump-diffusion model, which is important for applications.

Acknowledgment

This work was supported by RFBR grant 14-01-00271.

References

[1] I. Karatzas, “On the pricing of American options. Applied Mathematics and Optimization,” Applied Mathematics and Optimization, vol. 17, 1988, pp. 37–60.

[2] M. Broadie and P. Glasserman, “A Stochastic Mesh Method for Pricing High-Dimensional American Options,” Journal of Computational Finance, vol. 7, 2004, pp. 35–72.

[3] A. Avramidis and H. Matzinger, “Convergence of the stochastic mesh estimator for pricing Bermudan options,” Journal of Computational Finance, vol. 7, 2004, pp. 73–91.

[4] V. Bally, L. Caramellino, and A. Zanette, “Pricing and Hedging American Options by Monte Carlo Methods Using a Malliavin Calculus Approach,” Journal of Monte Carlo Methods and Applications, vol. 11, 2005, pp. 97–134.

[5] Y. N. Kashtanov, Financial Mathematics Models and Statistical Simulation. VVM, St.-Petersburg, 2013, ISBN: 978-5-9651-0741-4.

[6] G. Barone-Adesi and R. E. Whaley, “Efficient Analitic Approximation of American Option Values,” Journal of Finance, vol. 42, 1987, pp. 301–320.

[7] B. von Bahr and C.-G. Esseen, “Inequalities for the rth absolute moment of a some of random variables, $1 \leqslant r \leqslant 2$,” Ann. Math. Statist, vol. 36, 1965, pp. 299–303.

ISBN: 978-1-61208-371-1

Multiple Convolution Neural Networks for an Online Handwriting Recognition System

Phạm Việt Dũng
Computer Network Centre
Vietnam Maritime University
Haiphong City, Vietnam
e-mail: vietdungiitb@vimaru.edu.vn

***Abstract*— This paper focuses on a specific word recognition technique for an online handwriting recognition system which uses Multiple Component Neural Networks (MCNN) as the exchangeable parts of the classifier. As the most recent of approaches, the system proceeds by segmenting handwriting words into smaller pieces (usually characters), which are recognized separately. The recognition results are then a composition of individually recognized characters. They are sent to the input of a word recognition module to choose the most suitable one by applying some dictionary search algorithms. The proposed classifier overcomes obstacles and difficulties of traditional ones to large character classes. Furthermore, the proposed classifier also has expandable capacity, which can recognize other character classes by adding or changing component networks and built-in dictionaries dynamically.**

Keywords- online handwriting; recognition; convolution; neural network.

I. INTRODUCTION

Nowadays, Touch User Interfaces (TUI) devices are becoming increasingly popular and already play an important role in human-computer interaction. Tablets, smartphones and TUI computers accepting finger touch or pen based input are becoming a crucial part of many everyday life activities. Using fingers or a pen as an input device covers more and improves many functions as compared to the conventional mouse and keyboard. One major advantage of the pen over the mouse is the fact that it is a natural writing tool for humans while the computer's mouse is very hard to use as a writing tool. However, it needs a reliable transformation of handwritten text into a coding that can be directly processed by a computer, e.g., ASCII [6]. A traditional transformation model usually includes a preprocessor which extracts each word from image or input screen and divides it into segments. A neural network classifier then finds the likelihoods of each possible character class given the segments. These likelihoods are used as the input to a special algorithm which recognizes the entire word. In recent years, research in handwriting recognition has advanced to a level that makes commercial applications. Nevertheless, significant disadvantages of such single neural network classifiers are their complexity in big network organizations and expandable capacity [4].

A highly reliable recognition rate neural network can be easily built to recognize a small character class, but not larger ones. The larger inputs and outputs increase the number of layers of the neural network, neurons, and connections. Hence, it makes the network training process more difficult and especially the recognition rate, which is significantly decreased [8]. Furthermore, a single neural network classifier only works on a particular character class. It is not exchangeable or expandable to recognize additional character classes without recreating or retraining the neural network.

This paper presents a new online handwriting recognition system that is based on multiple Convolutional Neural Networks (CNNs). As is well known, CNNs are efficient for various applications [9]. They are presented in Section 2. Section 3 presents a new classifier that includes a collection of very high recognition rate component CNNs working together. Each CNN can only correctly recognize a part of the big character class (digits, alphabet, etc.), but when these networks are combined by programing algorithms, they can create a flexible classifier which can recognize differential big character classes by simply adding or removing component CNNs and language dictionaries. The computer simulation results are shown in Section 4. Finally, the conclusion is presented in Section 5.

II. CONVOLUTION NEURAL NETWORK

CNNs are a special kind of multi-layer neural networks. Like almost every other neural network~~,~~ they are trained with a version of the back-propagation algorithm. Where they differ is in their architecture. CNNs are designed to recognize visual patterns directly from pixel images with minimal preprocessing. They can recognize patterns with extreme variability (such as handwritten characters), and with robustness to distortions and simple geometric transformations.

The LeNET 5 (see Fig. 1) for handwritten digit recognition has allowed a reliable recognition rate of up to 99% to MNIST dataset [1].

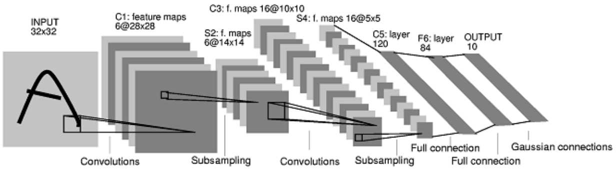


Figure 1. A Typical Convolutional Neural Network (LeNET 5) [1]

The input layer is of size 32 x32 and receives the gray-level image containing the digit to recognize. The pixel intensities are normalized between −1 and +1. The first hidden layer C1 consists of six feature maps, each having 25 weights, constituting a 5x5 trainable kernel, and a bias. The values of the feature map are computed by convolving the input layer with the respective kernel and applying an activation function to obtain the results. All values of the feature map are constrained to share the same trainable kernel or the same weights values. Because of border effects, the feature maps' size is 28x28, smaller than the input layer.

Each convolution layer is followed by a sub-sampling layer which reduces the dimension of the respective convolution layer's feature maps by a factor of two. Hence, the sub-sampling maps of the hidden layer S2 are of size 14x14. Similarly, layer C3 has 16 convolution maps of size 10x10 and layer S4 has 16 sub-sampling maps of size 5x5. The functions are implemented exactly as layers C1 and S2 perform. The S4 layer's feature maps are of size 5x5, which is too small for a third convolution layer. The C1 to S4 layers of this neural network can be viewed as a trainable feature extractor. Thereafter, a trainable classifier is added to the feature extractor, in the form of 3 fully connected layers (a universal classifier).

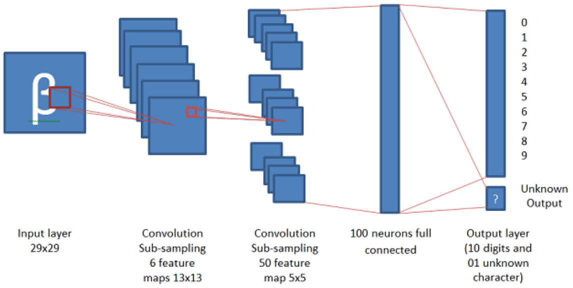


Figure 2. A convolution network based on P. Simard's model [3]

Another model of CNN for handwritten digit recognition that integrates convolution and sub-sampling processes into a single layer also grants recognition rates of over 99% [3]. This model, presented in Fig. 2, extracts simple feature maps at a higher resolution, and then converts them into more complex feature maps at a coarser resolution by sub-sampling a layer by a factor of two. The width of the trainable kernel is chosen to be centered on a unit (odd size), to have sufficient overlap to not lose information (3 would be too small with only one unit overlap), but yet not to have redundant computation (7 would be too large, with 5 units or over 70% overlap). Padding the input (making it larger so that there are feature units centered on the border) does not improve performance significantly. With no padding, a sub-sampling of two, and a trainable kernel of size 5x5, each convolution layer reduces the feature map size from n to (n-3)/2. Since the initial MNIST input used in this model is of size 28x28, the nearest value which generates an integer size after 2 layers of convolution is 29x29. After 2 layers of convolution, the feature of size 5x5 is too small for a third layer of convolution. The first two layers of this neural network can be viewed as a trainable feature extractor after which a trainable classifier is added to the feature extractor in the form of 2 fully connected layers (a universal classifier).

III. MULTIPLE COMPONENT NEURAL NETWORKS CLASSIFIER

CNN can secure a significantly higher recognition rate than traditional multilayer perceptron neural network to small character classes such as digits [3] or the English alphabet (26 characters). However, creating a larger neural network that can reliably recognize a bigger collection (62 characters) is still a challenge. Finding an optimized and large enough network becomes more difficult. Training networks by large input patterns takes a much longer time. Convergent speech of the network is slower and the accuracy rate is significantly decreased because of bigger badly written characters, similar and confusable characters etc.

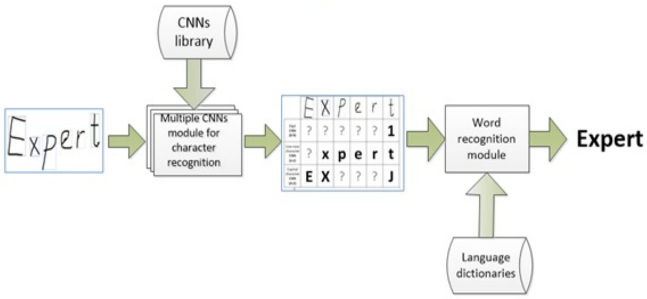
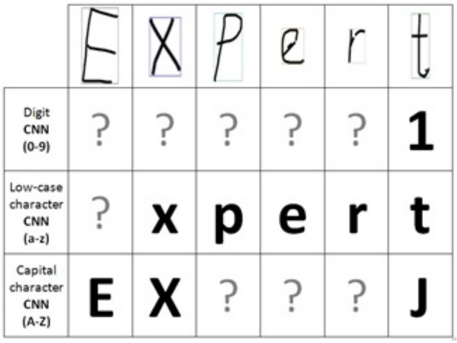


Figure 3. A MCNNs online handwriting recognition system

The proposed solution to the above problem is replacing a unique complex neural network with multiple smaller networks which have a high recognition rate to these own output sets [4]. Fig. 3 illustrates explicitly the working process of this new system. Each component network has an additional unknown output (unknown character) beside the official output sets (digit, letters, etc.). This means that, if the input pattern is not recognized as a character of official outputs, it will be understood as an unknown character.

The character recognition module of the classifier is a collection of multiple component neural networks, which work simultaneously with the input patterns. A handwritten word is pre-processed by segmenting the isolated character visual patterns [4]. These patterns are then given to the inputs of all the component neural networks, which will recognize likelihoods of each character class. A visual pattern can be recognized by one, some, or all component networks because there are several similar characters in different classes. If a network cannot recognize the pattern as a likelihood of its own character class, it will return an unknown character (null character). The module's output result, presented in Fig. 4, is a table of possible characters which is composed of possible words, such as "Exper1, Expert, ExperJ, EXper1, EXpert, EXperJ" in the above example. Unknown characters (null characters) are not used in word composition. These words are then given to the next word recognition module in turn to choose the most suitable one to become the output of overall classifier. In this example the word "Expert"will be chosen.



	E	X	P	e	r	t
Digit CNN (0-9)	?	?	?	?	?	1
Low-case character CNN (a-z)	?	x	p	e	r	t
Capital character CNN (A-Z)	E	X	?	?	?	J

Figure 4. Output of MCNNs classifier module

The algorithm of word composition uses in character recognition module:

Global variables:

- *charMatrix = List<List<Char>> {{E},{x,X},{p},{e},{r},{1,t,J}} // character table*
- *words =List<string> //list of composed word.*
- *startIndex: default is 0*
- *baseWord: default is "*

```
void GetWords(int startIndex, String baseWord)
 {
        String newWord = "";
        if (startIndex == charMatrix.Count - 1)
        {
           for (int i = 0; i < charMatrix[startIndex].Count; i++)
           {
              newWord = String.Format("{0}{1}", baseWord,
charMatrix[startIndex][i].ToString());
              words.Add(newWord);
           }
        }
        else
        {
           for (int i = 0; i < charMatrix[startIndex].Count; i++)
           {
              newWord = String.Format("{0}{1}", baseWord,
charMatrix[startIndex][i].ToString());
              int newIndex = startIndex + 1;
              GetWords(newIndex, newWord);
           }
        }
 }
```

Word recognition module is in fact a spell checker which uses several dictionary search algorithms and word correction techniques to get the best meaning of the word. All possible words from character recognition modules are given to the dictionary search sequentially. If one of the words is found in built-in dictionaries, it will become the output word of the classifier. Otherwise, some other word correction technique will be applied for choosing the most suitable word in automatic mode or showing a list of similar words to user in manual mode. Some of these techniques are:

- Swap each character one by one and try all the chars in its place to see if that makes a good word.

```
private bool ReplaceChars(String word, out String result)
{
    result = "";
    bool isFoundWord = false;
    foreach (WordDictionary dictionary in Dictionaries)
   {
       ArrayList replacementChars =
dictionary.ReplaceCharacters;
       for (int i = 0; i < replacementChars.Count; i++)
      {
          int split = ((string)replacementChars[i]).IndexOf(' ');
          string key = ((string)replacementChars[i]).Substring(0,
split);
          string replacement =
((string)replacementChars[i]).Substring(split + 1);
          int pos = word.IndexOf(key);
          while (pos > -1)
          {
              string tempWord = word.Substring(0, pos);
              tempWord += replacement;
              tempWord += word.Substring(pos + key.Length);
              if (this.TestWord(tempWord))
               {
                     result = tempWord.ToString();
                     isFoundWord = true;
                     return isFoundWord;
               }
                pos = word.IndexOf(key, pos + 1);
               }
            }
          }
          return isFoundWord;
       }
```

- try swapping adjacent chars one by one.

```
private bool SwapChar(String word, out String result)
{
          result = "";
          bool isFoundWord = false;
          foreach (WordDictionary dictionary in Dictionaries)
          {
             for (int i = 0; i < word.Length - 1; i++)
             {
```

```
                        StringBuilder tempWord = new
StringBuilder(word);
                        char swap = tempWord[i];
                        tempWord[i] = tempWord[i + 1];
                        tempWord[i + 1] = swap;
                        if (this.TestWord(tempWord.ToString()))
                        {
                            result = tempWord.ToString();
                            isFoundWord = true;
                            return isFoundWord;
                        }
                    }
                }
                return isFoundWord;
 }
```

- Try inserting a new character before every letter.

```
private bool ForgotChar(String word, out String result)
        {
            result = "";
            bool isFoundWord = false;
            foreach (WordDictionary dictionary in Dictionaries)
            {
                char[] tryme =
dictionary.TryCharacters.ToCharArray();
                for (int i = 0; i <= word.Length; i++)
                {
                    for (int x = 0; x < tryme.Length; x++)
                    {
                        StringBuilder tempWord = new
StringBuilder(word);
                        tempWord.Insert(i, tryme[x]);
                        if (this.TestWord(tempWord.ToString()))
                        {
                            result = tempWord.ToString();
                            isFoundWord = true;
                            return isFoundWord;
                        }
                    }
                }
            }
            return isFoundWord;
        }
```

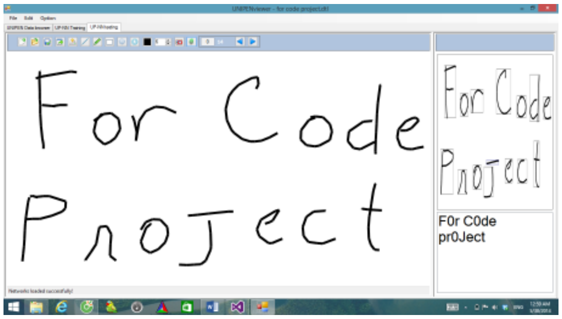
- Split the string into two pieces after every char. If both pieces are good words make them a suggestion etc.

```
private bool TwoWords(String word, out String result)
        {
            result = "";
            bool isFoundWord = false;
            for (int i = 1; i < word.Length - 1; i++)
            {
                string firstWord = word.Substring(0, i);
                string secondWord = word.Substring(i);
                if (this.TestWord(firstWord) &&
this.TestWord(secondWord))
                {
                    string tempWord = firstWord + " " + secondWord;
                    result = tempWord;
                    isFoundWord = true;
                    return isFoundWord;
                }
            }
            return isFoundWord;
        }
```

By using multiple language dictionaries simultaneously in the spell checker, the proposed classifier can correctly recognize different languages, if we supplement component neural networks being trained these languages' character classes.

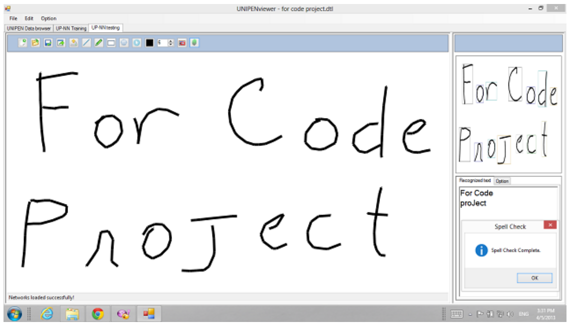
IV. EXPERIMENTS AND RESULTS

The demo program uses three well trained component CNNs which can recognize 97 % of the digit class and 90% of the upper or the lower alphabet classes of the UNIPEN online handwriting dataset [12], respectively, to identify 62 English characters set. The initial experiment, which took 450 words from 45 students who were required to carefully write 10 different words to a windows 8 touch screen device, had shown an extremely satisfactory results. Without word recognition module, the system could not recognize words properly due to the randomized of the process when choosing a possible word from a collection of recognized characters at MCNN module's outputs. The rate of exactly recognized words was lower than 30%, although the system could identify most of written characters. The main reason for this low rate was a misunderstanding of the system to similar characters "o","0" or "i","I", "l","1", etc.


UNIPENviewer - for code project.dtl
File
Edit
Option
UNIPEN Data browser
UP-NN Training
UP-NNtesting
Networks loaded successfully!

F0r C0de
pr0Ject

Figure 5. Recognized words without a spell checker



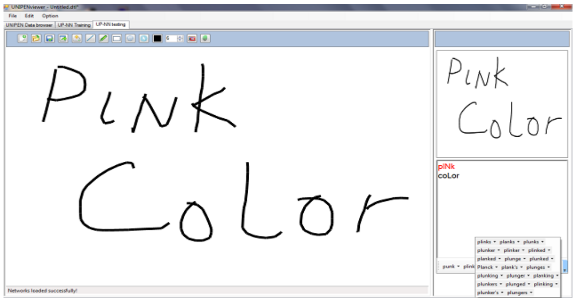
For Code
proJect

Spell Check

Spell Check Complete.

OK

Figure 6. Recognized words with a spell checker



pINk
coLor

Networks loaded successfully!

Figure 7. Word correction mode

The recognized word rate dramatically increased when the word recognition module was activated. 352 over 450 words in total were recognized exactly. The result should be higher if the system used correction function for faulty words. Figures 5 to 7 show different results to an image when the system works in different modes.

V. CONCLUSIONS AND FUTURE WORK

Although there are many limits such as writing tools, character segmentation techniques, the experimental program has shown satisfactory results to the proposed approach. Instead of using a traditional neural network, the new system uses several smaller well trained networks to recognize a large character set. The primary advantage of this approach is that the system can adapt to recognize different languages by changing component networks and dictionaries. Also, the new classifier will overcome obstacles and difficulties of traditional ones when faced with large character classes. The full project and source code can be downloaded from [13].

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition", Proceedings of the IEEE, vol. 86, no. 11, Nov. 1998, pp. 2278-2324.

[2] Y. LeCun, L. Bottou, G. Orr and K. Muller, "Efficient BackProp", in Neural Networks: Tricks of the trade, (G. Orr and Muller K., eds.), 1998.

[3] P. Y. Simard, D. Steinkraus and J. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," International Conference on Document Analysis and Recognition (ICDAR), IEEE Computer Society, Los Alamitos, 2003, pp. 958-962.

[4] P. V. Dung, “Online handwriting recognition using multi convolution neural networks”. In Proceedings of The Ninth International Conference on Simulated Evolution And Learning (SEAL 2012), Hanoi, Vietnam, December 2012, pp. 310-319.

[5] F. Lauer, Ching Y. Suen and G. Bloch, "A Trainable Feature Extractor for Handwritten Digit Recognition", Elsevier Science, February 2006.

[6] I. Guyon, L. Schomaker, R. Plamondon, R. Liberman and S. Janet, “Unipen project of on-line data exchange and recognizer benchmarks”. In proceedings of the 12th International Conference on Pattern Recognition, ICPR’94, Jerusalem, Israel, October 1994. IAPRIEEE, pp. 29–33.

[7] L. Vuurpijl, R. Niels, M. V. E. Nijmegen, “Verifying the UNIPEN devset”. In proceedings of Ninth International Workshop on Frontiers in Handwriting Recognition, IWFHR-9 2004, pp. 586 – 591.

[8] M. Parizeau, A. Lemieux, and C. Gagné, “Character Recognition Experiments using Unipen Data”. Parizeau & al., Proc. of ICDAR 2001, September 10-13. 2001 , pp. 481 – 485.

[9] List of publications by Dr. Yann LeCun. http://yann.lecun.com/exdb/publis/index.html

[10] Mike O'Neill, “Neural Network for Recognition of Handwritten Digits”. http://www.codeproject.com/Articles/16650/Neural-Network-for-Recognition-of-Handwritten-Digi

[11] Modified NIST ("MNIST") database (11,594 KB total). http://yann.lecun.com/exdb/mnist/index.html [retreived July 2014]

[12] The UNIPEN Project. http://unipen.nici.kun.nl/ [retrived July 2014]

[13] Codeproject website:
http://www.codeproject.com/Members/Vietdungiitb
[retrieved July 2014]

The Influence of Lateral, Roll and Yaw Motion Gains on Driving Performance on an Advanced Dynamic Simulator

Florian Savona, Anca Melania Stratulat, Emmanuelle Diaz, Vincent Honnet, Gilles Houze, Philippe Vars, Stéphane Masfrand, Vincent Roussarie
Sciences Cognitives et Facteurs Humains
PSA Peugeot Citroën
Vélizy-Villacoublay, France
e-mail : {florian.savona, ancamelania.stratulat, emmanuelle.diaz, vincent.honnet, gilles.houze, philippe.vars, stephane.masfrand, vincent.roussarie}@mpsa.com

Florian Savona, Christophe Bourdin
Aix-Marseille Université, CNRS
ISM UMR 7287
Marseille, France
e-mail : {florian.savona, christophe.bourdin}@univ-amu.fr

***Abstract*—The present study investigates the respective roles of lateral acceleration, and lateral, roll and yaw motions for self-motion perception and cornering behavior on a dynamic driving simulator. A recent study advises the use of motion gains (in the range 0.4 to 0.75) on these three components in order to improve self-motion perception. However, the role of each component in self-motion perception has not been individually addressed and the same motion gain is proposed for all components, independently of the level of acceleration. The aim of the present study is to extend this previous result by systematically reassessing the motion gains for the three lateral motion components for several levels of acceleration. A slalom task was chosen (with the level of lateral acceleration modified by changing the distance between posts) so that cornering behavior and self-motion perception could be assessed for various settings of the three parameters. The main results suggest that 1/ lateral motion gain should be decreased when lateral acceleration is increased; 2/ roll motion gain should be set to 1 to improve and facilitate driving perception and performance and 3/ the yaw component has a more controversial role but it seems to facilitate driving control without influencing motion perception. In conclusion, this study shows that the three motion components generally used to simulate lateral acceleration should be set individually and that use of the same motion gain for all three is not the best solution for improving the realism of the simulator. Therefore, it is proposed that each parameter be dynamically set based on the driving conditions.**

Keywords-Lateral acceleration; motion gains; driving performance; tilt-coordination.

I. INTRODUCTION

On dynamic driving simulators, motion perception is produced by stimulating the vestibular and somatosensory systems in addition to the visual system [1]. However, the intricacy of the multisensory stimulations undergone when driving a car makes the optimization of motion based simulators quite complex. For instance, it has already been shown that the motion on a driving simulator is overestimated when simulated at 1-to-1 rate [2]–[4]. In order to avoid this overestimation, some technical tricks are used, like the scale factor, called gain, and/or a combination of tilt and translation, called tilt-coordination [3]. However, both the gain and tilt-coordination needed to reproduce a positive or negative acceleration (e.g., take-off or braking) are highly dependent on the level of the simulated acceleration [4][5].

For turning manoeuvers, the control of the simulator appears to be more complex than for longitudinal manoeuvers because, in addition to lateral acceleration, there are also the yaw and the roll motions of the car that have to be simulated. However, the main sensory information on which the driver depends in making the manoeuvers is lateral acceleration. Indeed, the driver controls the speed and the trajectory of the car to keep this acceleration in a comfortable range and to ensure a safety margin [6][7]. In most dynamic driving simulators, the simulation of lateral acceleration is produced by using the tilt-coordination technique (lateral translation and lateral tilt). However, during cornering, the car is subject not only to a lateral linear acceleration, but also to rotational motions, such as yaw and roll. These motion components are also taken into account for driving simulation and they are highly dependent on the steering behavior of the car during cornering. Therefore, Berthoz et al. (2013) [8] proposed that motion gains (for lateral and rotational acceleration) should be within the range 0.4-0.75. One limitation of this study is that the gain for linear translations, roll and yaw and their interactions were not systematically varied for different levels of acceleration.

To go further this limitation, the present study, conducted on PSA’s (Peugeot Society Automobile) dynamic driving simulator SHERPA² [9], is focused on cornering manoeuvers. It aims at systematically reassessing the motion gains for the three lateral motion components (lateral, yaw and roll motions) for several levels of lateral acceleration. In order to evaluate the individual effects of the three parameters on driving behavior, a slalom driving task was selected. Through subjective and objective analyses, we tried to identify and quantify the major sources of motion perception and driving performance in cornering, and to identify the best set of parameters for each level of acceleration to simulate. More precisely, the aim is to define a mapping of motion gain set-ups to improve the realism of the driving simulation for a wider range of lateral accelerations. It is hypothesized that the motion gains for the different parameters are not necessarily linked [10][11], and that they could be different depending on the level of lateral acceleration.

The paper will be structured as follows: in Section II "Methods", the experiment is presented (participants, devices, scenario etc.). Section III "Results" gives the results of the study. Finally, Section IV "Discussion of Results and Conclusion" is a discussion of the results of the study and their applications.

II. METHODS

A. Participants

27 volunteers (2 women and 25 men), aged between 22 and 49 (mean age: 28) participated in the study. All were PSA (Peugeot Society Automobile) employees who volunteered for the study, and none had significant experience of the simulator (average dynamic driving simulator experience less than 1.5 hours).

B. Experimental Devices

SHERPA² is a dynamic driving simulator equipped with a hexapod and an X-Y platform. The cell placed on the hexapod contains a fully-equipped half-cab Citroen C1 (2 front adjustable seats, seat belts, steering wheel, pedals, gearbox, rearview mirror and side-view mirrors) where the driver sits. The motion limits of the hexapod are ±30 cm, ±26.5 cm and ±20 cm, on X, Y and Z respectively [11]. Rotational movements are limited to ±18 deg, ±18 deg and ±23 degrees, on pitch, roll and yaw respectively. The X-Y motion platform can reproduce linear movements of 10 and 5 meters. The maximum longitudinal and lateral acceleration is 5 m/s², and is actually produced by a combination of tilt and translation (in this paper, lateral tilt/translation is called "lateral motion").

C. Experimental Scenario

The vehicle dynamics model (car dynamics and audio) selected for the present experiment was a Peugeot 208 1.4 HDi. The visual scene consisted of a straight two-lane road (road width: 8m). Guardrails were placed at both sides of the road to delimit the allowed maximum excursion of the car. The slalom driving scenario consisted of a series of 8 posts a constant distance apart (for a given level of acceleration). In addition, multiple mini-cones were used to represent the optimal sinusoidal pathway and help the subjects to perform the task [9]. The posts were alternately placed 0.9 m to the right and left side of the road centerline.

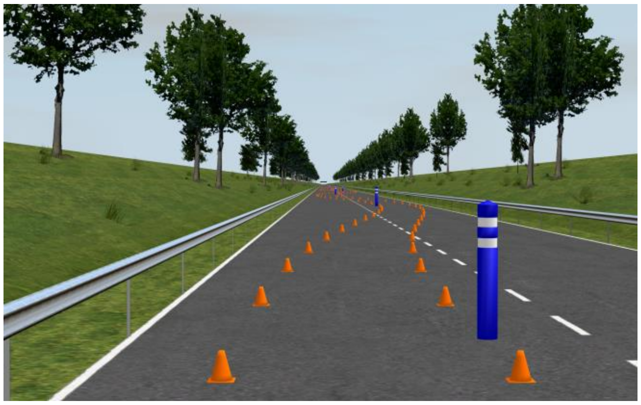


Figure 1. Visual environment of slalom task.

The velocity of the car was set at 70 km/h. Then, by adjusting the distance separating two posts, various theoretical lateral accelerations were imposed. Hence, this gave three different slalom scenarios leading to three theoretical lateral accelerations, of 1, 2 and 4 m/s², corresponding to post spacings of 86.39, 61.09 and 43.19 meters, respectively. The equation enabling calculation of the theoretical lateral acceleration was borrowed from Grácio, Wentik and Païs (2011) [12].

D. Task

Drivers were asked to perform a slalom course on the dynamic driving simulator by following the mini-cone path, without touching any posts or leaving the road. The run was performed in cruise control at a constant speed of 70 km/h.

E. Experimental Design

For each level of lateral acceleration (1, 2 and 4 m/s²), the motion gains of the 3 motion components (lateral motion; yaw and roll) were individually varied, leading to a total of 25 different conditions (see TABLE I).

The motion conditions varied according to different gains applied to the three simulator motion components. Slaloms 1, 2 & 3 respectively correspond to 1, 2 and 4 m/s² acceleration levels. Condition 20 corresponds to the current SHERPA² configuration. Each participant performed 3 repetitions per condition for a total of 75 trials divided into two sessions to avoid fatigue. The trials were organized using a central composite experimental design [13]. The choices of motion gains were made taking into account the physical limitations of the simulator (position, speed, linear and angular acceleration).

ISBN: 978-1-61208-371-1

TABLE I. THE LIST OF THE 25 MOTION CONDITIONS TESTED FOR EACH SPECIFIC SLALOM.

	Lateral Motion Acceleration Gain			**Roll Angle Gain**	**Yaw Acceleration Gain**
Slalom	**1**	**2**	**3**	**1, 2 & 3**	**1, 2 & 3**
Condition					
1	**0.2**	**0.2**	**0.2**	**0.2**	**0.2**
2	**0.8**	**0.8**	**0.6**	**0.2**	**0.2**
3	**0.2**	**0.2**	**0.2**	**0.8**	**0.8**
4	**0.8**	**0.8**	**0.6**	**0.8**	**0.8**
5	**0.2**	**0.2**	**0.2**	**0.2**	**0.2**
6	**0.8**	**0.8**	**0.6**	**0.2**	**0.8**
7	**0.2**	**0.2**	**0.2**	**0.8**	**0.8**
8	**0.8**	**0.8**	**0.6**	**0.8**	**0.8**
9	**0**	**0**	**0**	**0.5**	**0.5**
10	**1**	**1**	**0.8**	**0.5**	**0.5**
11	**0.5**	**0.5**	**0.4**	**0**	**0.5**
12	**0.5**	**0.5**	**0.4**	**1**	**0.5**
13	**0.5**	**0.5**	**0.4**	**0.5**	**0**
14	**0.5**	**0.5**	**0.4**	**0.5**	**1**
15	**0.5**	**0.5**	**0.4**	**0.5**	**0.5**
16	**0.5**	**0.5**	**0.4**	**0.5**	**0.5**
17	**0.5**	**0.5**	**0.4**	**0.5**	**0.5**
18	**0.5**	**0.5**	**0.4**	**0.5**	**0.5**
19	**0.5**	**0.5**	**0.4**	**0.5**	**0.5**
20	**-1**	**-1**	**-1**	**-1**	**-1**
21	**0**	**0**	**0**	**0**	**0**
22	**1**	**1**	**0.8**	**1**	**1**
23	**1**	**1**	**0.8**	**0**	**0**
24	**0**	**0**	**0**	**1**	**0**
25	**0**	**0**	**0**	**0**	**1**

During the first session, the participants started with a simulator familiarization phase (10 min of rural driving) and a slalom learning phase (one trial for each slalom without motion of the simulator). This first session was followed by twenty-five trials of one slalom (same level of acceleration). The second session, performed four hours later, included another slalom learning phase along with the 50 remaining trials. The order of presentation of the three different slaloms was balanced over the total panel of participants. The order of the conditions was chosen using a Williams Latin Square, to balance the effect of the position and carryover effect between samples. The use of a central composite experimental design meant that the maximum information could be obtained in a minimum duration, and a model estimating nonlinear effects constructed. Furthermore, at the end of each trial, the participants answered a couple of questions to provide information about their subjective perception of the realism of the vehicle's behavior and the facility of the task. Two 11-point qualitative scales were used, ranging from 0 (“Not Realistic” or “Not Easy) to 10 (“Very Realistic” or “Very Easy”). In addition, motion sickness level was monitored throughout the experiment via a Motion Sickness Susceptibility Questionnaire (MSSQ) [10].

F. Data Analysis

During the driving task, some dynamic variables were recorded from the vehicle and simulator (e.g., lateral acceleration, steering wheel angle, lateral position). All these measurements were used to conduct an objective analysis of driver behavior. The Steering Wheel Reversal Rate (SWRR) was calculated from steering wheel angle. SWRR is a performance indicator that quantifies the amount of steering correction, and means that the effort required to accomplish a certain task can be determined [14]. This metric measures the frequency of steering wheel reversals larger than a finite angle, or gap. The magnitude of this gap, the gap size, is thus a key parameter for this metric [15]. In the present study, the number of reversals per slalom was counted. The steering signal was filtered using a second-order low-pass Butterworth filter with a cutoff frequency depending on the slalom level, specifically, 0.6, 2 and 5 Hz for the 1, 2 and 4 m/s² acceleration levels respectively. The algorithm for detecting the reversal was extracted from “Reversal Rate 2” in Östlund's study (2005) [15], and a difference greater than or equal to 2° (gap size) indicates one reversal.

Driving accuracy was quantified as lateral deviation from the reference trajectory (center of the mini-cone path) and computed as Root Mean Squared Error (RMSE) of the vehicle path.

The subjective and objective data were analyzed using the NEMRODW [9] software package. For each subjective and objective variable, data was collected from each participant and a principal component analysis (PCA) was performed, in order to determine if there was consensus among subjects; if no consensus was found, an ascending hierarchical classification was performed. Afterwards, a model was constructed (using NEMRODW) so that nonlinear effects and the best set of parameters could be estimated for a specific slalom level. The model contains first and second order coefficients on the three motion components. From these coefficients, statistical analyses were performed using multiple linear regressions in order to determine significant coefficients.

III. RESULTS

A. Subjective Analysis

1) Motion Sickness

During the experiment, four subjects felt motion sickness and were not able to finish all experimental conditions (Misery Score ≥ 6). Three of these participants felt motion sickness during the highest slalom level and with the highest lateral motion gains (Condition 10, 22 or 23 in TABLE I). The remaining twenty-three subjects were able to conduct the experiment without serious motion sickness (average Misery Score = 0.78 ± 1.2).

2) Realism of Vehicle Behavior

According to the PCA, no consensus was found among participants, so the data was centered, and a hierarchical clustering performed to identify homogeneous groups of subjects. The results of this analysis identified 2 groups (G1 and G2). The experimental results for the two groups were analyzed separately. The analyses of model's coefficients were performed to determine the optimal motion configuration. The coefficients are labeled as follows: "B0" is the model's constant, "B1" is the linear coefficient applied to lateral motion gain, "B2" is the linear coefficient applied to roll gain, and "B1-1" is the squared coefficient of lateral motion gain.

As shown in the TABLE II, for the first slalom (1 m/s²), lateral motion was a significant factor for both groups, while roll motion was a significant factor only for the second group. This means that changing their values should modify the perceived realism of the simulator. The model coefficients are presented in TABLE II for both groups.

TABLE II. THE MODEL'S COEFFICIENTS AND THEIR SIGNIFICANCE FOR GROUPS G1 AND G2, REGARDING REALISM OF VEHICLE BEHAVIOR FOR THE FIRST SLALOM (LATERAL ACCELERATION LEVEL OF 1 M/S²).

Name	G1 Coeff	Sign	G2 Coeff	Sign
B0	7.756	<0.01***	6.889	<0.01***
B1	-0.374	0.518***	1.174	<0.01***
B2	-0.081	48.5	0.405	2.13*
B1-1	-1.227	<0.01***	-4.5	0.05***

According to the answers of group 1 (G1), the experimental model assesses as more realistic a motion configuration with: lateral motion gain = 0.5, roll motion gain = 1, and yaw motion gain = 0.

According to the answers of group 2 (G2), the best set of parameters for realism is: lateral motion gain = 0.85, roll motion gain = 1, and yaw motion gain = 0. Figure 2 shows a 2D representation of the experimental model of Lateral and Roll motion gains for vehicle behavior realism in the first slalom and according to G2. In Figure 2, yaw motion gain is set at 0. As can be seen on this figure, the quality of realism grows with the amplitude of lateral motion gain, until a maximum at 0.85. This figure also shows the importance of roll motion gain (see TABLE II), which give the best result with a value of 1.

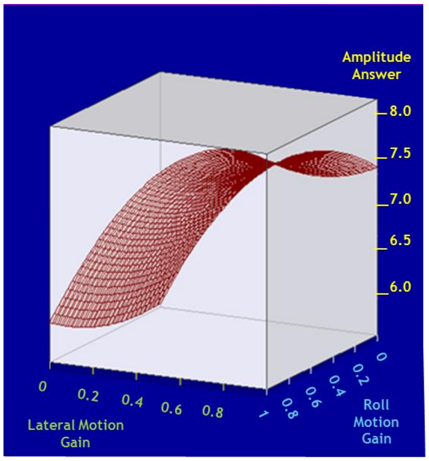


Figure 2. 3D representation of the experimental model for realism of vehicle behavior, for the first slalom and second group.

For the second slalom, the only significant factor for both groups was lateral motion (p<0.01). The lateral motion gain should be set to 0.4 and 0.7 for G1 and G2 respectively, to optimize the realism. In the third slalom, and for G1, the three motions were significant factors (p<0.01). For best realism, lateral, roll and yaw motion gain should be set to 0.25, 1 and 0 respectively. For G2, only lateral motion gain was a significant factor (p<0.01), and should be set to 0.5.

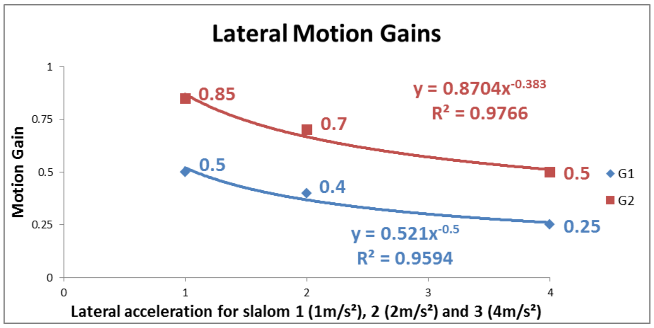


Figure 3. Best lateral motion gains for the two groups and the three slaloms.

Figure 3 presents the most realistic lateral motion gains, according to both groups. Lateral motion gains are digressive (reducing with increased acceleration) for both groups. Furthermore, for both groups, a roll motion gain of 1 always gives the best result in all slaloms.

3) Facility of Achieving Slalom

In the first slalom, no difference was found between all configurations. The first slalom was certainly very easy, and so drivers did not need external help to perform the task, and consequently, did not feel perturbed by the motion of the simulator.

For the second and the third slaloms, PCA analysis yielded a consensus between participants. Consequently, participants were included in the same group for subsequent analysis and computation of the model. The only significant factor in the second and the third slaloms was lateral motion (p<0.01). Contrary to the first slalom, participants found the second and the third slaloms less easy to perform, notably when lateral motion gain exceeded 0.2 in the second slalom and 0 in the third.

The best motion gains for perceived facility are presented in TABLE III.

TABLE III. BEST MOTION GAINS FOR ALL SLALOMS, REGARDING FACILITY OF ACHIEVING SLALOM.

	Lateral Motion Gain	Roll Gain	Yaw Gain
Slalom 1	**0 to 1**	**0 to 1**	**0 to 1**
Slalom 2	**0.2**	**0.3**	**0**
Slalom 3	**0**	**1**	**1**

Facility depends on slalom level, hence the motion gains, in particular lateral gain, should be adapted as a function of slalom level.

B. Objective Analysis

1) Steering-Wheel Reversal Rate

The PCA revealed a consensus among the participants, for all slalom levels. Hence, all 23 participants were analyzed together for the three slalom levels. For all slalom levels, the main significant factor was lateral motion gain.

For the first slalom level, the results showed that the number of reversals decreases with an increase in lateral motion gain, so that more steering corrections were required with low lateral motion gain. The analysis also suggests that roll motion gain has no effect on driving performance, although the best result was obtained for a roll motion gain of 0.

TABLE IV. BEST MOTION GAINS FOR THE SWRR VARIABLE FOR ALL SLALOMS.

	Lateral Motion Gain	Roll Gain	Yaw Gain
Slalom 1	**1**	**0**	**0**
Slalom 2	**0.5**	**1**	**1**
Slalom 3	**0.25**	**1**	**1**

Contrary to the first slalom, for the second and the third slaloms, the best model was obtained with a roll motion gain of 1. However, lateral motion gain has to be reduced when lateral acceleration to simulate increase. The yaw motion effect, although not significant in the model, seems to be best adjusted with a motion gain of 1 (see TABLE IV).

2) Path Root Mean Square Error

As for the previous variables, a single group was used for the model constructions. No difference was found between the motion configurations for the first and the second slalom. It is possible that the mini-cone path was helpful for accurate driving. Nonetheless, differences were found in the third slalom (4 m/s²). Again, the significant factor was lateral motion gain.

The experimental model found two configuration settings that lead to the same performance (see TABLE V).

TABLE V. BEST MOTION GAINS FOR THE RMSE VARIABLE FOR ALL SLALOMS.

	Lateral Motion Gain	Roll Gain	Yaw Gain
Slalom 1	**0 to 1**	**0 to 1**	**0 to 1**
Slalom 2	**0 to 1**	**0 to 1**	**0 to 1**
Slalom 3			
First configuration	**0.25**	**1**	**0**
Second configuration	**0.35**	**0 or 1**	**1**

Figure 4 shows the results for the second configuration (lateral motion gain of 0.35); it can been seen that the curve is mainly influenced by the amplitude of lateral motion gain. With low or high lateral motion gains, the drivers take wider trajectories; thus, extreme lateral motion gains decrease drive accuracy.

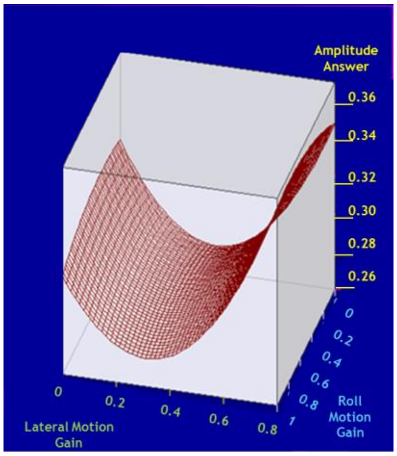


Figure 4. 3D representation of the experimental model for the RMSE variable, for the third slalom and the second configuration.

Thus, the results of objective variables seem to corroborate the results of subjective variable concerning the setting of lateral and roll motion gains, which is not the case for the yaw motion gain.

IV. DISCUSSION OF RESULTS AND CONCLUSION

The present research aims to reassess motion gains for the three lateral motion components (lateral, yaw and roll movements) for several levels of lateral acceleration. The slalom task has already been validated by several earlier studies. This research on dynamic driving simulators recommended use of unit motion gains in cornering, to improve motion perception and driving behavior [8][14].

However, these studies did not systematically investigate possible changes in the various motion gains depending on the levels of acceleration to be simulated. In fact, these studies only used equal motion gains for all three lateral components, and did not consider the effect of their decoupling on final performance and motion perception. In the present study, it was found that the three lateral motion components should not be set to the same gain, and should change as a function of lateral acceleration level.

A. *Motion Gains*

These results clearly show that, for all subjective and objective evaluations, lateral motion gain should be reduced as lateral acceleration increases. The equations from the Figure 3, which characterize lateral motion gain as function of lateral acceleration, could be used for simulator settings in order to improve realism. Nevertheless, these results seem to suggest there were two groups of drivers in the test population, preferring different lateral motion gains; it is possible that some of the participants assessed “comfort” instead of the realism. With regard to the subjective perception of task facility, a decrease in preferred lateral motion gain was observed as slalom level increased. Drivers found it more difficult to perform the second and the third slalom for configurations with lateral motion gains greater than 0.2 and 0 respectively. Greater physical fidelity (motions gains close to 1) and hence greater discomfort is probably the cause (large driver movements, lower facility of driving). Nonetheless, as showed by the objective analysis of steering-wheel corrections and lateral deviations, lateral motions gains below 0.2 are not recommended for maintaining accurate driving. Indeed, varying the amount of lateral motion in a simulated slalom affects driver performance. Except for the first slalom, where a lateral motion gain of 1 enabled optimal steering, driving accuracy for the two others slaloms was better with lower lateral motions gains. However, a lateral motion gain of 0 is not recommended for good driving performance and accuracy, as shown by the RMSE variable (see TABLE V).

With regard to roll motion gain, the experimental model showed that a roll motion gain equal to 1 is evaluated as being the most realistic situation, despite the fact that the two groups preferred different lateral motion gains. Although roll motion is not the most significant of the three lateral movements, it does enhance driving control for slaloms levels ≥2 m/s², notably with a gain of 1 (see TABLE IV).

Yaw motion was never a significant factor for vehicle behavior realism, except for G1 and the third slalom. Indeed, results on maneuverability show that drivers are sensitive to variations in yaw gains, and this motion can contribute to a change in the perceived maneuverability. However, as can been seen from the objective data from the second slalom, a better model is produced with a yaw motion gain of 1, probably because a very reactive car enables better handling (see TABLE IV).

B. *Lateral Motion*

As presented in Section 3, both subjective and objective variables show that lateral motion gain should be reduced when the lateral acceleration to simulate increases, so as to improve self-motion perception and driving performance. This very important result could be related to a previous result obtained in longitudinal acceleration, where the way the acceleration is produced (tilt/translation ratio) and the motion gain required to reproduce braking depend on the level of the acceleration [3]. In addition, previous research [8] has shown a decrease in steering corrections when lateral motion gain is increased. Nevertheless, they analyzed only one level of acceleration (1.2 m/s²). In the present study, it is shown that with increased lateral acceleration (i.e., slalom level), control of the vehicle demands more attention to accomplish the slalom with a unit or near a unit lateral motion gain.

The perception of simulated self-motion can tolerate significant discrepancies between the physical and visual motion cues [16]. Nevertheless, the tilt coordination technique was used to reproduce lateral acceleration; it is also possible than tilt is more easily perceived as lateral acceleration increases. A previous study [17] has shown that the limit of lateral tilt (perceived as a tilt and not as a lateral acceleration) is higher for active drivers than for passive passengers [2]. This research advised limiting tilt rate to 6°/s, twice the limit found for passive subjects. In our study, for the second and third slaloms and for the higher lateral motion gains, lateral tilt could reach 14° of inclination and an angular velocity of 12°/s (the limit set by our motion cueing algorithm or MCA). These magnitudes are higher than recommended by Nesti et al. [17], and higher than the threshold for roll tilt [18]; hence, the tilt of lateral motion is not perceived as lateral acceleration, but rather as a roll motion of higher amplitude than natural roll.

C. *Roll Motion*

A roll motion gain equal to 1 was perceived as the most realistic; moreover, the results from the RMSE and SWRR variables confirm this. Contrary to that for lateral motion gain, this result represents a new advance in the domain of simulation. It seems to confirm previous results obtained with expert drivers [11]. In the present study, this result has been extended to the wider population of “normal” drivers. A previous study [8] did not find this result. Indeed, as they used lateral motion gains of below 1, they never used a lateral motion gain of below 1 with a roll motion gain of 1. The driving simulator used in the present study reproduces exactly the roll angle and its derivates, and is temporally coherent with the visual roll. The absolute threshold of roll motion is around 2°/s [19]. Thus, a roll motion with a downscale factor is not necessarily felt by the driver. A roll motion supra-threshold has been shown to reduce latency of vection [20]. Hence, roll motion must remain at upper-thresholds to improve realism and driving performance. This means that a roll motion gain equal to 1 is advisable, regardless the level of lateral acceleration.

D. *Yaw Motion*

Surprisingly, yaw motion only slightly influences final perception, contrary to results obtained in a recent study showing that the presence of yaw motion in a curve affected driving behavior [21]. This different result could be explained by the fact that its intensity was probably not felt (yaw rate in slalom 1 (≤3°/s) or was masked by the two other components, i.e., the lateral and roll motion, unless visual yaw is sufficient in this slalom task (with low radius of curvature). In the previous study [21], the task was a corner negotiation (90° of rotation); the angle of curvature along with the total simulator velocity and angle were probably greater than in this slalom task. Thus, at higher velocity, drivers could probably more easily discern a configuration with and without yaw, which was not necessarily the case in the present study. Further work is required to elucidate this point, notably in a task requiring higher angular velocity, total angle, and lower linear speed.

In conclusion, the results of the present study clearly demonstrate that lateral motion gain should be adjusted as a function of the level of lateral acceleration to simulate. This seems to be mandatory for the settings of dynamic driving simulators and represents a new advance in the domain of simulation. However, tilt limit has to be considered. In addition, roll and notably yaw motions seem to have less influence on perception and driving performance. Surprisingly, it is therefore suggested that 1-to-1 gain could be the best setting for roll (as roll is not necessarily perceived by drivers with a lower gain). Although, the results from the subjective and objective variables with regard to yaw are not in agreement, yaw motion gain is never a factor significantly influencing the results in a positive way, at least in this slalom task. Thus, we cannot recommend using any specific yaw motion gain. Consequently, and in order to improve driver perception and control performance, the MCA should be changed by decreasing the lateral motion gain, while keeping the roll motion gain equal to 1. Yaw motion gain needs to be studied under different conditions.

REFERENCES

[1] A. Kemeny and F. Panerai, " Evaluating perception in driving simulation experiments. ", Trends Cogn Sci, vol. 7, n^o. 1, January. 2003, pp. 31–37.

[2] E. L. Groen and W. Bles, " How to use body tilt for the simulation of linear self motion ", J. Vestib. Res. Equilib. Orientat., vol. 14, n^o. 5, 2004, pp. 375‑385.

[3] A. Stratulat, V. Roussarie, J.-L. Vercher, and C. Bourdin, " Improving the realism in motion-based driving simulators by adapting tilt-translation technique to human perception ", 2011, pp. 47‑50.

[4] A. M. Stratulat, V. Roussarie, J.-L. Vercher, and C. Bourdin, " Perception of longitudinal acceleration on dynamic driving simulator ", presented at Proceeding of the Driving Simulation Conference 2012, Paris, France, 2012, pp. 33-40.

[5] D. R. Berger, J. Schulte-Pelkum, and H. H. Bülthoff, " Simulating believable forward accelerations on a stewart motion platform ", ACM Trans. Appl. Percept., vol. 7, n^o. 1, January. 2010, pp. 1‑27.

[6] E. Felipe and F. Navin, " Automobiles on Horizontal Curves: Experiments and Observations ", Transp. Res. Rec., vol. 1628, n^o. 1, January. 1998, pp. 50‑56.

[7] G. Reymond, A. Kemeny, J. Droulez, and A. Berthoz, " Role of lateral acceleration in curve driving: driver model and experiments on a real vehicle and a driving simulator ", Hum. Factors, vol. 43, n^o. 3, 2001, pp. 483‑495.

[8] A. Berthoz et al. " Motion Scaling for High-Performance Driving Simulators ", IEEE Trans. Hum.-Mach. Syst., vol. 43, n^o. 3, May. 2013, pp. 265‑276.

[9] Chapron, Thomas, and Colinot, Jean-Pierre, " The new PSA Peugeot-Citroën Advanced Driving Simulator Overall design and motion cue algorithm ", presented at Proceeding of the Driving Simulation Conference 2007, North America, Iowa City, 2007, pp. 44-52.

[10] B. J. Correia Grácio, J. E. Bos, M. M. Paassen, and M. Mulder, " Perceptual scaling of visual and inertial cues: Effects of field of view, image size, depth cues, and degree of freedom ", Exp. Brain Res., vol. 232, n^o. 2, November. 2013, pp. 637‑646.

[11] M. Dagdelen,, J.-C. Berlioux, F. Panerai, G. Reymond, and A. Kemeny, " Validation Process of the Ultimate high-performance driving simulator ", presented at Proceeding of the Driving Simulation Conference 2006, Paris, France, 2006, pp. 37-47.

[12] B. J. C. Grácio, M. Wentink, and A. R. Valente Pais, " Driver Behavior Comparison Between Static and Dynamic Simulation for Advanced Driving Maneuvers ", Presence Teleoperators Virtual Environ., vol. 20, n^o. 2, April. 2011, pp. 143‑161.

[13] W. Tinsson, " The notion of experimental plan ", in Experimental Plans: buildings and statistical analyses, vol. 67, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 3‑37.

[14] P. Feenstra, R. van der Horst, B. J. C. Grácio, and M. Wentink, " Effect of Simulator Motion Cuing on Steering Control Performance: Driving Simulator Study ", Transp. Res. Rec. J. Transp. Res. Board, vol. 2185, n^o. 1, December, 2010, pp. 48‑54.

[15] J. Östlund, B et al. " Driving Performance Assessment—Methods and Metrics. ", Report AIDE IST-1-507674-IP (D 2.2.5). European Union, 2005.

[16] A. R. Valente Pais, M. M. (René) Van Paassen, M. Mulder, and M. Wentick, " Perception Coherence Zones in Flight Simulation ", J. Aircr., vol. 47, n^o. 6, November. 2010, pp. 2039-2048,.

[17] A. Nesti, C. Masone, M. Barnett-Cowan, P. R. Giordano, H. H. Bülthoff, and P. Pretto, " Roll rate thresholds and perceived realism in driving simulation ", presented at Proceeding of the Driving Simulation Conference 2012, Paris, France, 2012, pp. 23-32.

[18] L. Bringoux, Sž. Schmerber, V. Nougier, G. Dumas, P. A. Barraud, and C. Raphel, " Perception of slow pitch and roll body tilts in bilateral labyrinthine-defective subjects. " Neuropsychologia, vol. 40, n^o. 4, 2002, pp. 367–372.

[19] A. J. Benson, E. C. Hutt, and S. F. Brown, " Thresholds for the perception of whole body angular movement about a vertical axis. ", Aviat Space Env. Med, vol. 60, n^o. 3, March. 1989, pp. 205–213.

[20] E. L. Groen, I. P. Howard, and B. S. Cheung, " Influence of body roll on visually induced sensations of self-tilt and rotation ", Perception, vol. 28, n^o. 3, 1999, pp. 287‑297.

[21] J. H. Hogema, M. Wentink, and G. P. Bertollini, « Effects of Yaw Motion on Driving Behaviour, Comfort and Realism », presented at Proceeding of the Driving Simulation Conference, Paris, France, 2012, pp. 149-158.

Different User Behavior’s Impact on Simulated Heating Demand in Energy Efficient Buildings

Hans Bagge
Lund University
Building Physics
Lund, Sweden
hans.bagge@byggtek.lth.se

Dennis Johansson
Lund University
Building Services
Lund, Sweden
dennis.johansson@hvac.lth.se

***Abstract*—To design a building that fulfills requirements regarding low energy use, it is crucial to perform energy simulations of the building in question during the design process and the simulations must be representative of the building during operation. All countries within the European Union will require new buildings to be nearly zero energy buildings beginning in 2019. In nearly zero energy buildings and passive houses, the user related energy uses, household electricity and domestic hot water heating, make up about 80 % of the total energy use since the use of space heating is low. The building's heating demand is affected by the occupants’ use of domestic hot water and household electricity. Increased use of domestic hot water increases the heating demand, while a high use of household electricity can reduce the heating demand. Different user characteristics will result in different heating demand in the same building, and in low energy buildings, different user characteristics will have a relatively higher impact compared to less energy efficient buildings. There is a lack of studies that analyze resulting energy use of dwellings based on distribution of measured user related input data. The aim of this paper was to annually measure household electricity and domestic hot water volume in 562 apartments, present the measured distributions and analyze the influence on the apartment heating demand of energy efficient buildings and typical buildings by use of simulations of the building physics and the building services. The results show that, in order to predict the energy use of energy efficient residential buildings, with a reasonable accuracy, the different users’ characteristics regarding household electricity and domestic hot water must be taken into account. Furthermore, to determine the impact that different users will have on a building's heating demand, the analysis must be based on the actual building and the result should be given as a distribution rather than as a single figure.**

Keywords - space heating; household electricity; domestic hot water; user behavior.

I. INTRODUCTION

To design a building that fulfills requirements regarding low energy use, it is crucial to perform energy simulations of the building in question during the design process and the simulations must be representative of the building during operation [1]. Research on the agreement between predicted and actual use of space heating in residential buildings in Sweden shows that measured use of energy for space heating during operation exceeds the predicted energy use by between 50% and 100%, even in low energy buildings [1][2][3][4].

Karlsson et al. [5] stressed the importance of accurate input data for the energy simulations of buildings. The building users’ behavior is very important in low energy buildings and is the hardest to model according to [5]. Low energy buildings have well insulated building enclosures and efficient ventilation heat recovery systems which lead to small transmission and ventilation heat losses. During a large part of the year, internal heat gains from people, household electricity and solar heat gains balance the heat losses with a zero heating need as a result.

The use of household electricity is influenced strongly by the building users’ behavior and is a major internal heat gain. Household electricity is defined as all electricity, not used for heating and ventilation, used in an apartment or a house. Domestic hot water is the hot water from taps in an apartment or a house. A large variation has been measured in equal apartments by [6]. Example of reasons for this variation can be occupancy levels and shower habits.

All energy uses in a building are part of the buildings energy balance and, for example the heat gains from a higher use of household electricity should result in a lower use of heating given that the heating control systems work as intended. Different user characteristics will result in different heating demand in the same building, and in low energy buildings, different user characteristics will have a relatively higher impact compared to less energy efficient buildings.

The users’ relatively larger impact on the building's performance in today's and tomorrow's buildings must be taken into account during design and management [7]. The users’ impact on the building performance has usually been described by different categories of users, for example families with children or single elders. Bagge [8] proposed to describe the different users’ lifestyle and impact by statistical distributions of user related parameters and combinations of parameters for the reason that it is unknown who will live in an apartment or a house over time. A vast majority of residential buildings are certainly not built for a specific category, but for a cross section of the population leading to an urgent need for a statistical approach on users’ variation. Energy use in buildings is commonly regulated. For example in Sweden, there are requirements on the sum of heating and non household electricity [9]. Heating is the sum of space heating for keeping the interior at desired temperature and domestic hot water heating. Non household electricity is used for ventilation and purposes outside apartments. A building's heating demand is affected by the occupants' use of domestic hot water and household electricity. Increased use of domestic hot water increases the heating demand, while a high use of household electricity can reduce the heating demand.

Various combinations of high and low uses of household electricity and hot water may be more or less favorable for achieving a low heating demand depending on building characteristic such as insulation standard and efficiency of ventilation heat recovery.

All countries within the European Union will require new buildings to be nearly zero energy buildings beginning in 2019. In order to make accurate predictions and decisions regarding future buildings, it is important to have a good statistical description on energy related user characteristics and its impact on buildings energy use and not only use average values of guessed or measured user related parameters.

There is a lack of studies that analyze resulting energy use of dwellings based on distribution of measured user related input data. The aim of this paper was to annually measure household electricity and domestic hot water volume, which by definition are totally user influenced, in 562 apartments, present the measured distributions and analyze the influence on the apartment heating demand of energy efficient buildings and typical buildings by use of simulations of the building physics and the building services taking into account the user of the building.

The paper is organized as follows. Section II presents the methods used and has four subsections A. Measurements, B. Simulated buildings and apartments, C. Parametric study and D. Simulation tool. Section III presents the measurement and simulation results and discusses these. Section IV presents the conclusions.

II. METHOD

The measured use of household electricity and domestic hot water in 562 apartments were used as input data in simulations of total heating demand of apartments in order to study the effect of different users. Parametric studies were performed to study the effect of different window area, ventilation heat exchanger efficiency, the average heat transmittance of the building and the location of the apartment within the building on the total heating demand taking into account the 562 different users.

As defined in the introduction, heating refers to the sum of space heating and domestic hot water heating. Space heating is the heating supplied by the heating system to the interior and the ventilation supply air, that means excluding domestic hot water heating. Household electricity is in practice completely used within the apartments, and is the only electricity used within the apartments [10]. There are also other internal heat gains such as solar radiation or occupant heat gain that is handled by the simulation program.

A. Measurements

As a basis for individual billing, household electricity and domestic hot water was measured during 2012 in 562 one bedroom apartments in buildings located in Karlstad, Sweden at latitude 59.39°, and built between 1932 and 2007 with a large portion built in 1980 and in the period 1961-1965. To obtain the domestic hot water heating, the measured domestic hot water volume was multiplied by 55 kWh/m^3 based on [11]. The distribution of the uses and their average values are presented as well as the actual relationship between the measured parameters in the studied apartments. Each measured apartment's use of household electricity and domestic hot water describes a user which means that in this study, 562 users are described.

B. Simulated buildings and apartments

The 562 combinations of use of household electricity and domestic hot water heating were used as input data in a simulation model of two different apartments in two different buildings. Building 1 had building technology representing a typical Swedish building designed during 2014 in accordance with the Swedish building code [9]. Building 2 had building technology representing a Swedish passive house [12].

In each building, the heating demands of one bedroom apartments, at two different locations in the building, were simulated. In both buildings, one of the apartments, Apartment 1, was located in the center of the building with adjacent apartments on two sides, above the ceiling and below the floor. That means that the apartment was going all way through the cross section of a floor of the building with half of the exterior surfaces facing north and the other half south. The other apartment, Apartment 2, was at the eastern gable and on the top floor meaning that exterior surfaces also included the eastern wall and the roof. The eastern wall did not include any windows. The outdoor climate data were obtained from Meteonorm [13] for Karlstad, the same city as measurements were from. Figure 1 presents the hourly outdoor temperature during the normal year. The normal year average outdoor temperature is 6.4°C.

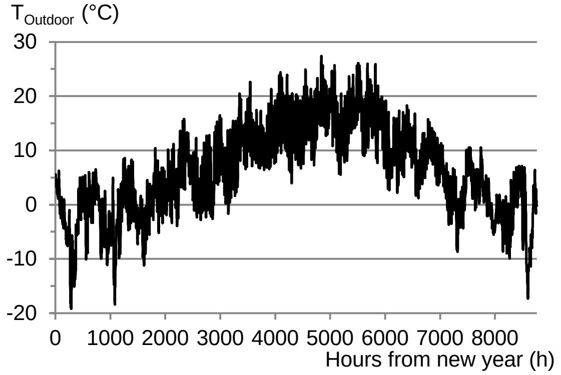


Figure 1. Hourly outdoor temperatures in Karlstad according to the data used in the simulations.

ISBN: 978-1-61208-371-1

Building 1 had exterior walls with a U-value of 0.18 W/(m²·K), roof with U-value of 0.13 W/(m²·K) and windows with a U-value of 1.3 W/(m²·K). Building 2 had exterior walls with a U-value of 0.1 W/(m²·K), roof with U-value of 0.08 W/(m²·K) and windows with a U-value of 0.8 W/(m²·K). Thermal bridging was estimated by adding 20 % to the described transmission losses for all building components. Mechanical supply and exhaust ventilation with heat recovery was used, 75 % temperature efficiency. The apartments had a heated floor area of 60 m² and a ceiling height of 2.4 m^2. The apartments north and south facades were 18 m^2 respectively and the eastern façade was 19.2 m^2 in the gable apartments. 60 % of the window area was facing south and 40 % of the window area was facing north. The ventilation airflow was 25 l/s and the leakage airflow was 0.04 $l/(s·m^2)$ and 0.013 $l/(s·m^2)$ referring to exterior surface area for Building 1 and Building 2 respectively. The apartments were heated to 22 °C and the occupancy was 0.03 persons/m² based on actual measurements of occupancy in the city of Karlstad [14]. The solar heat gain coefficient of the widows was set to 0.4.

Energy simulations in practice are commonly based on one-zone calculations. The same approach was chosen in this study to match sector practice.

C. Parametric study

In order to study different building technology characteristics impact on the heating demand with the different user scenarios, a parametric study was carried out. Window area was varied from zero to 45 % of heated floor area in steps of 5 %. Ventilation heat recovery temperature efficiency was varied from 50 % to 95 % in steps of 5 %. When the window area was varied, the ventilation heat recovery temperature efficiency was set to 75 %. When the ventilation heat recovery temperature efficiency was varied, the window area was set to 25 % of heated floor area. For each step, the heating demand was calculated with the 562 different user scenarios which results in the same number of different heating demands for each of the four different apartments studied. Statistics regarding the heating demand are presented for each step for each of the studied apartments respectively. The presented statistics are minimum, 10, 25, 50, 75 and 90 percentile and maximum as well as average values of total heating demand. That means that a total of 45120 simulations of heating demand were carried out to obtain the results of the parametric study.

D. Simulation tool

Code was developed in the Delphi programming language to simulate the energy use hourly over a normal year explicitly by help of the power balance shown in Figure 2 [15][16] to handle user scenarios and parametric studies effectively. ROOM is the simulated zone. P_{trans} is the transmitted heat through the envelope, P_{cap} is the heat from a first order heat capacitor with the temperature t_{cap} and a heat capacitance of 15000 J/(m²·K). P_{solar} is incoming shortwave solar radiation that heats the room and P_{vent} is the power needed to change the temperature of the supply air, t_{sa}, 19°C, to the temperature of the exhaust air, t_{ex}. It is assumed that the room temperature, t_{room}, is 22°C and can rise to 27°C and is the same as the exhaust temperature. P_{int} refers to the load from people and household electricity that both were assumed to be constant during the year based on the measurements.

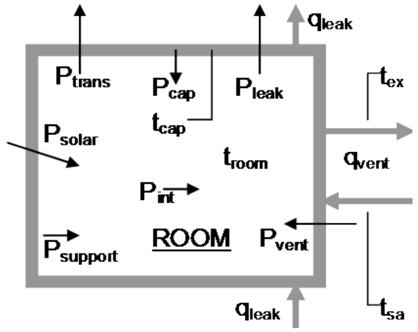


Figure 2. Power balance used in the simulation tool for the building. Quantities are given in the text.

$P_{support}$ is the energy needed to keep the room in balance at the desired t_{room}. Since no cooling system was used, $P_{support}$ could not be negative. Air heating after the heat recovery of the heating recovery ventilation is included but not shown in Figure 2. Also no air cooling was included. The SFP value of the air handling unit was set to 2 kW/(m³/s). Freezing protection of the heat recovery is modelled by keeping the exhaust air above freezing temperature.

III. Results and discussion

All measured calculated and simulated energy results are annual with the area referring to heated apartment floor area. The abbreviations HEL is used for household electricity and DHW for domestic hot water heating. As defined, Heating is the sum of Space heating and DHW.

Figure 3 shows the distribution of the annual use of HEL and DHW respectively. Average annual use of HEL was 26.7 kWh/m^2 and average annual DHW was 21.4 kWh/m^2. The median values are by definition the 50 percentile, directly readable in the figure. The highest use of HEL was 100 kWh/m^2 and the highest use of DHW was 104 kWh/m^2.

Figure 4 shows DHW as a function of HEL, a regression line and its coefficients are given. There is a rather weak correlation with a coefficient of determination of 0.14. In Figure 4, an apartment with low use of household electricity, about 11.5 kWh/m^2, had the highest use of domestic hot water, and an apartment with high use of household electricity, about 63 kWh/m^2, had a domestic hot water use close to zero which indicates the weak correlation. This implies that it is not straight forward to define a typical user of HEL and DHW. Hence, the actual distribution of combinations needs to be taken into account.

Figures 6 and 8 present statistics regarding the simulated total heating demand for different window areas in the apartments in Building 1 while Figures 7 and 9 present corresponding statistics for the apartments in Building 2. When the window area increases, the total transmission losses increases due to the higher transmission losses through a window compared to a wall. However, a larger window area can result in more solar heat gains.

Figures 10 and 12 present statistics regarding the simulated total heating demand for different ventilation heat recovery temperature efficiencies in the apartments in Building 1. Figures 11 and 13 present corresponding statistics for the apartments in Building 2. In Figures 6 through 13, maximums are given in the figure caption and percentile curve types are presented in Figure 5.

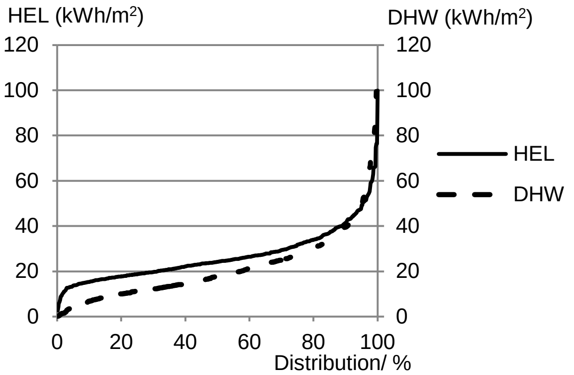


Figure 3. Duration of annual use of household electricity (HEL) and domestic hot water heating (DHW).

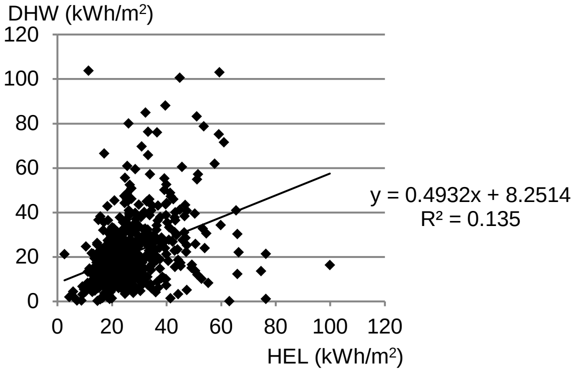


Figure 4. Use of annual domestic hot water heating (DHW) as a function of the use of annual household electricity (HEL).

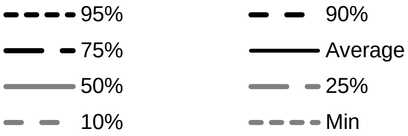


Figure 5. Curve types used in Figures 6 through 13.

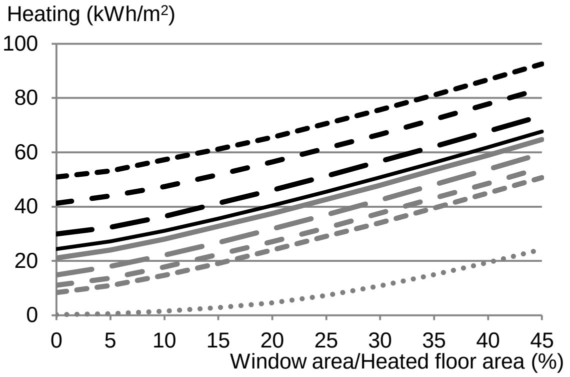


Figure 6. Annual heating, Apartment 1 in Building 1. Maximums: 111, 115, 119, 125, 130, 135, 141, 146, 152 and 158 kWh/m^2.

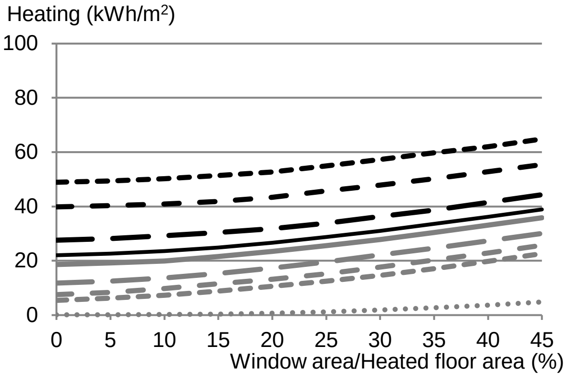


Figure 7. Annual heating, Apartment 1 in Building 2. Maximums: 105, 107, 108, 111, 113, 116, 118, 121, 124 and 127 kWh/m^2.

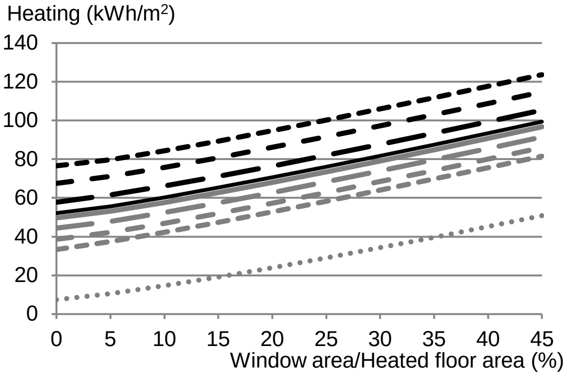


Figure 8. Annual heating, Apartment 2 in Building 1. Maximums: 146, 148, 153, 157, 163, 168, 174, 179, 185 and 191 kWh/m^2.

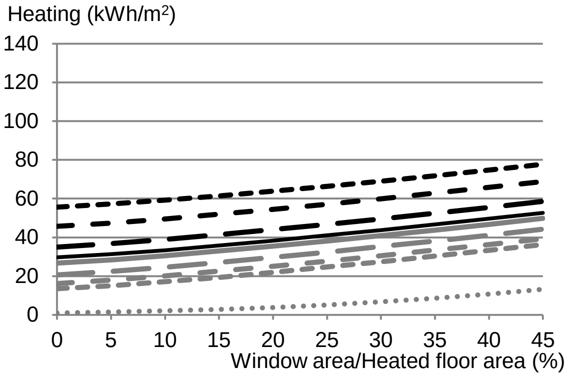


Figure 9. Annual heating, Apartment 2 in Building 2. Maximums: 119, 120, 122, 125, 127, 130, 133, 136, 139 and 142 kWh/m^2.

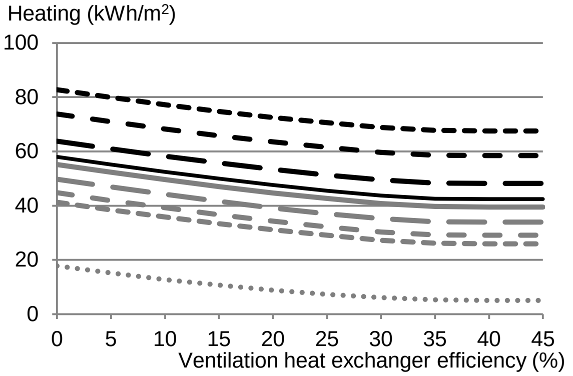


Figure 10. Annual heating, Apartment 1 in Building 1. Maximums: 148, 145, 143, 140, 137, 135, 133, 132, 132 and 132 kWh/m^2.

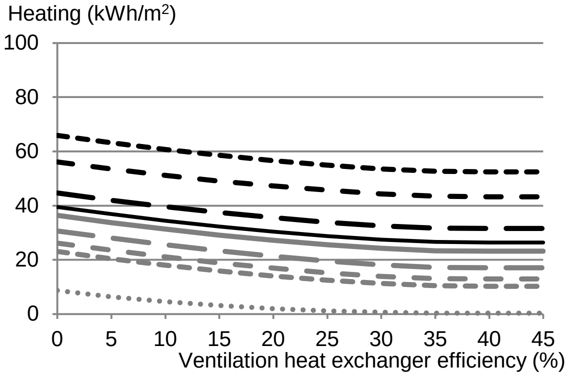


Figure 11. Annual heating, Apartment 1 in Building 2. Maximums: 127, 125, 122, 120, 118, 116, 114, 113, 113 and 113 kWh/m^2.

The results show that the heating demand increases with raised window area and decreases with raised ventilation heat exchanger temperature efficiency. In Building 1, the average heating demand increased 43 kWh/m^2 in Apartment 1 and 47 kWh/m^2 in Apartment 2 when 45 % window area was used compared to zero. In Building 2, the corresponding increase was 17 kWh/m^2 in Apartment 1 and 23 kWh/m^2 in Apartment 2.

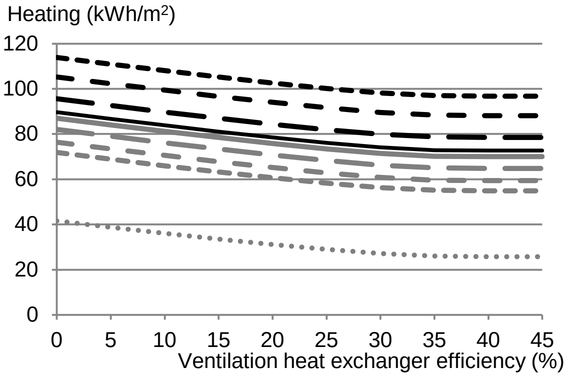


Figure 12. Annual heating, Apartment 2 in Building 1. Maximums: 181, 179, 176, 173, 170, 168, 166, 165, 165 and 165 kWh/m^2.

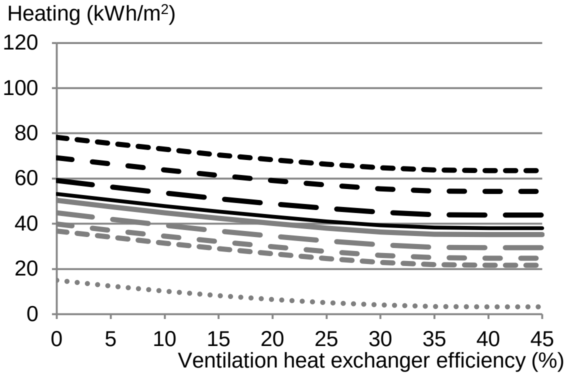


Figure 13. Annual heating, Apartment 2 in Building 2. Maximums: 143, 140, 137, 135, 132, 130, 128, 127, 127 and 127 kWh/m^2.

The difference between the 95 percentile and the 5 percentile of heating demand in Figures 6 through 13 represents the resulting span of heating demand taking into account 90 % of the users excluding the 5 % low and high extremes. The average difference between the 95 and 5 percentile of heating demand was about 42 kWh/m^2 for all studied window sizes in both buildings both apartments. In Building 1's Apartment 1, the span in heating demand, taking into account the middle 90 % of the users, is about the same as the average increase in heating demand when having 45 % window area compared to zero while the span is slightly lower than the heating demand increase in Apartment 2. In the more energy efficient passive house type Building 2, the span in heating demand was more than twice as high compared to the average increase in heating demand when having 45 % window area compared to zero in Apartment 1 and almost twice as high in Apartment 2. The above mentioned analysis compares the heating demand at a very large window area, 45 % of heated floor area, to the heating demand at no window area. It is not likely to have apartments with no windows due to daylight requirements. The difference in heating demand between all other studied window percentages will be smaller while the difference between the 95 and the 5 percentile of heating demand is about the same for all studied window sizes. The different user behaviors have a higher impact on the heating demand compared to different window areas and this increases with increasing energy efficiency regarding building enclosure and ventilation heat exchanger efficiency.

In Building 1, the average heating demand decreased 16 kWh/m^2 in Apartment 1 and 17 kWh/m^2 in Apartment 2 when 95 % temperature efficiency of the heat recovery was used compared to 55 %. In Building 2, the corresponding decrease was 13 kWh/m^2 in Apartment 1 and 15 kWh/m^2 in Apartment 2. The average difference between the 95 and 5 percentile of heating demand was about 42 kWh/m^2 for all studied efficiencies in both buildings both apartments. The span in heating demand was about three times higher than the average decrease in heating demand when having 95 % efficiency compared to 55 %. The difference in heating demand between all other studied temperature efficiencies will be smaller while the difference between the 95 and the 5 percentile of heating demand is about the same for all studied efficiencies. The different user behaviors have much higher impact on the heating demand compared to different ventilation heat exchanger temperature efficiencies.

As expected, the heating demand increases with raised window area and decreases with raised ventilation heat changer efficiency. However, different user behavior can have a much higher impact on the heating demand compared to different window sizes and heat exchanger efficiencies. Since today’s buildings are, and tomorrows buildings will be even more, energy efficient, the user related energy uses will be an even larger part of the buildings energy balance. As important as it is to accurately model the physical properties of the building enclosure and the building services in simulation tools, as important is it to take different users’ behavior into account.

The building technique and building services might have actual performance that differs from design values. For example, the thermal transmittance of the building enclosure is to some extent dependent on the quality of the construction work on site. However, according to the results, even relatively large variations in thermal transmittance, exemplified by the difference between Building 1 and Building 2, at 25 % window area and 75 % heat exchanger efficiency, is 17 kWh/m^2 which is less than half of the difference in heating demand taking into account the difference from the middle 90 % of the different users and slightly higher than the difference between the 75 and 25 percentile representing the middle 50 percent of the users.

The results imply that energy simulations of residential buildings should take the variation in user behavior into account and rather than presenting a single figure, present the predicted energy use in a span that represents the variation of the user behavior, for example based on the middle 90 % or 50 % of the users. Based on the performed simulations, the span representing 90 % of the different user behaviors is about 42 kWh/m^2 and the span representing 50 % is about 14 kWh/m^2.

IV. CONCLUSION

Annual use of household electricity and domestic hot water was measured in 562 one bedroom apartments in Sweden. Simulations of annual heating demand, taking into account the 562 different users, show that in order to predict the energy use of energy efficient residential buildings, with a reasonable accuracy, the different users’ characteristics regarding household electricity and domestic hot water must be taken into account. Furthermore, to determine the impact that different users will have on a building's heating demand, the analysis must be based on the actual building and the result should be given as a distribution rather than as a single figure.

ACKNOWLEDGMENT

This Research was founded by LÅGAN, SBUF and KBAB.

REFERENCES

[1] H. Bagge and D. Johansson, “Energy use in multi-family dwellings – demands and verification” Proceedings of 5th Nordic conference on construction economics and organization, vol. 1, 2009, pp 185-192.

[2] A. Elmroth, “Energy use in theory and practice” Contribution to the Anthology: More efficient energy use in residential buildings An anthology on future means of control (in Swedish), Swedish Energy Agency, Eskilstuna, Sweden, 2002, pp 66-75.

[3] A. Lindén, “Hammarby Sjöstad – crazy crossbar height” (in Swedish) VVS teknik & installation. The 2006 October issue, Stockholm, VVS forum, 2006, pp 2-5.

[4] A. Nilsson, “Energy use in newly built residential blocks at the Bo01 area in Malmö” (in Swedish) Lund, Building Physics LTH, Lund University, 2003.

[5] F. Karlsson, P. Rohdin, and ML. Persson, “Measured and predicted energy demand in a low energy building: important aspects when using Building Energy Simulations” Building Services Engineering Research and Technology, vol. 28, 2007, pp 223-235.

[6] H. Bagge, L. Lindstrii, and D. Johansson, “User related energy use – Result from mesurements in 1300 apartments” (in Swedish), Sveriges Byggindustrier, FoU-Väst rapport 1240, 2012.

[7] V. Corrado and HE. Mechri, “Uncertainty and sensitivity analysis for building energy rating” Journal of Building Physics, vol. 33, 2009, pp 125-155.

[8] H. Bagge, “Building Performance – Methods for Improved Prediction and Verification of Energy Use and Indoor Climate” Building Physics LTH, Lund University, Sweden. 2011.

[9] The Swedish National Board of Housing, Building and Planning. Building Regulations, BBR, BFS 2011: 26. 2014.

[10] H. Bagge, “Household electricity – measurements and analysis” Proceedings of Building physics symposium 2008 in Leuven, 2008, pp 95-99.

[11] Swedish Energy Agency , http://www.energimyndigheten.se/Hushall/Varmvatten-och-ventilation/Vatten-och-varmvattenberedare/, Accessed 2014-03-14.

[12] Sveriges centrum för nollenergihus, ”Kravspecifikation för nollenergihus, passivhus och minienergihus” http://www.nollhus.se/dokument/Kravspecifikation%20FEBY12%20-%20bostader%20sept.pdf. Accesssed 2014-05-24.

[13] Meteotest, “Meteonorm handbook, manual and theoretical background” Switzerland. 2011.

[14] D. Johansson, H. Bagge, and L. Lindstrii, “Measurements of occupancy levels in multi-family dwellings – Application to demand controlled ventilation” Journal of Energy and Buildings, vol. 43(9), 2011, pp 2449–2455.

[15] D. Johansson, “Modelling Life Cycle Cost for Indoor Climate Systems” Building Services LTH, Lund University, Sweden. 2005.

[16] International Organization for Standardization, “Energy performance of buildings – Calculation of energy use for space heating and cooling” EN ISO 13790. 2008.

Rare Event Handling in Signalling Cascades

Benoît Barbot, Serge Haddad and Claudine Picaronny
LSV, ENS Cachan & CNRS & Inria,
Cachan, France
{barbot,haddad,picaronny}@lsv.ens-cachan.fr

Monika Heiner
Brandenburg University of Technology,
Cottbus, Germany
monika.heiner@b-tu.de

***Abstract*—Signalling cascades are a recurrent pattern of biological regulatory systems whose analysis has deserved a lot of attention. It has been shown that Stochastic Petri Nets (SPN) are appropriate to model such systems and evaluate the probabilities of specific properties. Such an evaluation can be done numerically when the combinatorial state space explosion is manageable or statistically otherwise. However, when the probabilities to be evaluated are too small, random simulation requires more sophisticated techniques for the handling of rare events. In this paper, we show how such involved methods can be successfully applied for signalling cascades. More precisely, we study three relevant properties of a signalling cascade with the help of the COSMOS tool. Our experiments point out interesting dependencies between quantitative parameters of the regulatory system and its transient behaviour. In addition, they demonstrate that we can go beyond the capabilities of MARCIE, which provides one of the most efficient numerical solvers.**

Keywords–Rare event; Importance sampling; Signalling cascade.

I. INTRODUCTION

Signalling cascades: Signalling processes play a crucial role for the regulatory behaviour of living cells. They mediate input signals, i.e., the extracellular stimuli received at the cell membrane, to the cell nucleus, where they enter as output signals the gene regulatory system. Understanding signalling processes is still a challenge in cell biology. To approach this research area, biologists design and explore signalling networks, which are likely to be building blocks of the signalling networks of living cells. Among them are the type of signalling cascades which we investigate here.

A signalling cascade is a set of reactions which can be grouped into levels. At each level a particular enzyme is produced (e.g., by phosphorylation); the level generally also includes the inverse reactions (e.g., dephosphorylation). The system constitutes a cascade since the enzyme produced at some level is the catalyser for the reactions at the next level. The catalyser of the first level is usually considered to be the input signal, while the catalyser produced by the last level constitutes the output signal. The transient behaviour of such a system presents a characteristic shape, the quantity of every enzyme increases to some stationary value. In addition, the increases are temporally ordered w.r.t. the levels in the signalling cascade. This behaviour can be viewed as a signal travelling along the levels, and there are many interesting properties to be studied like the travelling time of the signal, the relation between the variation of the enzymes of two consecutive levels, etc.

In [1], it has been shown how such a system can be modelled by a Petri net which can either be equipped with continuous transition firing rates leading to a continuous Petri net which determines a set of differential equations or by stochastic transition firing rates leading to a SPN. This approach emphasises the importance of Petri nets which, depending on the chosen semantics, permit to investigate particular properties of the system. In this paper, we wish to explore the influence of stochastic features on the signalling behaviour, and thus we focus on the use of SPN.

Analysis of SPNs can be performed either numerically or statistically. The former approach is much faster than the latter and provides exact results up to numerical approximations, but its application is limited by the memory requirements due to the combinatory explosion of the state space.

Statistical evaluation of rare events: Statistical analysis means to estimate the results by evaluating a sufficient number of simulations. However, standard simulation is unable to efficiently handle *rare events*, i.e., properties whose probability of satisfaction is tiny. Indeed, the number of trajectories to be generated in order to get an accurate interval confidence for rare events becomes prohibitively huge. Thus, *acceleration* techniques [2] have been designed to tackle this problem whose principles consist in (1) favouring trajectories that satisfy the property, and (2) numerically adjusting the result to take into account the bias that has been introduced. This can be done by *splitting* the most promising trajectories [3] or *importance sampling* [4], i.e., modifying the distribution during the simulation. In a previous work [5], some of us have developed an original importance sampling method based on the design and numerical analysis of a reduced model in order to get the importance coefficients. This method was first proposed for checking “unbounded until” properties over models whose semantics is a discrete time Markov chain, it has been extended to also handle “bounded until” properties and continuous time Markov chains [6].

Our contribution: In this paper, we consider three families of properties for signalling cascades that are particularly relevant for the study of their behaviour and that are (depending on a scaling parameter) potentially rare events. From an algorithmic point of view, this case study raises interesting issues since the combinatorial explosion of the model quickly forbids the use of numerical solvers and its intricate (quantitative) behaviour requires elaborated and different abstractions depending on the property to be checked.

Due to these technical difficulties, the signalling cascade analysis has led us to substantially improve our method and in particular the way we obtain the final confidence interval. From a biological point of view, experiments have pointed out interesting dependencies between the scaling parameter of the model and the probability of satisfying a property.

Organisation: In Section II, we present the biological background, the signalling cascade under study and the properties to be studied. Then, in Section III after some recalls on SPN, we model signalling cascades by SPNs. We introduce the rare event issue and the importance sampling technique to cope with in Section IV. In Section V, we develop our method for handling rare events. Then, in Section VI, we report and discuss the results of our experiments. Finally in Section VII, we conclude and give some perspectives to our work. Additional explanations related to model abstractions, algorithmic considerations, and experimental statistical analysis can be found in the following research report [7].

II. Signalling cascades

In technical terms, signalling cascades can be understood as networks of biochemical reactions transforming input signals into output signals. In this way, signalling processes determine crucial decisions a cell has to make during its development, such as cell division, differentiation, or death. Malfunction of these networks may potentially lead to devastating consequences on the organism, such as outbreak of diseases or immunological abnormalities. Therefore, cell biology tries to increase our understanding of how signalling cascades are structured and how they operate. However, signalling networks are generally hard to observe and often highly interconnected, and thus signalling processes are not easy to follow. For this reason, typical building blocks are designed instead, which are able to reproduce observed input/output behaviours.

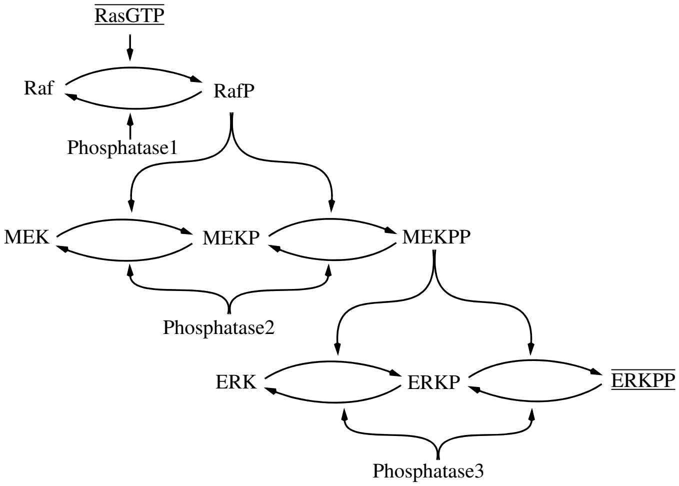


Figure 1. The general scheme of the considered three-level signalling cascade; RasGTP serves as input signal and ERKPP as output signal.

The case study we have chosen for our paper is such a signalling building block: the mitogen-activated protein kinase (MAPK) cascade [8]. This is the core of the ubiquitous ERK/MAPK network that can, among others, convey cell division and differentiation signals from the cell membrane to the nucleus. The description starts at the RasGTP complex which acts as an enzyme (kinase) to phosphorylate Raf, which phosphorylates MAPK/ERK Kinase (MEK), which in turn phosphorylates Extracellular signal Regulated Kinase (ERK). We consider RasGTP as the input signal and ERKPP (activated ERK) as the output signal. This cascade (RasGTP → Raf → MEK → ERK) of protein interactions is known to control cell differentiation, while the strength of the effect depends on the ERK activity, i.e., concentration of ERKPP.

The scheme in Figure 1 describes the typical modular structure for such a signalling cascade [9]. Each layer corresponds to a distinct protein species. The protein Raf in the first layer is only singly phosphorylated. The proteins in the two other layers, MEK and ERK, respectively, can be singly as well as doubly phosphorylated. In each layer, forward reactions are catalysed by kinases and reverse reactions by phosphatases (Phosphatase1, Phosphatase2, Phosphatase3). The kinases in the MEK and ERK layers are the phosphorylated forms of the proteins in the previous layer. Each phosphorylation/dephosphorylation step applies mass action kinetics according to the pattern $A + E \rightleftharpoons AE \rightarrow B + E$. This pattern reflects the mechanism by which enzymes act: first building a complex with the substrate, which modifies the substrate to allow for forming the product, and then disassociating the complex to release the product; for details see [10].

Having the wiring diagram of the signalling cascade, a couple of interesting questions arise whose answers would shed some additional light on the subject under investigation. Among them are an assessment of the signal strength in each level, and specifically of the output signal. We will consider these properties in Sections VI-A and VI-B. The general scheme of the signalling cascade also suggests a temporal order of the signal propagation in accordance with the level order. What cannot be derived from the structure is the extent to which the signals are simultaneously produced; we discuss this property in the technical report [7].

III. Petri net modelling

Stochastic Petri nets: Due to their graphical representation and bipartite nature, Petri nets are highly appropriate to model biochemical networks. When equipped with a stochastic semantics, yielding SPN [11], they can be used to perform quantitative analysis.

Definition 1 (SPN)**.** *A SPN* $\mathcal{N}$ *is defined by a tuple* $(P, T, \mathbf{Pre}, \mathbf{Post}, \{\mu_t\}_{t \in T})$ *where* P *is a finite set of* places*,* T *is a finite set of* transitions*,* $\mathbf{Pre}$*,* $\mathbf{Post}$ *are matrices from* $P \times T$ *to* $\mathbb{N}$*, and* $\{\mu_t\}_{t \in T}$ *is a set of mappings from* $\mathbb{N}^P$ *to* $\mathbb{R}_{>0}$.

A *marking* m of SPN $\mathcal{N}$ is an item of $\mathbb{N}^P$. A transition $t \in T$ is fireable in marking m if for all *places* $p \in P\ m(p) \geq \mathbf{Pre}(p,t)$. Its firing leads to marking m' defined by: for all $p \in P\ m'(p) = m(p) - \mathbf{Pre}(p,t) + \mathbf{Post}(p,t)$. This firing is denoted either $m \xrightarrow{\sigma} m'$ or as $m \xrightarrow{t}$ omitting the next marking. We extend these notations for any $\sigma = \sigma_1 \ldots \sigma_n \in T^*$ of successive fireable transitions, if σ is fireable from m, that is if there exists a sequence of markings $m = m_0, m_1, \ldots, m_n$ such that for all $0 \leq k < n$, $m_k \xrightarrow{\sigma_k} m_{k+1}$. Let m_0 be an initial marking, the reachability set $Reach(\mathcal{N}, m_0)$ is defined by: $Reach(\mathcal{N}, m_0) = \{m \mid \exists \sigma \in T^*\ m_0 \xrightarrow{\sigma} m\}$. The initialised SPNs $(\mathcal{N}, m_0)$ that we consider do not have deadlocks: for all $m \in Reach(\mathcal{N}, m_0)$ there exists $t \in T$ such that $m \xrightarrow{t}$. each transition t is equipped with a mapping μ_t. In a marking m, each enabled transition of the Petri net randomly selects an execution time according to a Poisson process with rate $\mu_t(m)$. Then, the transition with earliest firing time is selected

TABLE I. DEVELOPMENT OF THE STATE SPACE FOR INCREASING N.

N	number of states	N	number of states
1	24,065 (4)	6	769,371,342,640 (11)
2	6,110,643 (6)	7	5,084,605,436,988 (12)
3	315,647,600 (8)	8	27,124,071,792,125 (13)
4	6,920,337,880 (9)	9	122,063,174,018,865 (14)
5	88,125,763,956 (10)	10	478,293,389,221,095 (14)

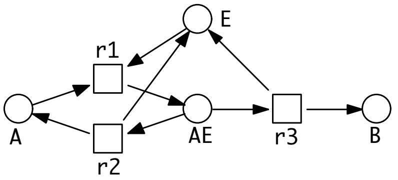


Figure 2. Petri net pattern for mass action kinetics $A + E \rightleftharpoons AE \rightarrow B + E$.

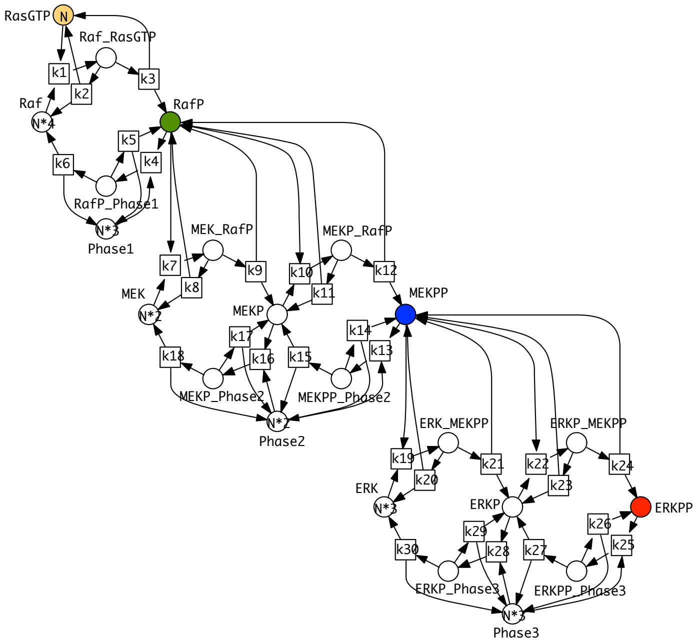


Figure 3. A Petri net modelling the three-level signalling cascade given in Figure 1; k_i are the kinetic constants for mass action kinetics, N the scaling parameter.

to fire yielding the new marking. Operational semantics of a SPN is a Continuous Time Markov Chain (CTMC). This can be formalized as follows.

Definition 2 (CTMC of a SPN)**.** *Let $\mathcal{N}$ be a SPN and m_0 be an initial marking. Then the CTMC associated with $(\mathcal{N}, m_0)$ is defined by its set of states which is $Reach(\mathcal{N}, m_0)$, its transition matrix* **P** *defined by Equation 1*

$$\mathbf{P}(m, m') = \frac{\sum_{m \xrightarrow{t} m'} \mu_t(m)}{\sum_{m \xrightarrow{t}} \mu_t(m)} \qquad (1)$$

and, for each state m, the rate λ_m defined by: $\lambda_m = \sum_{m \xrightarrow{t}} \mu_t(m)$

Running case study: We now explain how to model our running case study in the Petri net framework. The signalling cascade is made of several phosphorylation/dephosphorylation steps, which are built on mass/action kinetics. Each step follows the pattern $A + E \rightleftharpoons AE \rightarrow B + E$ and is modelled by a small Petri net component depicted in Figure 2. The mass action kinetics is expressed by the rate of the transitions. The marking-dependent rate of each transition is equal to the product of the number of tokens in all its incoming places up to a multiplicative constant given by the biological behaviour (summing up dependencies on temperature, pressure, volume, etc.).

The whole reaction network based on the general scheme of a three-level double phosphorylation cascade, as given in Figure 1, is modelled by the Petri net in Figure 3. The input signal is the number of tokens in the place RasGTP, and the output signal is the number of tokens in the place ERKPP.

This signalling cascade model represents a self-contained and closed system. It is covered with place invariants (see the research report for details), specifically each layer in the cascade forms a place invariant consisting of all states a protein can undergo; thus the model is bounded. Assuming an appropriate initial marking, the model is also live and reversible; see [1] for more details, where this Petri net has been developed and analysed in the qualitative, stochastic and continuous modelling paradigms. In our paper we extend these analysis techniques for handling properties corresponding to rare events.

We introduce a scaling factor N to parameterize how many tokens are spent to specify the initial marking. Increasing the scaling parameter can be interpreted in two different ways: either an increase of the biomass circulating in the closed system (if the biomass value of one token is kept constant), or an increase of the resolution (if the biomass value of one token inversely decreases, called level concept in [1]). The kind of interpretation does not influence the approach we pursue in this paper.

Increasing N increases the size of the state space and thus of the CTMC, as shown in Table I, which has been computed with the symbolic analysis tool MARCIE [12]. As expected, the explosion of the state space prevents numerical model checking for higher N and thus calls for statistical model checking.

Furthermore, increasing the number of states actually decrease the probabilities to be in a certain state, as the total probability of 1 is fixed. With the distribution of the probability mass of 1 over an increasingly huge number of states, we obtain sooner or later states with very tiny probabilities, and thus rare events. Neglecting rare events is usually appropriate when focusing on the averaged behaviour. But they become crucial when certain jump processes such as mutations under rarely occurring conditions are of interest.

IV. STATISTICAL MODEL CHECKING WITH RARE EVENTS

A. Statistical model checking and rare events

Simulation recalls: The statistical approach for evaluating the expectation $\mathbf{E}(X)$ of a random variable X related to a random path in a Markov chain is generally based on three parameters: the number of simulations K, the confidence level γ, and the width of the confidence interval lg (see [13]). Once the user provides two parameters, the procedure computes the remaining one. Then it performs K simulations of the Markov chain and outputs a confidence interval $[L, U]$ with a width of at most lg such that $\mathbf{E}(X)$ belongs to this interval with a probability of at least γ.

Statistical evaluation of a reachability probability: Let $\mathcal{C}$ be a Discrete Time Markov Chain (DTMC) with two absorbing states s_+ or s_-, such that the probability to reach s_+ or s_- from any state is equal to 1. Assume one wants to estimate p, the probability to reach s_+. Then the simulation step consists in generating K paths of $\mathcal{C}$ which end in an absorbing state. Let K_+ be the number of paths ending in state s_+. The random variable K_+ follows a binomial distribution with parameters p and K. Unfortunately, when $p \ll 1$, the number of paths required for a small confidence interval is too large to be simulated. This issue is known as the *rare event* problem.

Importance sampling: In order to tackle the rare event problem, the importance sampling method relies on a choice of a biased distribution that will artificially increase the frequency of the observed rare event during the simulation. The generation of paths is done according to a modified DTMC $\mathcal{C}'$, with the same state space, but modified transition matrix $\mathbf{P}'$. $\mathbf{P}'$ must satisfy Property 2:

$$\mathbf{P}(s,s') > 0 \Rightarrow \mathbf{P}'(s,s') > 0 \vee s' = s_- \qquad (2)$$

which means that this modification cannot remove transitions that have not s_- as target, but can add new transitions. The method maintains a correction factor called L initialised to 1; this factor represents the *likelihood* of the path. When a path crosses a transition $s \to s'$ with $s' \neq s_-$, L is updated by $L \leftarrow L\frac{\mathbf{P}(s,s')}{\mathbf{P}'(s,s')}$. When a path reaches s_-, L is set to zero. If $\mathbf{P}' = \mathbf{P}$ (i.e., no modification of the chain), the value of L when the path reaches s_+ (resp. s_-) is 1 (resp. 0). Let V_s (resp. W_s) be the random variable associated with the final value of L for a path starting in x in the original model $\mathcal{C}$ (resp. in $\mathcal{C}'$). By definition, the expectation $\mathbf{E}(V_{s_0}) = p$ and by construction of the likelihood, $\mathbf{E}(W_{s_0}) = p$. Of course, a useful importance sampling should reduce the variance of W_{s_0} w.r.t. to the one of V_{s_0} equal to $p(1-p) \approx p$ for a rare event.

V. Our methodology for importance sampling

A. Previous work

In [5][6], we provided a method to compute a biased distribution for importance sampling: we manually design an abstract smaller model, with a behaviour close to that of the original model, that we call the *reduced model*. This is done by lumping together some states with the objective of making the rare event less rare. We perform numerical computations on this smaller model to obtain the biased distribution. We applied this method in order to tackle the estimation of time bounded property in CTMCs, that is the probability to satisfy a formula $aU^{[0,\tau]}b$, when it is a rare event. Let us outline the different steps of the method which is depicted in Figure 4.

Abstraction of the model: As discussed above, given a SPN $\mathcal{N}$ modelling the system to be studied, we manually design an appropriate reduced one $\mathcal{N}^\bullet$ and a correspondence function f from states of $\mathcal{N}$ to states of $\mathcal{N}^\bullet$. Function f is defined at the net level (see Section VI).

Structural analysis: Importance sampling was originally proposed for DTMCs. In order to apply it for CTMC $\mathcal{C}$ associated with net $\mathcal{N}$, we need to uniformize $\mathcal{C}$ (and also $\mathcal{C}^\bullet$ associated with $\mathcal{N}^\bullet$) which means finding a bound Λ for exit rate of states, i.e., markings, considering Λ as the uniform exit rate of states and rescaling accordingly the transition probability matrices [14]. Since the rates of transitions depend on the current marking, determining Λ requires a structural analysis like invariant computations for bounding the number of tokens in places.

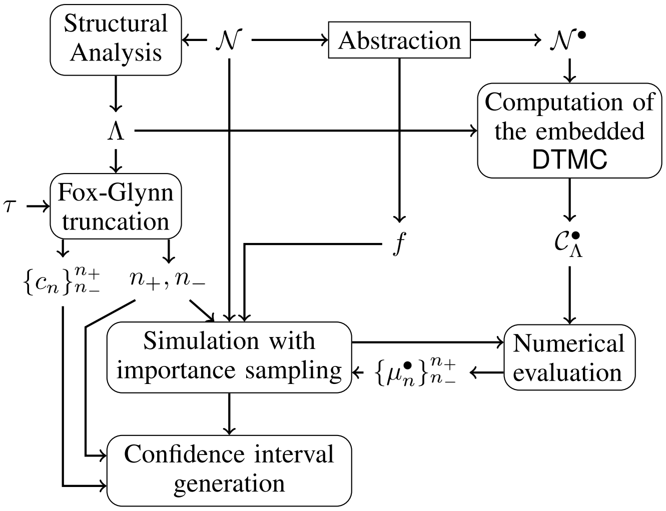

Structural Analysis
$\mathcal{N}$
Abstraction
$\mathcal{N}^\bullet$
Computation of the embedded DTMC
Λ
τ
Fox-Glynn truncation
f
$\mathcal{C}^\bullet_\Lambda$
$\{c_n\}_{n_-}^{n_+}$
n_+, n_-
Simulation with importance sampling
$\{\mu^\bullet_n\}_{n_-}^{n_+}$
Numerical evaluation
Confidence interval generation

Figure 4. Principles of the methodology for estimating probability of occurrence of rare events.

Fox-Glynn truncation: Given a uniform chain with initial state s_0, exit rate Λ, and transition probability matrix $\mathbf{P}$, the state distribution π_τ at time τ is obtained by the following formula: $\pi_\tau(s) = \sum_{n\geq 0} \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!}\mathbf{P}^n(s_0, s)$. This value can be estimated, with sufficient precision, by applying [15]. Given two numerical accuracy requirements α and β, truncation points n^- and n^+ and values $\{c_n\}_{n^-\leq n\leq n^+}$ are determined such that for $n^- \leq n \leq n^+$: $c_n(1-\alpha-\beta) \leq \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \leq c_n$, $\sum_{n<n^-} \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \leq \alpha$ and $\sum_{n>n^+} \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \leq \beta$.

Computation of the embedded DTMC: We build the embedded DTMC $\mathcal{C}^\bullet_\Lambda$ of $\mathcal{N}^\bullet$ after uniformization. Since we want to evaluate the probability to satisfy formula $aU^{[0,\tau]}b$, the states satisfying a (resp. $\neg a \wedge \neg b$) are aggregated into an absorbing accepting (resp. rejecting) state. Let $\mu_n(s^\bullet)$ be the probability to be in the accepting state at time τ starting from state $s^\bullet$.

Numerical evaluation: Matrix $\mathbf{P}'$ used for importance sampling simulation in the embedded DTMC of $\mathcal{N}$ to evaluate formulas $aU^{[0,n]}b$ for $n^- \leq n \leq n^+$, is based on the distributions $\{\mu^\bullet_n\}_{0<n\leq n^+}$, where $\mu^\bullet_n(s^\bullet)$ is the probability that a random path of the embedded DTMC of $\mathcal{N}^\bullet$ starting from $s^\bullet$ fulfills $aU^{[0,n]}b$. Such a distribution is computed by a standard numerical evaluation. However since n^+ can be large, depending on the memory requirements, this computation can be done statically for all n or dynamically for a subset of such n during the importance sampling simulation (more details are given in the research report).

Simulation with importance sampling: Here the random distribution of the successors of a state depend on both the embedded DTMC $\mathcal{C}_\Lambda$ and the values computed by the numerical evaluation. Moreover, all formulas $aU^{[0,n]}b$ for $n^- \leq n \leq n^+$ have to be evaluated increasing the time complexity of the method.

Generation of the confidence interval: After the simulations we get a family of confidence intervals indexed by $n^- \leq n \leq n^+$. Using Fox-Glynn truncation, we weight and combine the confidence intervals and return the final interval.

B. Tackling signalling cascades

Tackling signalling cascades leads us to two improvements: (1) we have performed a much more efficient importance sampling simulation and (2) we have proposed different ways of computing confidence intervals. We now detail these issues.

Importance sampling for multiple formulas: A naive implementation would require to apply statistical model checking of formulas $aU^{[0,n]}b$ for all n between n^- and n^+, but such a number can be large. A more tricky alternative consists in producing all trajectories until time horizon n^+ and updating the simulation results at the end of a trajectory for all the intervals $[0, n]$ with $n^- \leq n \leq n^+$ as follows. If the trajectory has reached the absorbing rejecting state s^- then it is an unsuccessful trajectory for all intervals. Otherwise if it has reached the absorbing accepting state s^+ at time n_0 then for all $n \geq n_0$ it is a successful trajectory and for all $n < n_0$ it is unsuccessful. Doing this way, every trajectory contributes to all evaluations, and we significantly increase the sample size without increasing computational cost. The accuracy of the results is improved.

Confidence interval estimation: The result of each trajectory of the simulation is a realisation of the random variable $W_{s_0} = X_{s_0} L_{s_0}$ where the binary variable X_{s_0} indicates whether a trajectory starting from s_0 is succesful and the positive random variable L_{s_0} is the (random) likelihood. Observe that $\mathbf{E}(W_{s_0}) = \mathbf{E}(L_{s_0}|X_{s_0} = 1)\mathbf{E}(X_{s_0})$. Since X_{s_0} follows a Bernoulli distribution, a confidence interval can be easily computed for $\mathbf{E}(X_{s_0})$. For $\mathbf{E}(L_{s_0}|X_{s_0} = 1)$ several approaches are possible among them we have selected three possible computations ranked by conservation degree. The first method assumes that the distribution is Gaussian (which is asymptotically valid if the variance is finite, thanks to the central limit theorem). Another method uses a pseudo Chernoff-Hoeffding bound. Whenever the random variable is bounded, this method is asymptotically valid. In our case we use the minimal and maximal values observed during the simulation as the bounds of L_{s_0}. The last method consists in returning the minimal and maximal observed values as the confidence interval.

VI. Experiments

We have analysed two properties, the one of them is inspired by [1]. Recall that the initial marking of the model is parametrized by a scaling factor N. For the first propertie, the reduced model is the same model but with local smaller scaling factors on the different layers of phosphorylation. Every state of the initial model is mapped (by f) to a state of the abstract model which has the "closest" proportion of chemical species. For instance let $N = 4$, which corresponds to 16 species of the first layer, a state with 6 tokens in Raf and 10 tokens in RafP is mapped, for a reduced model with $N = 3$, to a state with $4 = \lfloor 6 \times 3/4 \rfloor$ tokens in Raf and $8 = \lceil 10 \times 3/4 \rceil$ tokens in RafP (see the research report for a specification of f).

All statistical experiments have been carried out with our tool Cosmos [16]. Cosmos is a statistical model checker for the HASL logic [16]. It takes as input a Petri net (or a high-level Petri net) with general distributions for transitions. It performs an efficient statistical evaluation of the SPN by generating a code per model and formula. In the case of importance sampling, it additionally takes as inputs the reduced model and the mapping function specified by a C function and returns the different confidence intervals. All experiments have been performed on a machine with 16 cores running at 2 GHz and 32 GB of memory both for the statistical evaluation of Cosmos and the numerical evaluation of Marcie.

We perform additional experiments with a third property, which can be found in the research report.

A. Maximal peak of the output signal

The first property is expressed as a time-bounded reachability formula assessing the strength of the output signal of the last layer: "What is the probability to reach within 10 time units a state where the total mass of ERK is doubly phosphorylated?", associated with probability p_1 defined by:

$$p_1 = \Pr(\text{True } \mathbf{U}^{\leq 10}(\text{ERKPP} = 3N))$$

The inner formula is parametrized by N, the scaling factor of the net (via its initial marking). The reduced model that we design for Cosmos uses different scaling factors for the three layers in the signalling cascade. The first two layers of phosphorylation which are based on Raf and MEK always use a scaling factor of 1, whereas the last layer involving ERK uses a scaling factor of N. The second column of Table II shows the ratio between the number of reachable states of the original and the reduced models.

We have performed experiments with both Cosmos and Marcie. The time and memory consumptions for increasing values of N are reported in Table II. For each value of N we generate one million trajectories with Cosmos. We observe that the time consumption significantly increases between $N = 3$ and $N = 4$. This is due to a change of strategy in the space/time trade-off in order to not exceed the machine memory capacity. Marcie suffers an exponential increase w.r.t. both time and space resources. When $N = 3$, it is slower than Cosmos and it is unable to handle the case $N = 4$.

Table II depicts the values returned by the two tools: Marcie returns a single value, whereas Cosmos returns three confidence intervals (discussed above) with a confidence level set to 0.99. We observe that confidence intervals computed by the Gaussian analysis neither contain the result, the ones computed by Chernoff-Hoeffding do not contain it for $N = 3$, and the most conservative ones always contain it (when this result is available). An analysis of the likelihood L_{s_0} is detailed in the research report. It appears that the probability p_1 depends on N in an exponential way: $p_1 \approx 800(3 \cdot 10^{-15})^N$. The constants occurring in the formula could be interpreted by biologists.

B. Conditional maximal signal peak

The network structure of each layer in the signalling cascade presents a cyclic behaviour, i.e., phosphorylated proteins, serving as signal for the next layer, can also be dephosphorylated again, which corresponds to a decrease of the signal strength. Thus an interesting property of the signalling cascade is the probability of a further increase of the signal strength under the condition that a certain strength has already been reached. We estimate this quantity for the first layer in the signalling cascade, i.e., RafP, and ask specifically for the

TABLE II. COMPUTATIONAL COMPLEXITY AND NUMERICAL VALUES RELATED TO THE EVALUATION OF p_1

N	COSMOS						MARCIE		
	Reduction factor	time	memory	Gaussian CI	Chernoff CI	MinMax CI	time	memory	Output
1	-	-	-	-	-	-	4	514MB	2.07E-12
2	38	20,072	3,811MB	[3.75E-27,5.88E-26]	[3.75E-27,4.54E-25]	[3.75E-27,1.57E-23]	326	801MB	8.18E-26
3	558	15,745	15,408MB	[4.34E-42,1.72E-39]	[4.34E-42,1.82E-38]	[4.43E-42,1.87E-37]	43,440	13,776MB	2.56E-39
4	4667	40,241	3,593MB	[1.54E-57,8.54E-56]	[1.54E-57,1.98E-55]	[1.78E-57,7.05E-55]	Out of Memory: >32GB		
5	27353	51,120	19,984MB	[3.97E-73,2.33E-70]	[3.97E-73,7.30E-70]	[5.44E-73,2.24E-69]			

TABLE III. NUMERICAL VALUES ASSOCIATED WITH p_2

N	L	COSMOS		MARCIE		
		confidence interval	time	result	time	memory
2	2	[2.39E-13 , 1.07E-09]	31	5.55E-10	90	802 MB
2	3	[2.18E-10 , 6.92E-08]	110	6.64E-08	136	816 MB
2	4	[9.33E-08 , 3.54E-05]	256	3.01E-06	276	798 MB
2	5	[1.16E-05 , 6.08E-04]	1000	7.16E-05	759	801 MB
2	6	[5.42E-04 , 1.21E-03]	5612	1.27E-03	3180	804 MB
3	5	[1.82E-12 , 9.78E-09]	459	Time > 48 hours		
3	6	[3.41E-10 , 9.66E-08]	1428			
3	7	[1.81E-08 , 2.23E-06]	7067			
3	8	[8.72E-07 , 2.71E-06]	4460			
3	9	[1.42E-06 , 4.59E-05]	4301			
3	10	[2.69E-04 , 9.34E-04]	6420			
4	10	[5.12E-09 , 2.75E-08]	8423	Memory > 32GB		
4	11	[8.23E-08 , 2.97E-07]	7157			
4	12	[9.84E-07 , 1.86E-06]	18730			

probability to reach its maximal strength, $4N$: "What is the probability of the concentration of RafP to continue its increase and reach $4N$, when starting in a state where the concentration is for the first time at least L?". This is a special use case of the general pattern introduced in [1].

$$p_2 = \Pr_\pi((\text{RafP} \geq L)\ \mathbf{U}\ (\text{RafP} \geq 4N))$$

where π is the distribution over states when satisfying for the first time the state formula RafP $\geq L$ (previously called a filter). This formula is parametrized by threshold L and scaling factor N. The results for increasing N and L are reported in Table III (confidence intervals are computed by Chernoff-Hoeffding method). As before, MARCIE cannot handle the case $N = 3$, the bottleneck being here the execution time.

It is clear that p_2 is an increasing function of L. More precisely, experiments point out that p_2 increases approximatively exponentially by at least one magnitude order when L is incremented. However this dependency is less clear than the one of the first property.

VII. CONCLUSION AND FUTURE WORK

We have studied rare events in signalling cascades with the help of an improved importance sampling method implemented in COSMOS. Our method has been able to cope with huge models (nearly a hundred billion states) that could not be handled neither by computations nor by standard simulations. Analysis of the experiments has pointed out interesting dependencies between the scaling parameter and the quantitative behaviour of the model.

In future work, we intend to incorporate other types of quantitative properties, such as the mean time a signal needs to exceed a certain threshold, the mean travelling time from the input to the output signal, etc. We also plan to analyse other biological systems like mutation rates in growing bacterial colonies [17].

REFERENCES

[1] M. Heiner, D. Gilbert, and R. Donaldson, "Petri nets for systems and synthetic biology," in SFM 2008, ser. LNCS, M. Bernardo, P. Degano, and G. Zavattaro, Eds., vol. 5016. Springer, 2008, pp. 215–264.

[2] G. Rubino and B. Tuffin, Rare Event Simulation using Monte Carlo Methods. Wiley, 2009.

[3] P. L'ecuyer, V. Demers, and B. Tuffin, "Rare events, splitting, and quasi-monte carlo," ACM Transactions on Modeling and Computer Simulation (TOMACS), vol. 17, no. 2, 2007, pp. 1–44.

[4] P. W. Glynn and D. L. Iglehart, "Importance sampling for stochastic simulations," Management Science, vol. 35, no. 11, 1989, pp. 1367–1392.

[5] B. Barbot, S. Haddad, and C. Picaronny, "Coupling and importance sampling for statistical model checking," in TACAS, ser. LNCS, C. Flanagan and B. König, Eds., vol. 7214. Springer, 2012, pp. 331–346.

[6] ——, "Importance sampling for model checking of continuous time Markov chains," in Proceedings of the 4th International Conference on Advances in System Simulation (SIMUL'12), P. Dini and P. Lorenz, Eds. Lisbon, Portugal: XPS, Nov. 2012, pp. 30–35.

[7] B. Barbot, S. Haddad, M. Heiner, and C. Picaronny, "Rare event handling in signalling cascades," LSV, ENS Cachan, Tech. Rep. 14-10, July 2014.

[8] A. Levchenko, J. Bruck, and P. Sternberg, "Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties," Proc Natl Acad Sci USA, vol. 97, no. 11, 2000, pp. 5818–5823.

[9] V. Chickarmane, B. N. Kholodenko, and H. M. Sauro, "Oscillatory dynamics arising from competitive inhibition and multisite phosphorylation," Journal of Theoretical Biology, vol. 244, no. 1, January 2007, pp. 68–76.

[10] R. Breitling, D. Gilbert, M. Heiner, and R. Orton, "A structured approach for the engineering of biochemical network models, illustrated for signalling pathways," Briefings in Bioinformatics, vol. 9, no. 5, September 2008, pp. 404–421.

[11] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, Modelling with generalized stochastic Petri nets. John Wiley & Sons, Inc., 1994.

[12] M. Heiner, C. Rohr, and M. Schwarick, " MARCIE - Model checking And Reachability analysis done effiCIEntly ," in Proc. PETRI NETS 2013, ser. LNCS, J. Colom and J. Desel, Eds., vol. 7927. Springer, 2013, pp. 389–399.

[13] L. J. Bain and M. Engelhardt, Introduction to Probability and Mathematical Statistics, Second Edition. Duxbury Classic Series, 1991.

[14] A. Jensen, "Markoff chains as an aid in the study of markoff processes," Skand. Aktuarietidskr, 1953.

[15] B. L. Fox and P. W. Glynn, "Computing Poisson probabilities," Commun. ACM, vol. 31, no. 4, 1988, pp. 440–445.

[16] P. Ballarini, H. Djafri, M. Duflot, S. Haddad, and N. Pekergin, "HASL: An expressive language for statistical verification of stochastic models," in Proc. VALUETOOLS'11, Cachan, France, May 2011, pp. 306–315.

[17] D. Gilbert, M. Heiner, F. Liu, and N. Saunders, "Colouring Space - A Coloured Framework for Spatial Modelling in Systems Biology," in Proc. PETRI NETS 2013, ser. LNCS, J. Colom and J. Desel, Eds., vol. 7927. Springer, June 2013, pp. 230–249.

Program Generation Approach to Semi-Natural Simulators Design and Implementation

Emanuil Kirilov Markov, Vesselin Evgueniev Gueorguiev, Ivan Evgeniev Ivanov
Technical University Sofia, TUS
Sofia, Bulgaria
e-mail: {emarkov, veg, iei}@tu-sofia.bg

Abstract—**This paper presents ideas for design, implementation and results of creation of computerised semi-natural simulators of the different heterogeneous objects. The presented approach extends well known solutions for object simulators creation using the program generation technique. Simultaneous co-design of the control system and the simulator is explained as well. Both stand-alone and distributed objects and control systems are included in the presented work. Examples of different implementation of the presented approach are presented.**

Keywords – semi-natural simulation; program generation; distributed control system; hardware-in-the-loop.

I. INTRODUCTION

Computer-based simulators of many different complex objects are a reality today. They have become a reality with the expansion of cheap computers since the 80’s of the 20th century. The observation of many different implementations and approaches to simulators creations are available [1][2][3].

Large-scale hazardous objects like planes and nuclear power plants have been simulated for decades. Simulation of various real systems has multiple advantages compared to experimenting and use of the actual system. Some of the advantages are: the possibility to train the personnel to operate various types of machines; to use such simulations in preparations and optimizations of control algorithms; to repeat and analyse specific situations.

The two main classes of simulators – fully numerical simulators and physical simulators mark the boundaries. The first one implements a kind of mathematical model (any type). MATLAB® is a very good example of this approach [8]. The second one is physical (material) emulation of the object. There are a lot of simulators between these two endmost types. Now, we can find Hardware-in-the-loop (HIL) [9], Software-in-the-loop (SIL) [8], Agent-based simulators [10] and other approaches to simulators building inside the boundaries mentioned above. HIL and SIL are a bit opposite because HIL means that one implements a simulation of the object when the controller is ready-for-use. The SIL means that one has a computer-based model of the object and puts in the same environment the code of the controller and runs the two together only numerically. The Agent-based simulation is a new adoption of the component-oriented programming design approach and can be discussed in this paper as a way how to present distributed/de-centralized systems and to implement their simulations.

All these simulators implement some model of the object and communicate somehow with the controller or experimental environment.

A brief examination of the papers and on-line materials today shows many different simulators and simulation environments [4][5][6].

One of the definitions of control systems architecture concerns their geographical position. The variants are ‘concentrated/centralized’ and ‘distributed’ control systems. When we talk about distributed control systems their architecture reflects the structure of the object – distributed on the level of parameters (huge objects) or distributed as points of control. In both cases the control system is influenced by the communication network and all delays introduced by it, loss of packets, etc. [3]. Using the presented below approach both concentrated and distributed systems will be covered.

The paradigm for program generation of many kinds of software is not new. Its implementation in the area of control systems varies in aspects and mathematical background but has a stable place in today’s methodologies. Here we will discuss the version of program generation when the target system is build using pre-programmed library components which are only instantiated and linked in the real implementation. The full code-generation like the one used in MATLAB Embedded Studio [8] or ADA [11][12] based real-time systems is out of the scope of this paper.

The approach presented in this paper is focused on using one and the same tool (program generator) to design both the HIL-type “semi-natural” object simulator and the control system.

The name “semi-natural” means that the simulator is not only a computer, running the object model and connected to the controller via its physical interface, but additionally that the simulator can include parts of the real object’s hardware. In this case we have something that is mostly a simulator but has some elements of an emulator.

The present paper is structured as follows: Section II resents the proposed connection between control systems, objects and their simulators; Section III presents a short description of the used program generator and the formal model implemented by it; Section IV presents the implementation and analyses of several objects and their simulators; Section V is the conclusion.

II. BACKGROUND OF THE PRESENTED APPROACH

To implement a simulator using mathematical models of the object is an old and widely used idea. To implement an object emulator is a very old idea, too. To implement a simulator using both a mathematical model and elements of some physical hardware is a more recent one. To implement a simulator using a physical interface, which is the same as the original object interface is a much newer idea. It has been possible to do that only for the last 20 years when both computer hardware and peripheral devices became cheap enough and versatile. To implement a simulator for a distributed object is an old idea but it has been possible to implement it only in the last few years after the advance in communications and especially in guaranteed real-time communications.

A complex control system has a structure similar to that shown in Figure 1.

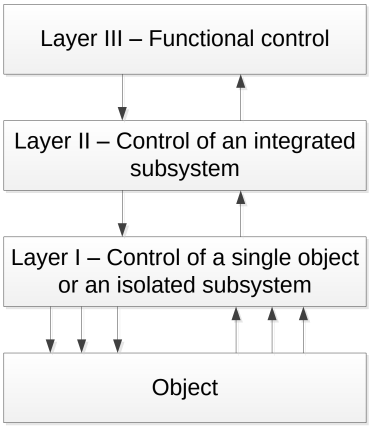

Layer III – Functional control
Layer II – Control of an integrated subsystem
Layer I – Control of a single object or an isolated subsystem
Object

Figure 1. General structure of a multi-layered control system.

Here, we assume the object to be a single one. Later, we will discuss how that view can be expanded for distributed objects. When we implement a computerized simulator with a real peripheral device, we can present it on the diagram shown in Figure 2.

When we combine both a control system and a simulator, the resulting diagram looks like the one shown in Figure 3.

This approach allows different ways to simulate the object to be used:

- full simulation using a physical interface of the same type as the real object (and possibly including parts of hardware from the object);
- partial simulation using a physical interface, emulating the real interface;
- partial simulation using signal exchange based on a type of networking.

In [1], several different variants for simulation are described but here we discuss only those that include parts of real physical hardware.

In all cases, if the object and/or control system has HMI it is presented “as is” in the couple “control system-simulator”.

Discussing simulation of a distributed object, we have to include simulation of its distributiveness in the object properties. Depending on the type of the distributiveness we can simulate transport delays, internal status (as temperature) propagation delays and inequalities, distribution of process run (like conveyor and machinery near it) and so on. The complexity of that simulation can be higher than the complexity of the control system. This complexity increases in the case shown in Figure 3, when the control system and the simulator are connected via an additional “line” for activity synchronisation.

III. THE PRGEN – PROGRAM GENERATOR FOR DISTRIBUTED CONTROL SYSTEMS AND OBJECT SIMULATORS

The program generator used for establishing a simulation model is designed for creating distributed real-time control systems. It has been implemented in many different versions over the years [14][15][16]. It is based on an extended Moore machine implementing specific actions in each node of the state machine. Specific elements of its design reflect the possibility to generate both stand-alone and distributed systems. The generated distributed systems can operate as a “virtual mono-machine” or as a component-based (or agent-based) system. A graph representation of the control algorithms is chosen. The system can be described by its activities. Each activity is a separate thread. The activity thread consists of two different graphs:

- A State Transition Graph (STG) – a graph model for modelling the finite automaton, describing the general behaviour of the activity thread. The graph is represented like statechart described in [17].
- A Signal Flow Graph (SFG) – a graph model representing the signal transformation flow (the dataflow [18]). It is built by Function blocks, very similar to Simulink®. It is mainly used to model the continuous part of the system, to handle the I/O and communication drivers and to calculate complex predicates used in transitions of the State Transition Graph.

A. State Transition Graphs

Each system node has at least one activity thread. This thread is implemented by its STG defining the logical behaviour (although there are some implementations containing only one state with an infinite loop to itself). The STG has one entry point (initial node) with no other function than pointing where exactly the execution of the STG should start when the system is started. For each state of the STG one or more SFGs can be attached. For each state one or more transitions should be defined. A transition to the same state is acceptable. Decision making for transition to be performed is based on an associated to each state Binary Decision Diagram (BDD). Values for the BDD’s predicates are taken from the node SFGs.

For each STG, an execution period is defined, defining how often the graph activates and executes its current state. The execution period can be modified if necessary during runtime (e.g., when the system is in idle/power-down state scanning inputs and refreshing outputs at a high rate are not necessary).

There are two types of transitions defined: synchronous and asynchronous. When a synchronous transition is

ISBN: 978-1-61208-371-1

executed the graph execution stops and the task goes in sleep mode until the execution period expires. When an asynchronous transition is activated, the task goes directly to execution of the state following the transition, without waiting for the period to expire.

B. Signal Flow Graphs

The SFG models the data flow of the system. Typically the SFG entry points are Function blocks representing I/O drivers, communication drivers or in specific cases data calculated by other SFGs. Then the data is passed to other Function blocks, which make transformations, check constraints and conditions and produce output for I/O drivers, communication, user visualization, database logging, etc. Each Function block in the SFG is executed only once per SFG execution. This is ensured by the connections between them. There are two types of connections: activating and non-activating. The SFG execution starts with Function blocks which do not have activating inputs. After their execution, the remaining blocks are checked for activation. The blocks that have all their activating inputs set, start execution and continue until the last block of the SFG finishes its job.

Each Function Block (FB) can have up to three types of inputs and two types of outputs.

The inputs of FB can be: Link Inputs (activating or non-activating); Parameter Inputs; Internal State Inputs. Each input can be linked to only one data source. An input which takes data from two different sources (e.g. from the outputs of two separate preceding FBs) is not available.

FBs can have two types of outputs. Static outputs – can be a data source of unlimited number of link inputs. Point to point outputs – cannot be a data source of inputs. Instead, they can be linked to a static input and change their value.

All inputs and outputs can hold matrix values, allowing complex system models to be generated.

What makes this implementation different from many others is that a communication subsystem is implemented as a number of communication modules using both logical and physical data exchange protocols. Using this approach the general model of the implemented system – control or simulator, is virtual mono-machine. The communication bus is implemented hidden, but observable [20][22].

Extended description of the model of the presented program generator can be found in [14][19].

IV. SIMULATOR PROJECTS

Hereafter, we will present three different objects where the control system and the simulator were implemented using the presented approach. All objects have dominantly analogue behaviour, but they are very different in size and general complexity. A comparison between the implemented control system and simulators will be provided as well.

These objects are a fuel tank farm, a business building and a machine for making sausages (food industry). The first two objects are big distributed objects and the third one is a stand-alone machine. The following points will be discussed for each object: 1) why we need to build a simulator; 2) real object structure; 3) how the simulator is built, and 4) problems when the program generator was used to implement the simulator and how they were solved.

A. Tank farm control system and simulator

The tank farm control system includes the loading and unloading of tanks, calculation of the available fuel based on several different measured parameters, control of the pipe system. The control system is built by using an industrial OPC-based system connected to the field devices via Profibus connections (both DP and PA).

The need to invest time and money in a simulator of the tank farm has several dimensions: 1) it is used to design and tune different elements and the integrated control system; 2) it is used to train operative personnel; 3) it is used to analyse situations like leaks, inconsistency between different meters, etc.

Each tank has several integrated sensors for external and internal temperature, density, height of the fuel in the tank. Internal temperature is measured using distributed (multipoint) thermometer measuring temperature in several points on different depth in parallel. The density meter and level (height) of the fuel is traversed in the fuel starting from level 0 and finishing at the bottom of the tank. The pipe system includes mass-meter computerized devices and valve control. Valves can control fuel flow but they do not have internal position stabilization and have to be controlled by the main system. A general view of a tank, its sensors/actuators and pipes, is shown in Figure 4.

Having more than 10 different tanks and a large number of pipes makes the control and measurement task rather complicated. All tank sensors and actuators were connected to the upper level control system via Profibus PA in compliance with explosion hazardous areas safety requirements. Massmeters and valves as specific hardware were simulated using other mixed SW-HW simulators of the devices. With this simulating environment in mind, a construction for object simulation was built. The structure of the multilayered tank simulator is shown in Figure 5.

The tank simulator is built using the program generation approach. It is generated by the program generator using building blocks from the block library (pre-programmed and pre-compiled elements like “PID-controller”, “first-order filter”, “ADC driver”, etc.). As has been said, the tank simulator is a multi-layered system. Each element is simulated separately. Every local simulator is implemented using Single Board Commuter (SBC) with an ARM core and the necessary peripheral devices. Each SBC is driven by RTOS and the real-time part of the program generator – the RT interpreter and communication library.

Every low-level element is connected to the upper level simulator layer which coordinates them, implements an upper-level tank model and logic and supports communication to the upper levels of the system simulator. Thus, every complex tank simulator operates as a component of components. Using this approach and combining this simulator for every tank with simulators for connecting pipes and switching valves, we designed and implemented the tank farm simulator. The implemented simulator operated in various types of modes for normal and abnormal operation,

ISBN: 978-1-61208-371-1

to work in real-time and simulated time-flow with the control system. It covered the following areas of use:

- Control functions test and tuning (control system tests, tuning and development);
- Operators education/training;
- Abnormal situations analyses.

The main problems of the implemented simulator were two: 1) the Profibus network; 2) the number of elements to be simulated.

The Profibus communication has one positive side – it has a well-known traffic scheme and communication load and introduced delays can be calculated. It has one drawback – it is very hard to simulate in a real environment because developing of a Profibus slave is a hard, slow, specific and expensive task. To avoid this problem we implemented Master-Slave network over RS 422 media with an upper level protocol similar to the Profibus. All delays were included in the network modules. Removing Profibus from the system for simulation purposes we switched all sensors and actuators to other (supported by them) networks.

The solution of the problem with the number of the simulated elements was solved using an approach from the computer science. We had several tank types, valve types, pipe types and communication links. A template configuration for every different type of device was created. An instance of every specific device was parameterized and loaded into a SBC. All SBCs were connected following the connection scheme. Thus, implementing each template only once and configuring it for every specific instance all simulator elements were built. Based on the component-based system structure, they were connected one-by-one to the controller. This allowed the design group to verify and validate the solution starting from low and going to high system complexity.

B. Business building simulator

The next object that will be presented is a business building. It has many different stores, restaurants and other objects.

The decision to build a simulator for building control purposes was taken for several reasons: 1) experiments with the real equipment are possible and real data were collected but they are expensive and after the beginning of the official object exploitation some of them became impossible (because they include measuring devices of utility companies – electricity, water); 2) Experiments with different scenarios of exploitation are impossible on the real object; 3) training of the exploitation personnel has to be done periodically; 4) the building owner needs an experimental field test for malfunction and abnormal situation analyses.

Building control and data acquisition included electricity load control, all separate electricity, water and heat metering devices for every object, implementation of a number of scenarios for lighting, heating and other equipment control, control for abnormal situations and functioning, full time system log, etc. The building has several hundred sensors – smart or simple, several hundred actuators, switchers and intelligent output devices. This complex object has multi-layered requirements for exploitation. This requires a multi-layered control system. Simulation and testing of the full control system is impossible using standard approaches for computer simulation.

The designed simulator consists of many low-level simulators for every single object. Controlled sub-objects can be separated in several classes of similarity. A single template for every object class was designed. After that they were instantiated and parameterized. The final structure is similar to that shown in Figure 3. The final simulator system has to have about 3000 different function blocks for 7 computational and 3 logical low-level configurations and 2 upper levels implementing simulation of integrated functions. It is very hard to implement such a system. That is why the simulator size was reduces to 1/10 of the real size but including all computational and logical structures. Additionally, modules were implemented simulating human streams, some behavioural scenarios, and abnormal situations in the electricity and water supply (malfunctions injectors). All simulation modules were implemented using SBC ARM-based computers with the appropriate physical periphery and communication. From a generalized point of view the controlled object is a heterogeneous distributed discrete-event system including a number of analogue inputs and outputs and several sub-objects of analogue type of functionality. Many of the controlled elements communicate to the controller via MBUS. The structure of the MBUS connections is shown in Figure 6.

The control system includes a number of PLCs connected to their upper level (a SCADA system running on an industrial PC computer).

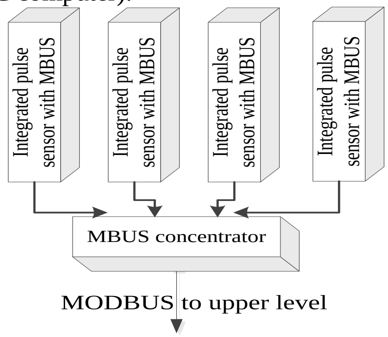

Integrated pulse sensor with MBUS
Integrated pulse sensor with MBUS
Integrated pulse sensor with MBUS
Integrated pulse sensor with MBUS
MBUS concentrator
MODBUS to upper level

Figure 6. MBUS to MODBUS connection

Every control loop is simulated by implementing an object model running on the SBC under RT OS and the real-time interpreter of the generated by the program generator configuration. One of the most important features of the simulator is the ability to be re-loaded with different system internal statuses (contexts) and to re-execute situations which have happened and have been logged. Additionally, the simulator can run in parallel with the real system in fast time and to predict the object behaviour. The implemented integral HMI was used to train building operators to understand and control its functioning and all included subsystems. Additionally, full database for process logging is included.

As in the previous example, one of the problems was the fact that simulation of a MBUS slave is hard. Fortunately, we did not need to simulate it to test functional implementation. There are a lot of MODBUS simulators for the upper level implementations. The communication library of the program generator was extended with modules implementing upper levels of MODBUS protocol. They were connected to the simulator of the low-level pulse sensors. The transmission from the simulation pulse module to MODBUS module was done using substitution of MBUS by I^2C, but following the scheme in Figure 6. All upper layers were kept as they were.

C. Food industry machine simulator

The third object that will be presented is a relatively small system from the food industry. It is a machine for making sausages. The size of the machine is about 2×2×3 m. It has very non-linear behaviour depending on many uncontrollable external disturbances – size and quantity of the prepared sausages or meat, quality of the bran for the smoke generator, variations of the current voltage and others. The mathematical model of the machine is statistical. It is a typical non-linear object with internally distributed parameters. Experiments with this machine are expensive and more over they are in-repeatable in general. Testing of the controller and its software is hard. It was decided to design and implement an object simulator to prepare an experimental environment with predictable behaviour and make testing and tuning of all control equipment repeatable. The simulator was based (as before) on a SBC computer with physical periphery as the real object. All analogue and discrete signals were implemented using process I/O devices with high resolution and precision. The mathematical model of the object was implemented using library modules from the program generator library. Additionally, a communication channel to the upper level of the simulator was implemented, which was used to set different parameters of the model (to switch between different sausages and meats and to simulate differences between different pieces of them in the machine). Using this component approach and program generation a simulator was built which covered the real machine behaviour up to 98%. The controller was designed to meet all project requirements. The main problem in this simulation was to build-in enough possibilities to induce disturbances in the object but, because this is mostly mathematical and logical and not a hardware problem. The controller was implemented using standard modules from program generator's library. The only problem was to connect triggers activating these modules and their re-parameterization to the upper level of the implemented simulator. Using external data lines similar to the ones shown in Figure 3 this was implemented.

V. Analyses of the presented approach for simulators implementation using program generation

In the paper, three different objects and the approach to designing their simulators were demonstrated. One is a small but very complex food machine, the second is a huge distributed dominantly discrete object (a business building) and the third is a heterogeneous distributed object including transport delays, a lot of non-linearites and additionally several problems in communication simulation. All designed simulators were of the type 'semi-natural' or 'partial emulators'. They include both specific and general-purpose hardware and operate in real-time software. The presented objects are very different. These simulators are based on one and the same approach – a component design using an automated tool, a program generator. This generator produces a system configuration that is executed by the real-time system. All hardware for the simulators is similar – SBC with ARM processors and physical periphery of the type similar to the real object hardware interface. Depending on the object size the real simulator is implemented using one or several SBCs. They are connected to the implemented control system via analogue, discrete, pulse and communication interfaces.

Analyses of the complexity of the implemented simulators compared with the control system show that they are on similar levels. In all three situations the design phase of the simulator facilitating the good understanding, modelling and design of the control system. The possibility to use simulators running with time speed different form the real time enabled both fast checks of situations and real-time prediction of eventual dangerous object behaviour to be conducted. The numerical nature of the simulators and control systems enables situation analyses using system logs and other status information for events that have happened. Education and training of operators of those systems is based on the same principles. The main difference is the size of the simulator. Hardware implementation depends on the number of inputs, outputs and communication lines to be simulated. The possibility of the program generator to build templates and to instantiate them by sets of real parameters speeds up many times the generation of systems with big number of similar elements (as every component-based system).

Comparing the time necessary for the simulators building we will say that the most hard for design from the modelling point of view was the food machine simulator. The most time consuming was the building simulator because it had the biggest number of I/Os and real elements to be simulated even using template instantiation.

VI. Conclusion

The paper presented an approach to building object simulators using program generators and one and the same toolset for the control system and simulator implementation. The presented semi-natural simulators enabled the control system and the simulator to be designed simultaneously. This approach reduces the investments and risks in the design and implementation phases of the control system design. The possibility to use that implemented simulator not only for control system tests but for personnel training and for events and abnormal situations analyses makes them a helpful, relatively inexpensive tool with great flexibility and versatility. The difference from other HIL is the ability to include easily parts of the real hardware together with the simulated one.

ACKNOWLEDGMENT

Parts of the presented work are funded by the Bulgarian NSF under DRNF02/3 project.

REFERENCES

[1] J. A. Carrasco and S. Dormido, Analysis of the use of industrial control systems in simulators: State of the art and basic guidelines, ISA Transactions, Volume 45, no. 1, January 2006, pp. 295-312

[2] A. Negahban and J. S. Smith, Simulation for manufacturing system design and operation: Literature review and analysis, Journal of Manufacturing Systems, Volume 33, Issue 2, April 2014, pp. 241–261

[3] W. Li, X. Zhang, and H. Li, Co-simulation platforms for co-design of networked control systems: An overview, Control Engineering Practice vol.23, 2014, pp. 44–56

[4] The Rapid Automotive Performance Simulator (RAPTOR), http://www.swri.org/4org/d03/vehsys/advveh/raptor/default.htm [last accessed: 08.08.2014].

[5] M. Pasquier, M. Duoba, and A. Rousseau, Validating Simulation Tools for Vehicle System Studies Using Advanced Control and Testing Procedure, http: //www.autonomie.net/docs/6 - papers/validation/validating_simulation_tools.pdf [last accessed: 08.08.2014].

[6] IAEA, Use of control room simulators for training of nuclear power plant personnel, Vienna, 2004, IAEA-TECDOC-1411, ISBN 92–0–110604–1.

[7] M. Johnstone, D. Creighton, and S. Nahavandi, Enabling Industrial Scale Simulation / Emulation Models, In Proceedings of the 2007 Winter Simulation Conference, 2007, pp. 1028-1034.

[8] MathWorks, Generate and verify embedded code for prototyping or production, http://www.mathworks.com/embedded-code-generation/, [last accessed: 08.08.2014].

[9] J. A. Ledin, Hardware-in-the-Loop Simulation, Embedded Systems Programming, Feb. 1999, pp. 42-60.

[10] C. Macal and M. J. North. Agent-based modeling and simulation. In Proceedings of the 2009 Winter Simulation Conference, ed. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc., 2009, pp. 86-98.

[11] D. A. Watt and B. A. Wichmann, W. Findlay, "Ada: Language and Methodology." Prentice-Hall, 1987

[12] A. Burns and A. Wellings, Real-Time Systems and Programming Languages (Fourth Edition) Ada 2005, Real-Time Java and C/Real-Time POSIX, April 2009, Addison Wesley Longmain, ISBN: 978-0-321-41745-9

[13] N. Baldzhiev, V. Bodurski, V. Gueorguiev, and I. E. Ivanov, Implementation of Objects Simulators and Validators using Program Generation Approach, DESE 2011, Dubai, UAE, December 2011

[14] C. K. Angelov and I. E. Ivanov, “Formal Specification of Distributed Computer Control Systems (DCCS). Specification of DCCS Subsystems and Subsystem Interactions”. Proc. of the International Conference “Automation & Informatics’2001”, May 30 - June 2, 2001, Sofia, Bulgaria, vol. 1, pp. 41-48.

[15] I. E. Ivanov and K. Filipova, “Integrated scheduling of heterogeneous CAN and Ethernet-based hard Real-Time network”, Proc. of IEEE spring seminar 27th ISSE, Annual School Lectures, Bulgaria, 2004,vol. 24, pp.481-485

[16] I. E. Ivanov and V. Georgiev, “Formal models for system design”, Proc. of IEEE spring seminar 27th ISSE, Annual School Lectures, Bulgaria, 2004,vol. 24, pp. 564-568

[17] D. Harel, “Statecharts: A visual formalism for complex systems” Science of Computer Programming 8 (1987), pp. 231-274

[18] K. M. Kavi and B. Buckles, “A Formal Definition of Data Flow Graph Models” IEEE Transactions on computers. vol. C-35, no. 11, November, 1986

[19] I. E. Ivanov, “Control Programs Generation Based on Component Specifications”, PhD thesis, 2005, Sofia, (in Bulgarian)

[20] C. K. Angelov, I. E. Ivanov, and A. A. Bozhilov. Transparent Real-Time Communication in Distributed Computer Control Systems. Proc. of the International Conference “Automation & Informatics’2000”, Oct. 2000, Sofia, Bulgaria, vol.1, pp. 1-4

[21] A. Dimov, I. E. Ivanov, and K. Milenkov, "Component-based Approach for Distributed hard Real-time Systems", Information Technologies and Control, 2, 2005

[22] A. Dimov and I. E. Ivanov, Towards development of adaptive embedded software systems, Proceedings of TU Sofia, vol. 62, book.1, 2012, pp. 133-140

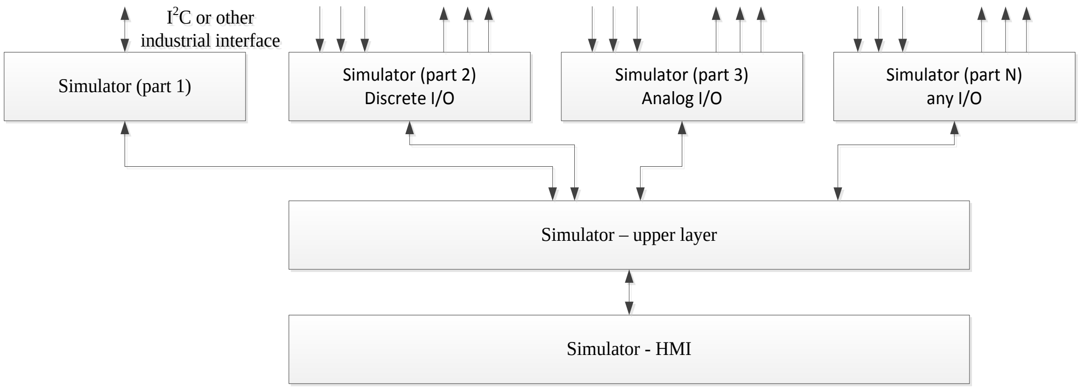

I^{2}C or other industrial interface
Simulator (part 1)
Simulator (part 2) Discrete I/O
Simulator (part 3) Analog I/O
Simulator (part N) any I/O
Simulator – upper layer
Simulator - HMI

Figure 2. General structure of a multi-layered semi-natural simulator.

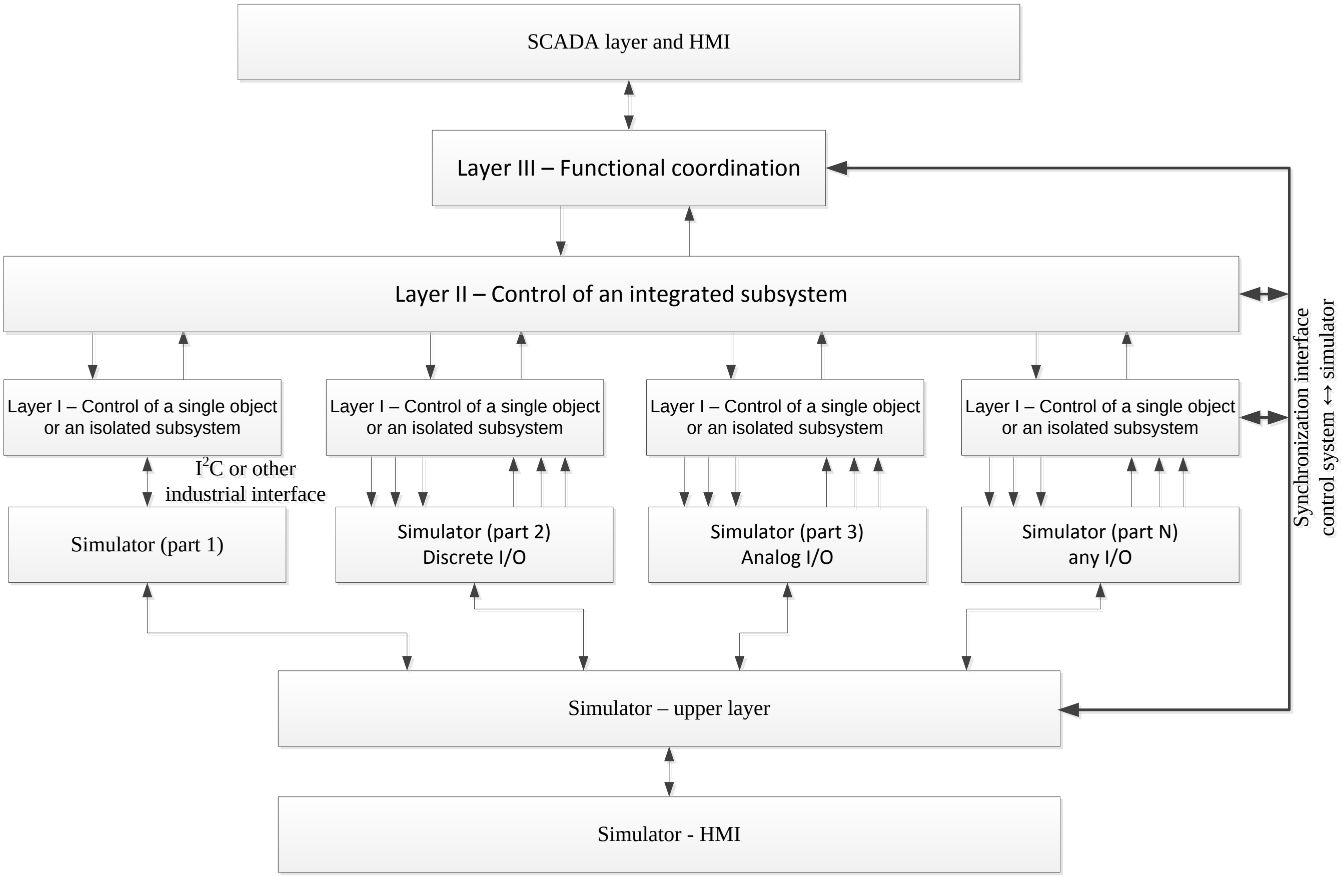

SCADA layer and HMI
Layer III – Functional coordination
Layer II – Control of an integrated subsystem
Layer I – Control of a single object or an isolated subsystem
Layer I – Control of a single object or an isolated subsystem
Layer I – Control of a single object or an isolated subsystem
Layer I – Control of a single object or an isolated subsystem
I²C or other industrial interface
Simulator (part 1)
Simulator (part 2) Discrete I/O
Simulator (part 3) Analog I/O
Simulator (part N) any I/O
Synchronization interface control system ↔ simulator
Simulator – upper layer
Simulator - HMI

Figure 3. Combined structure “control system ↔ simulator”

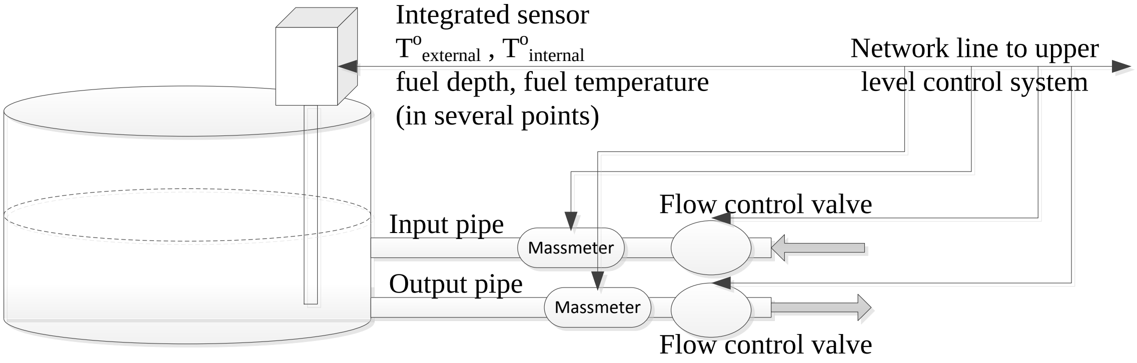


Figure 4. Fuel tank – general structure and sensors/actuators

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-371-1

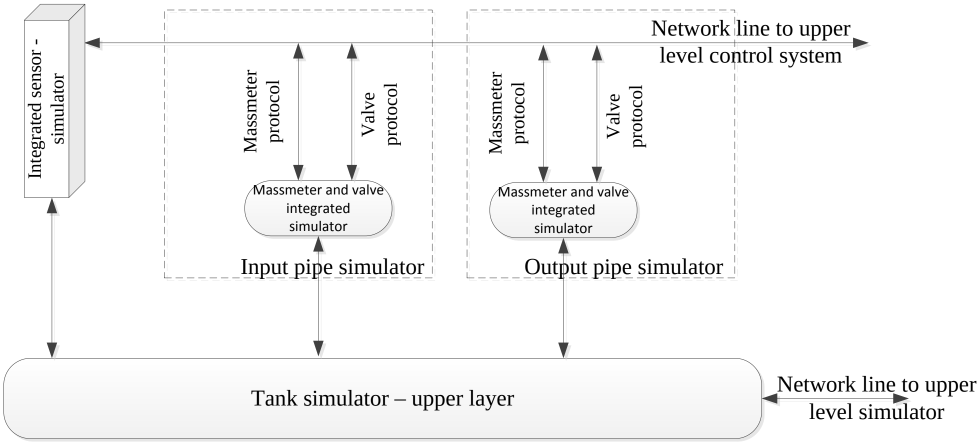

Integrated sensor - simulator
Massmeter protocol
Valve protocol
Massmeter protocol
Valve protocol
Network line to upper level control system
Massmeter and valve integrated simulator
Massmeter and valve integrated simulator
Input pipe simulator
Output pipe simulator
Tank simulator – upper layer
Network line to upper level simulator

Figure 5. Fuel tank – simulator structure

Semi-automated Generation of Simulated Software Components for Simulation Testing

Tomas Potuzak
Department of Computer Science and Engineering
Faculty of Applied Sciences, University of West Bohemia
Univerzitni 8, 306 14, Plzen, Czech Republic
e-mail: tpotuzak@kiv.zcu.cz

Richard Lipka
Department of Computer Science and Engineering/
NTIS – European Center of Excellence
Faculty of Applied Sciences, University of West Bohemia
Univerzitni 8, 306 14, Plzen, Czech Republic
e-mail: lipka@kiv.zcu.cz

***Abstract*—Using a component-based software development, applications can be constructed from individual reusable software components providing particular functionalities. The testing of a particular component is very important not only individually, but in the context of its neighboring components as well. In the SimCo simulation tool, which we developed, it is possible to perform tests of the software components in a simulation environment. For this purpose, the neighboring components of the tested components can be replaced by their simulation models. In this paper, we describe an approach for the semi-automated generation of the simulation model of a software component based on the analysis of its interface and on the observation of its behavior among its neighboring components.**

Keywords-software component; simulation testing; simulation model generation.

I. Introduction

Component-based software development is a spreading trend in contemporary software engineering. The main idea of this approach lies in the utilization of isolated reusable parts of software, which provide functionality via services. These parts – called *software components* – can then cooperate and form an entire application. Each component has a defined interface, which is a set of services the component provides. The component may also require services of other components in order to work. A single component can be used in multiple applications and particular components in a single application can be provided by different developers [1]. This reinforces the need for the testing of software components and their interactions.

The testing of software components should be ensured by their developers. However, it is also useful to test the functionality, quality of services, and extra-functional properties of the component while it is interacting with other components forming together an application. Such kind of testing has to be performed by programmers who did not create the components themselves, but use them to make up a new application. So, the source code, descriptions, and other resources may be unavailable to them. In an extreme case, only the public interfaces of the components may be known.

During our research, we developed the SimCo simulation tool, which enables simulation testing of particular components, sets of components, or entire applications directly – without creating additional models of the tested components [2][3][4]. When only a part of a component-based application (i.e. a single component or a set of components) is tested, it is necessary to satisfy all dependencies of the tested components. More specifically, the tested components can require services of other components, which are not considered necessary for the testing and thus are not present in the simulation environment. In the SimCo simulation tool, these required components are replaced by their simulation models in order to satisfy the dependencies of the tested components. They have the same interfaces as the components they are replacing, but the original functionality can be substituted for example by lists of pre-calculated values or random numbers generators [5].

So far, the behavior of the simulation models of components for the SimCo simulation tool is created manually, which is a lengthy and error-prone process. In this paper, we describe an approach for semi-automated generation of the simulation models of components, whose implementation is at our disposal, but the source code is not. The approach is based on the analysis of the public interface of the component and on the observation of the behavior of the component among its neighboring components. The result of the approach is a generated skeleton of the simulation model of the component with partial functionality and clues, which can be used by a programmer for finishing of the functionality of the simulation model of the component.

The remainder of this paper is structured as follows. Software components are described in Section II. In Section III, the simulation testing is discussed. Section IV describes the SimCo simulation tool. In Section V, the related work is discussed. The generation of simulated components demonstrated on a case study is described in Section VI. The future work is described in Section VII and the paper is concluded in Section VIII.

II. Software Components

Before we proceed with the description of our approach, we will briefly discuss the basics of the component-based software development.

A. Basic Notions

Using the component-based software development, the applications are constructed from software components. A software component is a black box entity with a well defined public interface, but no observable inner state. So, the components mutually interact solely using their interfaces. They should be reusable and the author of an application can be different from the author(s) of the particular components. This general definition is common for most component models [1].

A component model describes how the particular software components look, behave, and interact. A component framework is then a specific implementation of a component model and there can be several different implementations of each component model [1]. The existing component models are generally incompatible, since they use different approaches to solving particular issues of the software components.

B. OSGi

An example of a widespread component model is the OSGi (Open Service Gateway initiative), which is contemporary used in both industrial and academic spheres. There are several OSGi frameworks (i.e. implementations of the OSGi model), which are commonly used [6]. The OSGi component model is designed for Java programming language. It is dynamic in nature, which means that it enables to install, start, stop, and uninstall particular software components without the need to restart the OSGi framework itself [7].

A software component in the OSGi component model is referred as a *bundle* and has a form of a standard Java `.jar` file with additional component-model-related information (e.g., lists of services provided by the bundle and list of services required by the bundle). Each bundle can contain any number of classes. So, it can provide arbitrary complex functionality [6]. As the interface of the bundle, standard Java interfaces are used. So, the particular services provided by a bundle have the form of standard Java methods [6].

Since the OSGi component model is widespread and the transformation of one component model into another is difficult (see Section II.A), the SimCo simulation tool was developed solely for the OSGi [2]. However, the main ideas and approaches utilized in it can be used in other component models as well.

III. Simulation Testing

Now, as we described the basics of the component-based software development, we will briefly discuss the simulation testing.

A. Discrete-Event Simulation

A discrete-event simulation is a widely used simulation technique. The simulation run is subdivided into a sequence of time-stamped events representing incremental changes of the simulation state. The simulation time between two succeeding events can be arbitrary long. Nevertheless, the simulation does not perform a real waiting for the specified time. Instead, the simulation time is set to the time stamp of the event, which is processed. So, the simulation time "jumps" from the time stamp of an event to the time stamp of the next one [8].

The events are handled by a calendar, which incorporates the list of events ordered by their time stamps. At the simulation run start, the calendar removes the first event from the event list, sets the current simulation time on the time stamp of the event, and performs the action (i.e. a change to the simulation state) associated with the event. This action can (but does not have to) create one or more new events, which are added to the event list of the calendar on the positions corresponding to their time stamps. Then, the next event is removed from the event list and the entire process repeats until a stop condition is reached or the event list is empty [8].

The discrete-event simulation, which was briefly described in previous paragraphs, is used in the SimCo simulation tool [2].

B. Simulation Testing of Software Components

Although the simulation testing can be used for various systems, for the needs of this paper, we will focus on the simulation testing of software components. There are two approaches that can be used.

The first approach is to create a simulation model of the software component, which shall be tested. This model is then used in the simulation [9]. The advantage of this approach is that the simulation model of the software component is suited for the simulation. The disadvantage is the necessary creation of the model, which can be often a lengthy and error-prone manual process. Moreover, the model usually does not incorporate all aspects of the original software component, but only aspects, which are considered important for the simulation testing by the creator of the model. This can lead to an unintentional omission of features, which in fact can be important for testing. These disadvantages can be avoided or at least diminished by a full or a partial automation of the process of the creation of the component simulation model.

The second approach is to use the software component in the simulation directly. The advantage of this approach is that there is no need for creation of the simulation model of the tested component and the component is tested directly. So, its responses to the stimuli induced during the testing are genuine. The main disadvantage is that the component is not suited to run in a simulation environment. Hence, the simulation environment must be able to allow running of the software component as it would run in a real application. It should also be noted that the problems connected to the creation of simulation models (see previous paragraph) are not fully avoided by the direct testing of the software component in a simulation environment. The simulation model of the tested component is not created, but it is often necessary to create simulation models of other components required by the tested component. Similarly to the first approach, the disadvantages related to the simulation models creation can be diminished by automation of the creation process.

The simulation testing of the software components can be further divided according to the knowledge of the tested components. If their source code is known, it can be used for preparation of the tests. This is so-called *white box* testing. If the source code in not known, the tests must be prepared using interfaces or other available specifications of the tested components. This is so-called *black box* testing [10].

IV. SIMCO SIMULATION TOOL

The SimCo simulation tool, which we developed, enables to test the software components directly in the simulation environment. Its main features are described in the following sections.

A. SimCo Features

The SimCo is written using the Java programming language. It is based on the OSGi model [6] and, currently, it is running in the Equinox OSGi framework [11]. Hence, the software components, which can be tested within the SimCo, are limited to OSGi bundles. This restriction is not so hard, since the OSGi is quite widespread in various industry areas as well as in the academic sphere (see Section II.B).

Within the SimCo simulation tool, it is possible to test a single component (i.e. an OSGi bundle), set of components, or an entire component-based application. The SimCo itself is constructed from the components (OSGi bundles) as well in order to enable its simple modifications and extensions [5].

For the simulation testing, the SimCo utilizes the discrete-event simulation (see Section III.A) where the events correspond to the invocations of particular services of the tested components [4]. The initial events, which shall be performed during the simulation run, are imported to the event list from a testing scenario. This testing scenario is loaded from an XML file prior to the execution of the simulation testing [5].

B. SimCo Software Component Types

There are four types of the software components in the SimCo simulation tool – the *core* components, the *real tested* components, the *simulated* components, and the *intermediate* components [5].

The core components provide the functionality of the SimCo itself. These software components ensure the running of the simulation. Furthermore, they provide all necessary supplementary functions such as logging, measuring, and storing of the observed parameters, visualization of the simulation, and so on. The most important core component is the `Calendar`, which interprets the events from its event list. More specifically, it invokes corresponding services of the software components and so ensures the advancement of the simulation in time [5].

The real tested components are the components which are tested in the SimCo simulation tool. These components can be created by various manufacturers. So, the source code of the components may not be available to us, which implies the black box testing (see Section III.B). Since the components are running in the simulation environment, there are several issues, such as discrepancy of the real and the simulation time, undesirable hidden network communication of the real tested components, and so on. These issues can be solved using various means including aspect oriented programming or changing of the bytecode of the tested components. For more information, see [4] and [5].

The simulated components are simulation models of components, which provide services required by the real tested components. They are not necessary when an entire component-based application is tested within the SimCo simulation tool. However, when only a single component or a limited set of components are tested, components providing some services required by the real tested components do not have to be present in the simulation environment. The reason is that these components are not considered important for the testing, for example because they were already tested before and/or the execution of their services is time-consuming. Nevertheless, it is necessary to satisfy the requirements of the real tested components. So, the required components, which are not present in the simulation environment, are replaced by their simulation models – the simulated components. The simulated components can be useful for the speedup of the simulation, because they do not have to perform the calculations of the original components they are mimicking. Instead, prerecorded values, table of return values, or random numbers generators can be used, depending on the situation [5]. At the same time, the simulated components must exhibit the same external behavior as the original components they are mimicking. For example, if the consequence of a service invocation on the original component is the invocation of a service on another component, this must be true for the simulated component as well. The simulated components also ensure the control of the simulation by the SimCo simulation tool. All invocations of the services of the simulated components are performed using the events and the calendar [5].

The intermediate components are used to ensure the control of the simulation by the SimCo simulation tool even between two real tested components. More specifically, an intermediate component is used as a proxy for a real tested component. It provides the same services and all service invocations on the real tested component are in fact handled by this proxy (see Figure 1). The intermediate component ensures that the service invocation on a real tested component is performed using the events and the calendar, similar to the simulated components (see previous paragraph) [5]. Moreover, it performs the invocation of the same service on the real tested component, for which it serves as a proxy. This is necessary in order to obtain the return value and/or trigger further actions (e.g., an invocation of a service on another component).

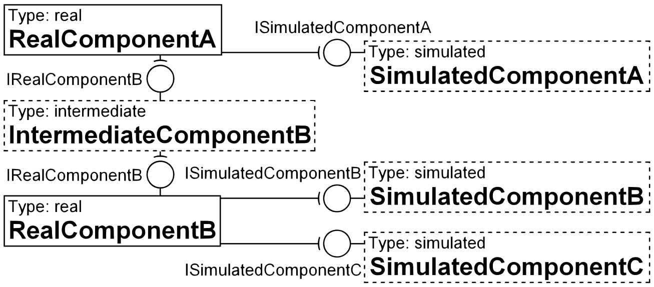


Figure 1. An example of the SimCo component types

The simulated components and the intermediate components can also be used for the placing of probes observing selected properties of the real tested components. For example, it is possible to use the intermediate component to measure execution times of the services of the real tested component, for which the intermediate component acts as its proxy [5].

An example of three component types (except the core components) of the SimCo simulation tool is depicted in Figure 1.

V. Related Work

The generation of the simulation models of software components, whose implementation is at our disposal, but their source code is not, is not common and it is difficult to find description of similar approaches in the literature. However, it is related to the black box testing of software components, since the inspection of the component behavior is performed in both cases. The relevant techniques are mentioned in the following sections.

A. Reverse Engineering

Reverse engineering is a process of analyzing a system (a software component in our case) and creating the representation of the system in another form or at a higher level of abstraction [12]. Reverse engineering can be used even when the source code is known (i.e. during white box testing) for example for creation of UML (Unified Modeling Language) diagrams or control-flow graphs. For this purposes, the static or dynamic analysis can be used [12]. In situations when the source code is not known (i.e. during black box testing) the reverse engineering can be used for the extraction of the source code from the binary representation of the application [13]. In our case, the application is the software component (an OSGi bundle) and the binary representation is the bytecode of its `.class` files.

There are several disadvantages of reverse engineering used for the obtaining of the source code. The names of the methods and parameters are lost, since they are irrelevant (and thus not present) in the binary representation. Similarly, the comments cannot be restored for the same reason [13]. So, it may be difficult for a human programmer to understand the obtained source code.

Although the reverse engineering of the Java bytecode is possible and, using specialized tools, quite straightforward, all mentioned issues stand. Moreover, our intention is not to replicate the exact inner functioning of the components. The simulated components should exhibit the same external behavior as their original components, but their inner functioning can be significantly different (e.g., in order to reduce the computation time). So, even with a successful reverse engineering of the bytecode, the analysis and modifications of the obtained source code by a programmer is still necessary. Due to these issues, a significant amount of manual work may be required.

B. Interface Probing

Interface probing is a technique that utilizes the public interface of the software component for the examination of its behavior and its features. This technique can be readily utilized during the black box testing, since the source code is not necessary. Using the interface probing, the public interface with the services of the tested software components is identified first. Then, the input values for the services are generated, the particular services are invoked with the generated inputs, and the outputs of the services of the component are observed [13][14]. For this purpose, the component can be wrapped in an encasing object, which controls the input and output data flows [15]. This is similar to the intermediate components used by the SimCo simulation tool (see Section IV.B).

The main disadvantage of the interface probing is the problematic generation of the input values. They can be created randomly or manually. In both cases, it is possible to omit values, which are in fact important for the uncovering of hidden features or a hidden behavior of the tested component and its services [13]. In order to reduce the problem, it is possible to use typical values, such as 0, -1, 1, minimal value, maximal value for an integer input or all possible values for a character input [13]. However, these examples are very general and can be useless (in the sense of the uncovering of a hidden behavior) for a specific service. So, it is necessary to use other typical and border input values for the given service, if possible. These values can be provided by a programmer based on textual or other description of the software component and its services, which can be (but does not have to) at the disposal. At the very least, the programmer can use the name of the service as a clue for the service's probable functioning.

C. Other Techniques

Other common techniques include the generation of the graphical user interface for the software component, which enables instant access to the particular services of the component. This graphical user interface then enables quick ad hoc testing of the particular services of the component using various input values without the necessity to create a client application for the component [16].

There are also more exotic approaches such as proposal for the extended component interface specification, which would enable easier black box testing [17] or the utilization of a genetic algorithm for the generation of the input data for the testing [18].

VI. Simulated Components Generation

As it was mentioned before, so far, the simulated components of the SimCo simulation tool are created manually. If the source code of the component, for which the simulated component is created, is known, the simulated component can be created using analysis of this source code. If the source code of the original component is not known (e.g., the component was created by another manufacturer), the situation is even worse, because we have only the public interface of the component and its bytecode (and the Javadoc or other documentation in some cases) at our disposal. In this case, it is difficult to analyze the behavior of the component.

Therefore, we have developed a semi-automated approach for the generation of the simulated components. To avoid confusion, the real component, for which the simulated component (i.e. its simulation model) will be generated, will be referenced only as the *original component* for short from now on.

A. Approach Description

The skeleton of the component (i.e. methods representing services of the component) can be generated easily using the analysis of the public interface of the original component. The basics of the behavior of the simulated component can be then generated using the analysis of the behavior of the original component while it is running among other real components.

The approach does not require the knowledge of the source code of the original component, but there are some limitations. First, in the majority of the cases, the approach is unable to determine the complete behavior of the original component, but can provide clues for this behavior. These clues can be then used by the programmer during the finishing of the simulated component. Thus the "semi-automated" attribute of the approach. Second, it is necessary to have the neighboring components of the original component available. These components require the services or are required by the services of the original component.

B. Case Study: Traffic Crossroad Control

The approach will be demonstrated using a case study – Traffic crossroad control. It represents a component-based application for the control of road traffic in a crossroad using traffic lights. It is expected to run on a specific hardware and operate a variety of hardware sensors and control units [4]. The components involving any direct contact with sensors and control units were replaced by (manually created) simulated components in order to enable running of the application on a standard desktop computer [2].

The Traffic crossroad control application consists of eight components (see Figure 2). The `TrafficCrossroad` component provides the information about the structure of the traffic crossroad. Moreover, its simulated version incorporates a nanoscopic road traffic simulation replacing the real traffic [4]. The `OpticDetection` and the `InductionLoop` components handle specific hardware sensors and provide measured data from these sensors. The simulated versions of these components utilize the data from the nanoscopic road traffic simulation of the `TrafficCrossroad` component. The `TrafficLightsController` component ensures the activation of particular traffic lights. The `ControlPanel` component is a user interface of the entire application. The `TrafficControlAlgorithm` component incorporates an algorithm for the control of the traffic lights. This algorithm can (but does not have to) require data from the sensors mediated by the `SensorAccess` component. This component is also used by the `StatisticsCollector` component, which periodically collects the data from it and provides various traffic statistics accessible via the `ControlPanel` component [4].

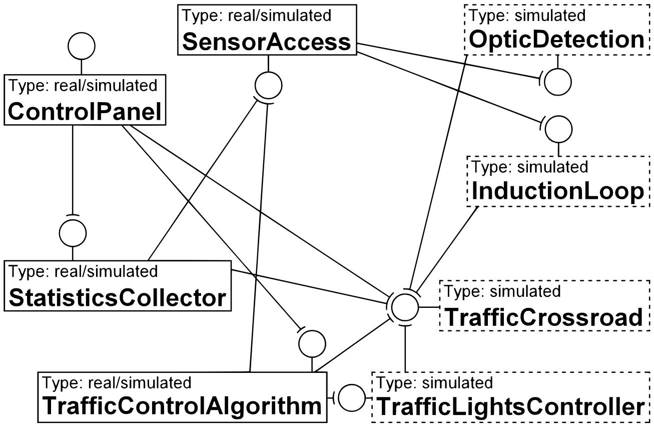


Figure 2. Software components of the Traffic crossroad control application

```
int getQueueLength(String)
boolean isVehicle(String)
DetectorType getCurrentDetectorType()
void setCurrentDetectorType(DetectorType)
```

Figure 3. List of services of the `SensorAccess` component

In the following sections, we will describe our approach for semi-automated generation of a simulated component for the `SensorAccess` component.

C. Generation of Component's Skeleton

The skeleton of the simulated component is generated using the analysis of the public interface of the original component. For this purpose, a service of the OSGi can be used, since it provides functionality to determine the public interface (i.e. a Java interface) of a component (bundle) [19]. For the determination of the particular services (i.e. Java methods) of this interface, the Java reflection can be used [20]. This way, we obtain a list of services provided by the component including the types of their parameters and their return values. The information obtained for the `SensorAccess` component using this approach is depicted in Figure 3.

Using the obtained list of services, the skeleton of the simulated component is generated. The skeleton of each simulated component is represented as one Java class with methods along with their parameters and return values corresponding to the particular services of the original component. The bodies of the generated methods are not completely empty, the logic necessary for adding of the events corresponding to the invocation of the services to the calendar are automatically added. The standard OSGi `Activator` class ensuring the connection of the generated simulated component to the particular components it requires [19] is automatically added to the simulated component as well. The generated skeleton of the simulated component with its services is depicted in Figure 4.

D. Generation of Component's Behavior

The more difficult part of our approach is the generation of the component behavior. For this purpose, the original

```
public class SimulatedSensorAccess implements
    ISensorAccess {
  private BundleContext bc;
  private IOpticDetection od;
  private IInductionLoop il;
  private ICalendar calendar;

  public SimulatedSensorAccess(BundleContext bc,
      IOpticDetection od, IInductionLoop il,
      ICalendar calendar) {
    this.bc = bc;
    this.od = od;
    this.il = id;
    this.calendar = calendar;
  }

  public int getQueueLength(String arg0) {
    calendar.insertEvent();
    return 0;
  }

  public boolean isVehicle(String arg0) {
    calendar.insertEvent();
    return false;
  }

  public DetectorType getCurrentDetectorType() {
    calendar.insertEvent();
    return null;
  }

  public void setCurrentDetectorType(
      DetectorType arg0) {
    calendar.insertEvent();
  }
}
```

Figure 4. Skeleton of the simulated `SensorAccess` component

component, for which the simulated component shall be created, is imported to the SimCo simulation tool along with other components, which are required by the services of the original component or require the services of the original component. It is also possible to import entire component-based application, whose part the original component is. The components, which require the services of the original component, are not necessary, but can provide information about the parameters of the services of the original component.

It is assumed that all the neighboring components of the original components are real, but they can also be simulated, assuming they exhibit the same behavior as the real components they are mimicking. Between all pairs of the neighboring real components, there are intermediate components designed to observe the invocations of the services. The behavior generation then works as follows.

A tree data structure incorporating the original component and all components, which can invoke services of the original component, and all the services of these components is initiated. Each service of each component contains a list of possible invocations with various parameters. These invocations are automatically generated based on the type of their parameters, similarly to [13].

For the `enum` parameters, all possible values (including `null`) are generated, since there are usually only several values. For the `char` parameters, the basic 256 characters and several higher values are generated (in Java, the size of char type is 2 bytes). For the number parameters, a set of representative and border values are generated (e.g., -128, -64, -16, -1, 0, 1, 16, 63, 127 can be generated for a `byte` parameter). It is not feasible to generate all possible values of the number parameters. For object parameters, only the `null` value is generated. If the service has multiple parameters, all combinations are generated. Of course, this can lead to the exponential increase of the number of generated invocations. Hence, the programmer can restrict the generated parameters to values, which he or she considers representative. Moreover, the programmer should provide values, which cannot be automatically generated, for example filled data objects, which are parameters of the invocations. The programmer can use Javadoc of the original component or other documentation for this purpose, if they are at the disposal.

Once the structure is initiated, it is explored component by component, service by service, invocation by invocation. Each invocation from the tree data structure is performed on the corresponding component and it can have various consequences – an exception, a return value, a change of the inner state of the component, and a subsequent invocation of a service of another component. These consequences, with the exception of the change of inner state, which is unobservable from outside, are captured by the intermediate components (and also by the simulated components if they are present) and inserted to the tree data structure, but only if they are not already present in the structure and they are related to the components, which are present in the structure. Simultaneously, the flag indicating that a change of the structure occurred is set. If the invocation consequence is a subsequent invocation, and this invocation is not already present in the structure, the invocation is added to the structure as well and the flag is again set. The exploration of the structure repeats while this flag is set. The pseudocode of the filling and exploring of the tree structure is depicted in Figure 5.

The iterative nature of the approach enables to explore multiple inner states of the components. As it was said before, the change of the inner state of a component is unobservable. However, the inner state of the component can be changed by invocation of its services (if the component can have different inner states at all). On the other hand, the inner state of the component can influence the behavior of its services – a services invoked with the same parameters can have different outcome due to a different inner state of the component. So, the repeating of the invocations, which were already performed, can bring new invocations and invocation consequences, which can be useful for a better exploration of the original component behavior. When no new invocation consequences are generated in the iteration, the process is stopped in order to avoid to be stuck in an infinite loop.

Consider now the Traffic crossroad control case study and the generation of the simulated component for the `SensorAccess` component. This component is then the original component and the `StatisticsCollector` and the `TrafficControlAlgorithm` components are the

```
//Initialization of the tree data structure
structure.addComponent(originalComponent);
structure.addComponents(
  simco.getInvokingComponents(originalComponent));

for (c: structure.components) {
  c.services = simco.getServicesOfComponent(c);
  for (s: c.services) {
    //Generate parameters and read parameters from
    //the user for the invocation
    s.invocations =
      simco.generateAndReadInvocations();
  }
}

//Exploration of invocations and consequences
structure.setChanged(true);
while (structure.isChanged()) {
  structure.setChanged(false);
  for (c: structure.components) {
    for (s: c.services) {
      for (i: s.invocations) {
        invocationConsequences =
          simco.performServiceInvocation(c, s, i);
        for (ic: invocationConsequences) {
          //Only consequences, which are not
          //already present
          if (!i.consequences.contains(ic)) {
            i.consequences.add(ic);
            structure.setChanged(true);
          }
          if (ic.type == SUBSEQUENT_INVOCATION) {
            sc = ic.subsequentComponent;
            ss = ic.subsequentService;
            si = ic.subsequentInvocation;

            //Only for components contained in
            //the structure and only invocations,
            //which are not already present
            if (structure.contains(sc) &&
                !ss.contains(si)) {
              structure.addInvocation(sc, ss, si);
              structure.setChanged(true);
            }
          }
        }
      }
    }
  }
}
```

Figure 5. Pseudocode of the tree data structure exploration

invoking components (see Figure 2). The user did not provide any parameters of the invocations. The filled explored tree data structure is then depicted in Figure 6. There, the components, services, invocations, and invocation consequences are marked with C, S, I, and IC, respectively. The value in the parentheses of I expresses the originator of the invocation – G for “generated at beginning”, U for “provided by user”, and A*x* for “added automatically during the *x*th iteration”. The value in the parentheses of IC denotes the index of the iteration, in which the invocation consequence was added to the tree data structure (starting with 0).

As can be seen in Figure 6, the invoking components add invocations of the services of the original component and provide `String` parameters, which would not be generated

```
C: StatisticsCollector //Invoking component #1
 S: Statistics collectStatistics()
  I(G): collectStatistics()
   IC(0): subsequent invocation
    [SensorAccess.getQueueLength("E_01")]
. . .
   IC(0): subsequent invocation
    [SensorAccess.getQueueLength("E_04")]
   IC(0): return value [statistics]
   IC(1): return value [statistics]

C: TrafficControlAlgorithm //Invoking component #2
 S: void updateTrafficLightsState()
  I(G): updateTrafficLightsState()
    IC(0): subsequent invocation
     [SensorAcces.isVehicle("E_01")]
. . .
    IC(0): subsequent invocation
     [SensorAcces.isVehicle("E_04")]

C: SensorAccess //Original component
 S: int getQueueLength(String)
  I(G): getQueueLength(null)
   IC(0): exception [NullPointerException]
  I(A0): getQueueLength("E_01")
   IC(0): subsequent invocation
    [OpticDetection.getVehiclesCount("E_01")]
   IC(0): return value [2]
   IC(1): subsequent invocation
    [InductionLoop.isVehicle("E_01")]
   IC(1): return value [1]
. . .
  I(A0): getQueueLength("E_04")
   IC(0): subsequent invocation
     [OpticDetection.getVehiclesCount("E_04")]
   IC(0): return value [0]
   IC(1): subsequent invocation
     [InductionLoop.isVehicle("E_04")]
   IC(1): return value [0]
 S: boolean isVehicle(String)
  I(G): isVehicle(null)
   IC(0): exception [NullPointerException]
  I(A0): isVehicle("E_01")
   IC(0): subseqeuent invocation
    [OpticDetection.getVehiclesCount("E_01")]
   IC(0): return value [true]
   IC(1): subseqeuent invocation
    [InductionLoop.isVehicle("E_01")]
   IC(1): return value [true]
. . .
  I(A0): isVehicle("E_04")
   IC(0): subseqeuent invocation
    [OpticDetection.getVehiclesCount("E_04")]
   IC(0): return value [false]
   IC(1): subseqeuent invocation
    [InductionLoop.isVehicle("E_04")]
   IC(1): return value [false]
 S: DetectorTypes getDetectorType()
  I(G): getDetectorType()
   IC(0): return value [DetectorTypes.OPTIC]
   IC(1): return value [DetectorTypes.INDUCTION]
 S: void setDetectorType(DetectorTypes)
  I(G): setDetectorType(null)
   IC(0): exception [NullPointerException]
  I(G): setDetectorType(DetectorTypes.OPTIC)
   IC(0): nothing observable
  I(G): setDetectorType(DetectorTypes.INDUCTION)
   IC(0): nothing observable
```

Figure 6. Example of the filled explored tree data structure

if the invoking components were not present. These parameters (e.g., “E_01”) are the IDs of the traffic lanes, in which the sensors survey the state of road traffic. In this case, the IDs are taken from the TrafficCrossroad component, which provides information about the traffic crossroad, by the StatisticsCollector and TrafficControlAlgorithm components. The invocations of the services of the TrafficCrossroad component are not present in the filled tree data structure, because the TrafficCrossroad is not present in the tree data structure, since it is not an invoking component of the original (i.e. SensorAccess) component. We know the origin of the IDs, only because we developed the Traffic crossroad control application. If this were not the case and the source code of the application were not available to us, the origin of the IDs would be hidden from us. Nevertheless, the IDs would still be there helping to probe the behavior of the original SensorAccess component.

The overall statistics of the content of the filled tree data structure is summarized in Table I. The tree data structure is filled in three iterations of the outermost (while) cycle (see Figure 5). In the third iteration, there are no newly added invocations or invocation consequences. So, the algorithm ends.

TABLE I. OVERALL STATISTICS OF THE FILLED TREE DATA STRUCTURE

Feature	Count
Number of invocations generated at beginning	8
Number of invocations provided by user	0
Number of automatically added invocations in 1st iteration (A0)	8
Number of automatically added invocations in 2nd iteration (A1)	0
Number of automatically added invocations in 3rd iteration (A2)	0
Number of generated consequences in 1st iteration (0)	31
Number of generated consequences in 2nd iteration (1)	17
Number of generated consequences in 3rd iteration (2)	0
Number of iterations	3

Using the part of the filled explored tree data structure containing data for the SensorAccess component and a set of rules, it is possible to generate limited bodies of the services of this component with clues for the programmer. The rules are following:

1. If there is a single return value for each invocation, generate an if-else construction depending on the parameters of the invocation for the particular return values of the service.
2. If there are multiple return values for a single invocation, generate an if-else construction depending on the parameters of the invocation for the return value of the service. All but the first return value for the same invocation put in the comments in the corresponding branch of the if-else construction.
3. If there is a single return value for a service without parameters, generate return statement with this value.
4. If there are multiple (different) return values for a service without parameters, this service can reflect state of the component (e.g., it is a getter). Generate an information comment for the programmer. Generate multiple return statements with all return values. All but the first return statements put in the comments.
5. If there is a single subsequent invocation for each invocation and the parameters of the invocation correspond by type and value to the parameters of the subsequent invocation, generate a method call corresponding to the subsequent invocation using the formal parameters of the service.
6. If there are multiple subsequent invocations of different services for a single invocation and the parameters of the invocation correspond by type and value to the parameters of the subsequent invocations, generate multiple methods calls corresponding to the particular subsequent invocations using the formal parameters of the service.
7. If there is a single subsequent invocation for each invocation and the parameters of the invocation do not correspond by type or value to the parameters of the subsequent invocation, generate an if-else construction depending on the parameters of the invocation for the particular methods calls corresponding to the subsequent invocations.
8. If there are multiple subsequent invocations of different services for a single invocation and the parameters of the invocation do not correspond by type or value to the parameters of the subsequent invocations, generate an if-else construction depending on the parameters of the invocation for the particular sets of the methods calls corresponding to the subsequent invocations.
9. If there are multiple (different) subsequent invocations for a single invocation in multiple iterations and the parameters of the invocation correspond by type and value to the parameters of the subsequent invocations, the subsequent invocation can depend on the inner state of the component. Generate an information comment for the programmer. Generate an if-else construction depending on the parameters of the invocation for the methods calls corresponding to the particular subsequent invocations. All but the first method call for the same invocation put in the comments in the corresponding branch of the if-else construction.
10. If there are multiple (different) subsequent invocations for a single invocation in multiple iterations and the parameters of the invocation do not correspond by type or value to the parameters of the subsequent invocations, the subsequent invocation can depend on the inner state of the component. Generate an information comment for the programmer. Generate multiple methods calls corresponding to the particular subsequent invocations. All but the first method call put in the comments.
11. If there is a single exception for a single invocation, generate an if-else construction depending on the parameters of the invocation for the throwing of the corresponding exceptions.

12. If there are multiple (different) exceptions for a single invocation, generate `if-else` construction depending on the parameters of the invocation for the throwing of the corresponding exception. Additional exceptions for the same invocation put in comments in the corresponding branch of the `if-else` construction.
13. If there are multiple exceptions or a single exception for a service without parameters, generate the throwing of the corresponding exceptions. Put them into the comments.

The resulting generated bodies of the services of the simulated `SensorAccess` component are depicted in Figure 7. The handling of the service using the calendar was added already during the component skeleton generation (see Section VI.C). The particular parts of the generated bodies – invocation of services, return values, throwing of exceptions – are not directly usable. It is possible to compile the generated simulated component, but its behavior will not correspond to the behavior of the original component. However, the programmer can use the generated parts as clues for what the particular services should do. It is much easier for him or her to finish the functionality of the simulated component than when he or she should program the functionality of the simulated components from the scratch. This is the most important contribution of our semi-automated approach.

In Figure 7, it is indicated with the comments in the generated source code, which parts of the services bodies were generated using which rule. For the entire `SensorAccess` component, only the rules 1, 2, 4, 9, and 11 were utilized.

E. Approach Evaluation

As it was demonstrated using the case study, the automatic part of our approach for the generation of the simulated components is working and provides clues of what the particular services of the component should do for the programmer. The programmer then finishes the functionality of the simulated component using these clues.

However, it should be noted that in a more complicated case with services with multiple parameters, the generated bodies of the services can become quite long and difficultly understandable by the programmer. In such cases, the restrictions of the generated parameters provided by the programmer are important (see Section VI.D). This restriction was not used in the presented case study, since the number of parameters of the services was low and the generated bodies of the services are easily readable.

The programmer also did not provide any parameters values for the services invocations. Despite of this, the invoking software components provided the `String` parameters, which enable a better exploration of the behavior of the services of the original component (see Section VI.D). Nevertheless, this depends solely on the nature of the component-based application. In many cases, the invoking components do not have to provide useful parameters. Then, the possibility to add the parameters of the services invocations by the programmer becomes more important.

```
public int getQueueLength(String arg0) {
  calendar.insertEvent();

  //Rule 11
  if (arg0 == null) {
    throw new NullPointerException();
  }

  //Rule 9
  if (arg0.equals("E_01"))
    od.getVehiclesCount("E_01");
    //il.isVehicle("E_01");
. . .
  else if (arg0.equals("E_04"))
    od.getVehiclesCount("E_04");
    //il.isVehicle("E_04");

  //Rule 2
  if (arg0.equals("E_01"))
    return 2;
    //return 1;
. . .
  else if (arg0.equals("E_04"))
    return 0;
  else
    return 0;
}

public boolean isVehicle(String arg0) {
  calendar.insertEvent();

  //Rule 11
  if (arg0 == null) {
    throw new NullPointerException();
  }

  //Rule 9
  if (arg0.equals("E_01"))
    od.getVehiclesCount("E_01");
    //il.isVehicle("E_01");
. . .
  else if (arg0.equals("E_04"))
    od.getVehiclesCount("E_04");
    //il.isVehicle("E_04");

  //Rule 1
  if (arg0.equals("E_01"))
    return true;
. . .
  else if (arg0.equals("E_04"))
    return false;
  else
    return false;
}

public DetectorType getCurrentDetectorType() {
  calendar.insertEvent();

  //Rule 4
  return DetectorTypes.OPTIC;
  //return DetectorTypes.INDUCTION;
}

public void setCurrentDetectorType(DetectorType
arg0) {
  calendar.insertEvent();

  //Rule 11
  if (arg0 == null)
    throw new NullPointerException();
}
```

Figure 7. Generated services bodies of the SensorAccess component

VII. Future Work

In our future work, we will focus on improving our approach in several ways.

A. *Better Exploration of the Tree Data Structure*

First of all, we will focus on a better exploration of the tree data structure, from which the simulated components are generated. This includes ordering of the particular invocations in order to explore more different inner states of the components. This can theoretically lead to acquiring of a higher number of subsequent invocations.

We also want to employ analysis of the names of the services/methods. This can be useful for example for determining whether a pair of services serves as a getter and a setter of a property of the component.

B. *Expansion of the Set of Rules*

The rules described in Section VI.D can be also expanded by other rules in order to provide more and/or better clues in the generated bodies of the services of the simulated component. The generation of the clues can be also refined to be clearer for the programmer. For example, an array or a table with values accessed using indices can be used instead of large `if-else` constructions.

VIII. Conclusion

In this paper, we described an approach for the semi-automated generation of a simulated component, which is used as a replacement of a real component during simulation testing of other real software components. The approach is based on the observation of the original component behavior while it is interacting with other real components, which require or are required by its services.

The resulting generated simulated component incorporates all services of the original component and parts of their bodies, which provide clues for the programmer. The programmer can then use these clues for the finishing of the behavior of the services of the simulated component. The approach was designed for the OSGi component model, but the ideas behind it can be utilized also in other component models. The approach is demonstrated using a case study.

Acknowledgment

This work was supported by the European Regional Development Fund (ERDF), project "NTIS – New Technologies for the Information Society", European Centre of Excellence, CZ.1.05/1.1.00/02.0090.

References

[1] C. Szyperski, D. Gruntz, and S. Murer, Component Software – Beyond Object-Oriented Programming, ACM Press, New York, 2000.

[2] T. Potuzak and R. Lipka, "Possibilities of Semi-automated Generation of Scenarios for Simulation Testing of Software Components," International Journal of Information and Computer Science, vol. 2(6), September 2013, pp. 95–105.

[3] T. Potuzak, R. Lipka, J. Snajberk, P. Brada, and P. Herout, "Design of a Component-based Simulation Framework for Component Testing using SpringDM," ECBS-EERC 2011 – 2011 Second Eastern European Regional Conference on the Engineering on Computer Based Systems, Bratislava, September 2011, pp. 167–168.

[4] T. Potuzak, R. Lipka, P. Brada, and P. Herout, "Testing a Component-based Application for Road Traffic Crossroad Control using the SimCo Simulation Framework," 38th Euromicro Conference on Software Engineering and Advanced Applications, Cesme, Izmir, September 2012, pp. 175–182.

[5] R. Lipka, T. Potuzak, P. Brada, and P. Herout, "Verification of SimCo – Simulation Tool for Testing of Component-based Application," EUROCON 2013, Zagreb, July 2013, pp. 467–474.

[6] The OSGi Alliance, OSGi Service Platform Core Specification. Release 4, version 4.2, 2009.

[7] D. Rubio, Pro Spring Dynamic Modules for OSGi^{TM} Service Platform, Apress, USA, 2009.

[8] R. M. Fujimoto, Parallel and Distributed Simulation Systems, John Wiley & Sons, New York, 2000.

[9] S. Becker, H. Koziolek, and R. Reussner, "The Palladio component model for model-driven performance prediction," Journal of Systems and Software, vol. 82(1), 2009, pp. 3–22.

[10] P. G. Sapna and H. Mohanty, "Automated Scenario Generation based on UML Activity Diagrams," International Conference on Information Technology 2008, December 2008, pp. 209–214.

[11] J. McAffer, P. VanderLei., and S. Archer, OSGi and Equinox: Creating Highly Modular Java^{TM} Systems, Pearson Education Inc., Boston, 2010.

[12] G. Canfora and M. Di Penta, "New Frontiers of Reverse Engineering," Future of Software Engineering (FOSE'07), Minneapolis, 2007, pp. 326–341.

[13] B. Korel, "Black-Box Understanding of COTS Components," Seventh International Workshop on Program Comprehension, Pittsburgh, 1999, pp. 92–99.

[14] S. Liu and W. Shen, "A Formal Approach to Testing Programs in Practice," 2012 International Conference on Systems and Informatics, Yantai, 2012, pp. 2509–2515.

[15] J. M. Haddox, G. M. Kapfhammer, and C. C. Michael, "An Approach for Understanding and Testing Third Party Software Components," Proceedings of Annual Reliability and Maintainability Symposium, Seattle, 2002, pp. 293–299.

[16] F. Naseer, S. U. Rehman, and K. Hussain, "Using Meta-data Technique for Component Based Black Box Testing," 2010 6th International Conference on Emerging Technologies, Islamabad, 2010, pp. 276–281.

[17] J. Ying, Y.-N. Li, X.-D. Fu, "The Support of Interface Specifications in Black-box Components Testing," 2010 Fifth International Conference on Frontier of Computer Science and Technology, Changchun, 2010, pp. 305–311.

[18] M. Fisher and R. Tönjes, "Generating Test Data for Black-Box Testing using Genetic Algorithms," 2012 IEEE 17th Conference on Emerging Technologies & Factory Automation (ETFA), Krakow, 2012, pp. 1–6.

[19] N. Bartlell, OSGi in Practice, eBook (Creative Commons), 2009.

[20] I. R. Forman and N. Forman, Java Reflection in Action. Manning Publications Co., Greenwich 2005.

Energy and Daylighting Performance of Senior Housing

Performance evaluation of a senior apartment in China

Yuan Fang and Soolyeon Cho
College of Design
North Carolina State University
Raleigh, USA
yfang4@ncsu.edu, soolyeon_cho@ncsu.edu

***Abstract*—Senior living facilities consume more energy than other types of residential buildings due to seniors' different lifestyles and comfort levels. In order to reduce the high energy consumption of senior living facilities, designers should consider sustainable design strategies. One important strategy is daylighting because of its potential to save energy and to benefit seniors' health. This paper presents a case study based on a senior apartment in Changsha, China. Simulation models are developed to analyze the building's energy consumption and daylighting conditions. Simulation input parameters are adopted through a literature survey to accommodate the real lifestyles and thermal comfort conditions of seniors. Daylight levels are tested, analyzed and considered in relation to energy performance. This paper shows that daylighting is a more efficient energy saving measure for senior living facilities than it is for regular residential buildings.**

Keywords-senior housing; thermal comfort; visual comfort; energy saving; daylighting.

I. Introduction

The elderly population is growing dramatically worldwide, hence the increasing demand for senior housing and senior-care facilities. The population of the elderly—people aged 60 years or over—is predicted to more than double between 2013 and 2050. Also, the proportion of aged people will increase from 12 percent in 2013 to 21 percent in 2050. The population is aging more quickly in developing countries than in developed countries, and China has one of the fastest-aging populations in the world [1].

As people age, their physical fitness and physiological functions degrade. In order to maintain seniors' comfort levels, special built environments need to be developed, specifically thermal [2], lighting [3], and acoustic environments [4]. However, most building standards and codes that currently govern the physical built environment are based on the needs of younger adults and do not specifically address seniors' requirements.

Senior-care facilities have nearly two times the Energy Use Intensity (EUI) of normal residential buildings [5]. The reason for the enormous EUI should be investigated in order to achieve the energy saving goals in senior housing.

This research studies aspects of a specially built environment, such as room temperature and lighting level, which affect the ability of seniors to maintain their health and comfort. The energy consumption features of senior housing are examined to find how they are different from those of housing for younger adults. The importance of daylighting for seniors and senior housing are proposed. The energy savings though daylighting are assessed through a modified simulation process using the OpenStudio [6], Radiance [7], and EnergyPlus [8] simulation programs.

The second section of the paper reviews the literature about the input values for simulation, which are the temperature and illumination for seniors' comfort. The third section describes the methodology of the research, including the simulation model, simulation tools, and simulation process. The fourth section presents the simulation of four energy models, the fifth section analyzes the simulation results, and the sixth section shows the conclusion of the research.

II. Literature Review

A. Temperature and Thermal Comfort

Thermal comfort is one of the most important criteria to consider when evaluating a built environment. The temperature setpoint has a huge impact on the building's energy consumption. According to current standards, the comfort temperature requirements are the same for people of all ages. However, researchers have found that the elderly have different thermal environment preferences from younger adults.

In a climate chamber research, it was found that older adults prefer a higher ambient temperature, 23.1 °C (73.5 °F), than do younger adults, 21.9 °C (71.4 °F), for a difference of about 1.2 °C (2.0 °F) [2]. It has also been found that the elderly prefer a higher temperature than do younger adults [9]. A field study about seniors' thermal comfort requirements in residential environments showed that seniors' temperature of thermal neutrality was 25.2 °C (77.4 °F) for the summer and 23.2 °C (73.8 °F) for the winter [10]. However, other studies found different results that older adults preferred the same environmental temperature as the younger adults and that the optimal temperature for thermal comfort was 21.1 °C (70.0 °F) for both age groups [11].

This study is based on the assumption that seniors prefer relatively high temperatures. People's temperature preferences tend to vary according to climate and culture, so data on the specific thermal condition preference of the seniors in China would be beneficial to future studies. In this study, the ASHRAE 90.1-2004 standard is used for the baseline simulations. In ASHRAE 90.1-2004, the heating and cooling temperature settings for mid-rise apartment are 21.1 °C (70.0 °F) and 23.9 °C (75.0 °F), respectively. In this study, higher temperature settings are used for the simulation models of senior housing; the heating and cooling temperatures are changed to 22.2 °C (72.0 °F) and 25.0 °C (77.0 °F), respectively.

B. *Illumination and Visual Comfort*

Lighting energy accounts for an important proportion of building energy consumption. ASHRAE standards normally set the lighting energy based on the Illuminating Engineering Society's (IES) lighting level recommendations for normally-sighted people.

However, because of changes in the human eye lens and the visual nervous system over time, the amount of effective light reaching the retina declines as a person ages. Seniors require about two to three times the light of younger adults to have proper visual acuity and to see comfortably [3]. Therefore, the lighting power density requirements of ASHRAE standards are not adequate for providing sufficient lighting for seniors. Insufficient lighting can cause falling, collision, vision loss, and other problems.

Recently, the IES has developed special lighting requirements for seniors. Table I shows the comparison of the illuminance requirements for senior housing versus the requirements for offices and normal residence environments. Senior housing needs about the same lighting level as offices, and the general lighting requirement is 4 times the illuminance of normal residences.

In this study, a simple approach is used for determining the lighting power density of senior housing. In ASHRAE 90.1-2004, the lighting power density is 3.88 W/m^2 (0.36 W/ft^2) for mid-rise apartments and 10.76 W/m^2 (1 W/ft^2) for offices. Because the illuminance requirement of senior housing is similar to that of an office, the lighting power density for senior housing is set to 10.76 W/m^2 (1 W/ft^2). Only general lighting is considered in this study; task lighting such as dining and reading lighting are not considered.

C. *Daylighting and Health*

Seniors tend to stay in their homes for more of the day than younger adults, and they need more light to see clearly, so daylighting is a more suitable energy-saving measure for senior housing than it is for other residential buildings. Daylighting also offers great health benefits to seniors.

TABLE I. COMPARISON OF ILLUMINANCE LEVEL [3][12]

Area	**Senior housing [Lux (FC)]**	**Office [Lux (FC)]**	**Residence [Lux (FC)]**
General lighting	300 (30)	300 (30)	75 (7.5)
Dining/Cafeteria	300 (30)	300 (30)	150 (15)
Bathrooms	300 (30)	150 (15)	150 (15)
Grooming	600 (60)	NA	300 (30)
Corridors	300 (30)	150 (15)	75 (7.5)

Daylight exposure can stimulate the production of vitamin D under the skin, improve the health of bones and muscles, and prevent seniors from falling and suffering fractures [3]. As people age, the ability of their skin to produce vitamin D decreases, and they tend to spend less time outdoors [3]. Another possible effect of insufficient or improper lighting is circadian dysfunction, which can lead to Seasonal Affective Disorder (SAD) and Sick Building Syndrome (SBS) [13]. Therefore, sufficient natural light in the rooms is especially important for senior housing. Daylighting design should be an important consideration in senior housing.

III. METHODOLOGY

A. *Research Framework*

The study includes the simulation of four energy models with the same building geometry. Figure 1 shows the framework of the research. The first energy model is the baseline model for a regular mid-rise apartment. The second energy model is the baseline model for a senior apartment. The comparison of the first and second models shows how a regular apartment and a senior apartment have different energy performance. The third energy model is the regular apartment with daylighting strategies, and the fourth energy model is the senior apartment with the same daylighting strategies. The daylighting conditions of the models are evaluated, and the energy saving through daylighting is found by comparing the four models. The difference in the energy saving shows how daylighting performs differently in regular housing as compared to senior housing.

B. *Simulation Tools and Process*

The OpenStudio program, developed by the National Renewable Energy Laboratory (NREL), is used as the primary simulation platform. OpenStudio runs as a plug-in for the SketchUp program that develops three-dimensional geometries. As a graphic user interface, OpenStudio uses the EnergyPlus program, developed by the US Department of Energy, to run thermal simulations, and the Radiance program, developed by the Lawrence Berkeley National Laboratory, to run daylight performance simulations.

Typically, the daylighting simulation workflow in OpenStudio is to first export the building geometry and model data into a rad file for simulation in Radiance. A lighting schedule is generated automatically after the daylighting simulation and is embedded into the OpenStudio

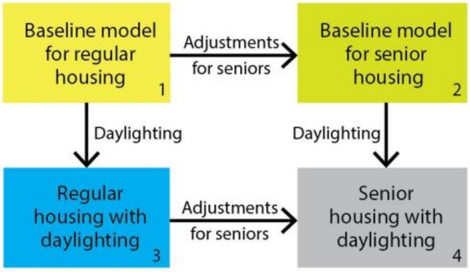

Baseline model for regular housing
1
Adjustments for seniors
Baseline model for senior housing
2
Daylighting
Daylighting
Regular housing with daylighting
3
Adjustments for seniors
Senior housing with daylighting
4

Figure 1. Research Framework

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-371-1

file for final whole-building energy simulation [6]. However, this typical process is not appropriate for the present study. No input adjustments of the Radiance simulation process are currently allowed in OpenStudio. The different illuminance targets for senior housing and regular housing cannot be demonstrated, and corresponding lighting schedules cannot be generated, so the simulation results would not be accurate. Thus, for this study, a modified process is developed to address this issue.

Figure 2 shows the tools and the process for this simulation methodology. After the regular building energy simulation in OpenStudio, the annual daylight illuminance data from the ill file simulated by Radiance are located, and analyzed in Excel. The average hourly monthly illuminance tables for each space are generated for daylighting condition evaluation and lighting schedule calculation. Figure 3 shows one example of the illuminance table, for a room with the illuminance target of 300 Lux. Each cell shows the average illuminance of a room at one hour in a month. Figure 4 shows the lighting schedule generated for the same room. The lighting schedule of each hour is determined by how much more illuminance is needed to reach the illuminance target and what percentage of the target it accounts for. The hours considered for daylighting are 7:00 to 17:00. If the average illuminance of an hour exceeds the illuminance target, the light will be set to 10% on for a conservative consideration. Then, the new schedules need to be synthesized with the existing lighting schedules and manually input into OpenStudio for the final simulation. The third and fourth models are simulated using this method. The cooling, heating, equipment, lighting, and total energy consumption of the four models are the final results needed.

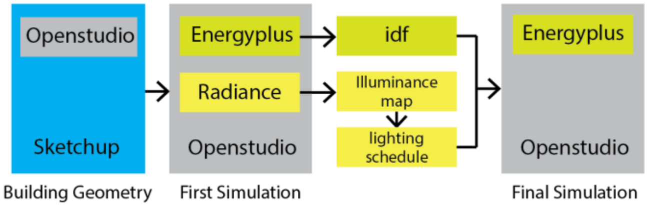
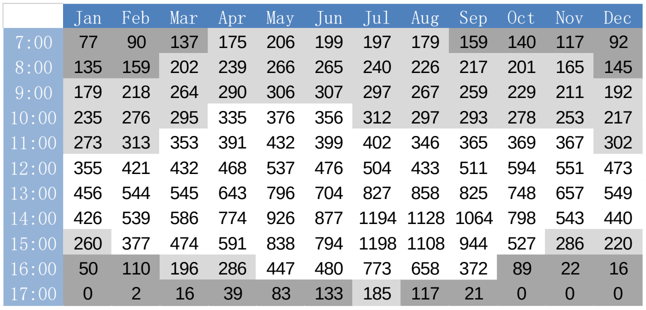
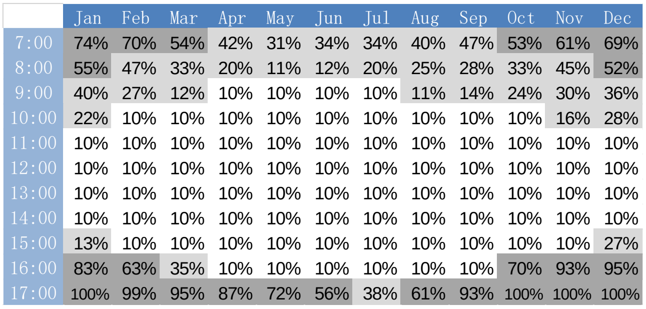


Figure 2. Simulation tools and process



	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
7:00	77	90	137	175	206	199	197	179	159	140	117	92
8:00	135	159	202	239	266	265	240	226	217	201	165	145
9:00	179	218	264	290	306	307	297	267	259	229	211	192
10:00	235	276	295	335	376	356	312	297	293	278	253	217
11:00	273	313	353	391	432	399	402	346	365	369	367	302
12:00	355	421	432	468	537	476	504	433	511	594	551	473
13:00	456	544	545	643	796	704	827	858	825	748	657	549
14:00	426	539	586	774	926	877	1194	1128	1064	798	543	440
15:00	260	377	474	591	838	794	1198	1108	944	527	286	220
16:00	50	110	196	286	447	480	773	658	372	89	22	16
17:00	0	2	16	39	83	133	185	117	21	0	0	0

Figure 3. Monthly hourly illuminance of one room (Lux)



	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
7:00	74%	70%	54%	42%	31%	34%	34%	40%	47%	53%	61%	69%
8:00	55%	47%	33%	20%	11%	12%	20%	25%	28%	33%	45%	52%
9:00	40%	27%	12%	10%	10%	10%	10%	11%	14%	24%	30%	36%
10:00	22%	10%	10%	10%	10%	10%	10%	10%	10%	10%	16%	28%
11:00	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
12:00	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
13:00	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
14:00	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
15:00	13%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	27%
16:00	83%	63%	35%	10%	10%	10%	10%	10%	10%	70%	93%	95%
17:00	100%	99%	95%	87%	72%	56%	38%	61%	93%	100%	100%	100%

Figure 4. Lighting schedule of one room

IV. SIMULATION

A. Simulation Model Introducation

The simulation model is a senior apartment located in Changsha, south-central China. Changsha is at the longitude 112 °E and the latitude 28 °N. It has a humid, subtropical climate [14]. According to the Köppen-Geiger classification, this climate is the same as that of the southeastern US; thus, the ASHRAE standard for climate zone 4 is used.

This apartment, which has seven floors, is one of the most common types of housing in China. Figure 5 shows the model of the building. This building is an independent living apartment. The first floor is the public area, and the second through seventh floors are the living area. Figure 6 shows the floor plan. There are 28 rooms on each floor, and there are five major types of rooms: west-facing rooms, recessed south-facing rooms, south-facing rooms with balconies, south-facing rooms, and east-facing rooms. In this study, one room of each type is selected for simulation. Table II shows the characteristics of the five rooms. Ideal loads air system is used for simulation.

B. First Energy Model

The first energy model is the baseline model for a regular apartment. It uses the ASHERAE 90.1-2004, midrise apartment, climate zone 4, as the template. To simplify the

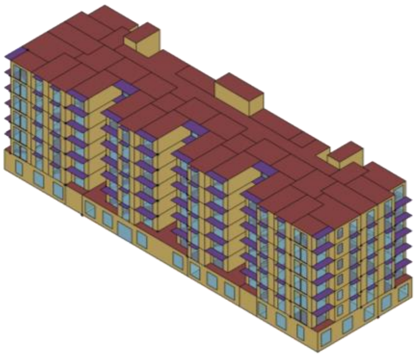


Figure 5. The building model

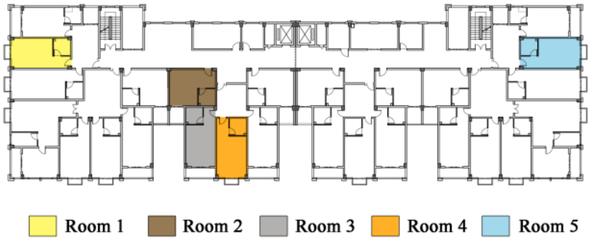


Figure 6. Floor plan and room selection

TABLE II. CHARACTERISTICS OF ROOMS

ROOM	Description	Room dimension (width*depth*height) [m]	Area [m^2]	People / room	Gross Wall Area [m^2]	Window area [m^2]	Window to floor ratio	Balcony dimension (Width*depth) [m]
1	West facing	4.20*8.20*3.25	34.44	2	13.66	4.86	14.1%	2.00*0.90
2	South facing and recessed	6.40*5.10*3.25	32.64	2	7.15	4.86	14.9%	2.20*2.50
3	South facing with balcony	4.20*8.20*3.25	34.44	2	40.33	4.86	14.1%	4.20*1.50
4	South facing	4.20*8.20*3.25	34.44	2	23.42	4.86	14.1%	2.00*0.90
5	East	4.20*8.20*3.25	34.44	2	13.66	4.86	14.1%	2.00*0.90

simulation, only the five rooms are modeled in detail, and each of the five rooms is assigned to an individual thermal zone.

C. Second Energy Model

The second energy model is the adjusted baseline model used to demonstrate the energy consumption of senior housing. To meet the special lighting and thermal levels for seniors' comfort and health, the lighting power density and temperature setpoint are adjusted. To represent a more accurate living pattern and the occupation status of seniors, the schedules are changed accordingly.

1) Lighting power density:

As stated earlier in the literature review, the lighting power density is changed from 3.88 W/m^2 (0.36 W/ft^2) to 10.76 W/m^2 (1 W/ft^2) to meet the lighting level for seniors' visual comfort.

2) Temperature setpoint:

The heating temperature setpoint is changed from 21.1 °C (70.0 °F) to 22.2 °C (72.0 °F). The cooling temperature setpoint is changed from 23.9 °C (75.0 °F) to 25.0 (77.0 °F).

3) Schedule:

As distinct from regular residential buildings, senior living buildings are occupied more hours of the day, sometimes even 24 hours a day. In this study, data from the American Time Use Survey (ATUS) [15] are used to determine the daily activity of the elderly, and the average lighting, equipment, and occupancy schedules are generated. In future studies, more accurate schedules for the seniors in China need to be developed. In normal mid-rise apartments during the day, the lighting and occupancy schedule can be as low as 10% to 30% occupied, and the equipment schedule can be about 60% to 70% on. For this study, the lighting and equipment schedules for senior housing are changed to 100% on during the day. The occupancy schedule is changed to 80% to 90% occupied during the day.

D. Daylighting Simulation

For the daylighting simulation, using the OpenStudio plug-in in SketchUp, 5×5 illumination maps of 25 sensor points are placed into the five rooms, and the daylighting controls are placed in the center of the rooms. They are all at the height of 762 mm (2.5 ft). Simple glazing system windows is used in the model. The U-factor of the window is 2.95 W/m^2·K (0.52 Btu/ft^2·h·R). The Solar Heat Gain Coefficient is 0.39. The visible transmittance is 0.27. The annual daylight illuminance data are simulated so that they could be used for the lighting schedule and daylighting condition analysis. Because the building geometry and material data are the same for the third and fourth model, the daylighting simulation results are applied to both of them.

E. Third Energy Model

The third model is the regular apartment with daylighting. The building geometry and all the building data, except the lighting schedule, are the same as for the first model. 75 Lux (7.5 FC) is used as the target illuminance, and lighting schedules are generated using the data from daylighting simulation.

F. Fourth Energy Model

The fourth model is the senior apartment with daylighting. Lighting schedules are generated based on the target of 300 Lux (30 FC). All the other model data are the same as for the second energy model.

V. RESULTS AND ANALYSIS

A. Daylighting Simulation Results

Figure 7 shows the annual average illuminance and average peak illuminance of the five rooms. The red line is the average illuminance target for senior housing, which is 300 Lux (30 FC), as required by IES. The yellow line is the average illuminance target for regular housing. An optimal design should have the average illuminance above the goal without high peak illuminance.

The glazing material has a visible transmittance of only 0.27, which is low for daylighting purpose. Therefore, the average illuminance for all the rooms is acceptable for normal housing, but insufficient for senior housing. The south room without balcony has the highest average illuminance, and the west room has the highest peak illuminance.

In Figure 8, the blue line shows the percentage of time that a room's illuminance reaches the target of 300 Lux. The red line shows the percentage of time the room illuminance exceeds double the target, in which case the light might cause too much heat gain or a glare, and the occupants might close the window blind. The green line shows the percentage of time that the room illuminance is between 300 and 600 Lux, which is the appropriate level of light for seniors. The lighting conditions for all the rooms need improvement because more than half of the time the illuminance cannot reach the target. The daylighting performance of the recessed

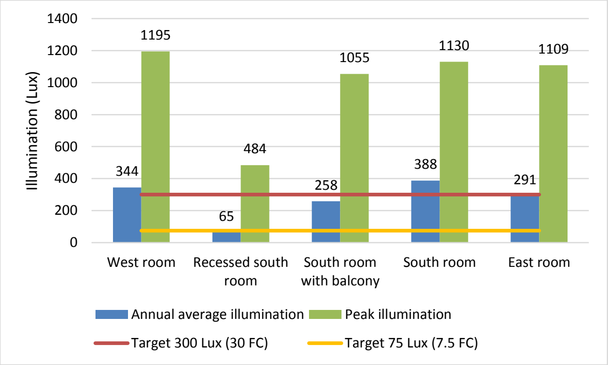


Figure 7. Annual average illuminance of 5 rooms

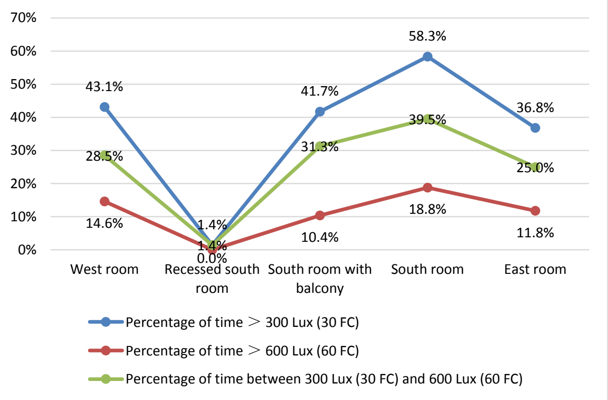


Figure 8. Percentage of time reaching target

south room is far from ideal, and this type of room should be avoided in senior housing design. The other four room types all have the potential for improvement.

Daylighting conditions of this apartment could be improved through design decisions such as adjusting the size and position of windows, increasing the transmittance of glazing material, changing the reflectance of interior material, and adding proper shading devices.

B. Energy Simulation Results

Figure 9 shows the energy simulation results of the four models. The blue dots are the total EUI, and the clustered columns show the EUI breakdown. Figure 9 (a) shows the EUI of the baseline model for regular housing. Of the five selected rooms, the recessed south room has the lowest total EUI, and the south room with balcony has the highest EUI. These results are explained by the fact that these two rooms have the least and greatest exterior wall area. Cooling energy is the largest component of energy consumption. Equipment energy is the second largest, and lighting and heating energy only account for a small proportion of total energy.

Figure 9 (b) shows the EUI of baseline model for senior housing. Compared to the first energy model, the average EUI increases 2.33 times, which matches the previous findings [5]. The south room has the highest EUI, but the total EUI difference among the five rooms is not obvious. The lighting energy increases 5.40 times, and the cooling energy increases 2.57 times. Even though the temperature setpoint for senior housing is higher, the cooling energy is even larger, and the heating energy is even smaller. The main reasons for this result are the extra heat gain from the increased lighting and the greater number of occupied hours.

Figure 9 (c) shows the EUI of the regular housing with daylighting. The average energy saving is 1.07%. The average lighting energy saving is 6.57%, and the average cooling energy saving is 0.98%. Because the rooms are not occupied for most of the daylighting hours, the lighting energy saving is not obvious. There is a slight decrease in cooling energy and increase in heating energy because of the lower amount of heat generated by the lower level of electrical lighting.

Figure 9 (d) shows the EUI of the senior housing with daylighting. The average energy saving is 16.75%. The average lighting energy saving is 32.39%, and the average cooling energy saving is 14.16%. Even without any design improvements, there is significant decrease in lighting and cooling energy compared to the energy savings in regular apartment. Therefore, daylighting is a more efficient energy savings method for senior housing. Design optimizations will be experimented in future studies to further improve the energy performance. Temperature settings can also influence the energy consumption greatly. Due to the high cooling energy and low heating energy in senior housing, higher temperature setpoints could be considered so long as a comfortable thermal environment is maintained. The EUI of the recessed south room is much higher than the other four rooms, due to its limited access to daylight.

VI. CONCLUSION

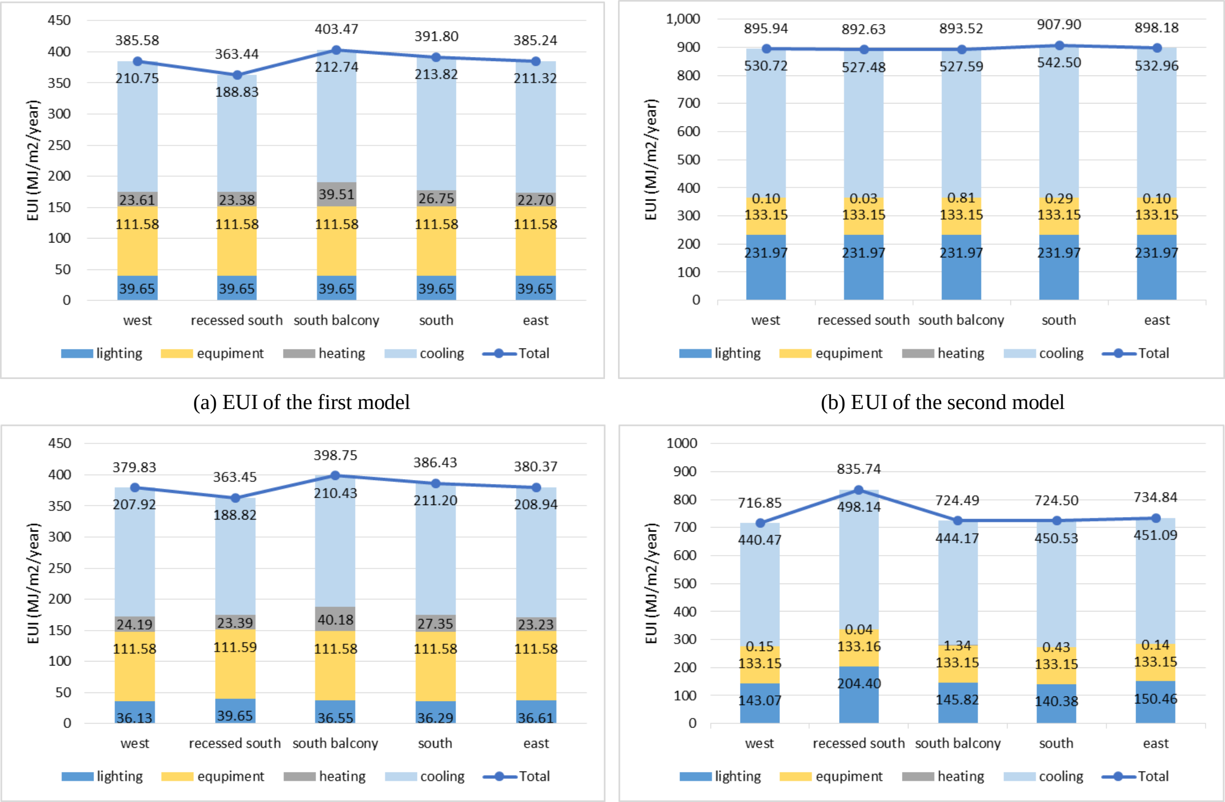
This paper presented an approach for modeling the energy performance of senior housing and the energy savings through daylighting.

The energy consumption of senior housing is more than two times higher than that of normal housing, and the energy increase is mostly from cooling and lighting energy. Daylighting is a suitable energy savings strategy for senior housing. The natural light in the case study building is inadequate and the daylighting condition needs to be optimized, but the energy saving is obvious even in current condition. Daylighting can reduce the building’s lighting and cooling energy, and it works more effectively for senior housing than for regular residential buildings.

The energy performance of the five rooms in this building varied greatly when simulated as regular housing or senior housing. These results indicate that senior housing should be specially designed to achieve the optimal energy and daylighting performance.

ACKNOWLEDGMENT

A special thank you goes to Innovative Design that has provided technical comments and drawings of the case study building for the daylighting simulation and analysis. This study is funded by the key R&D project (No. 2014-0035) of Korea Institute of Civil Engineering and Building Technology (KICT).



(a) EUI of the first model

(b) EUI of the second model

(c) EUI of the third model

(d) EUI of the fourth model

Figure 9. EUI of the four models

REFERENCES

[1] United Nations, Department of Economic and Social Affairs, Population Division. World Population Ageing 2013. [Online]. Available from: http://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2013.pdf. [retrieved: Jun. 2014]

[2] K. Issing, C. Blatteis, and H. Hensel, “Static temperature sensations, local static thermal comfort, and thermal thresholds in various age groups,” Pflügers Archiv European Journal of Physiology, vol. 394, 1982, pp. 41-41.

[3] Illuminating Engineering Society. Lighting and the Visual Environment for Senior Living. Illuminating Engineering Society, 2007.

[4] W. Davies, T. Cox, A. Kearon, B. Longhurst, and C. Webb, “Hearing loss in the built environment: The experience of elderly people,” Acta Acustica united with ACUSTICA, vol. 87, iss. 5, 2001, pp. 610-616.

[5] US Environmental Protection Agency. Portfolio Manager DataTrends: Energy Use Benchmarking. [Online]. Available from: http://www.energystar.gov/buildings/sites/default/uploads/tools/DataTrends_Energy_20121002.pdf?a807-9506 [retrieved: Jun. 2014]

[6] NREL, NREL: OpenStudio. [Online]. Available from: https://openstudio.nrel.gov/. [retrieved: Jun. 2014]

[7] Radsite, Radiance - A Validated Lighting Simulation Tool - Radsite. [Online]. Available from: http://www.radiance-online.org/. [retrieved: Jun. 2014]

[8] U.S. Department of Energy, Building Technologies Office: EnergyPlus Energy Simulation Software. [Online]. Available from: http://apps1.eere.energy.gov/buildings/energyplus/?utm_source=EnergyPlus&utm_medium=redirect&utm_campaign=EnergyPlus%2Bredirect%2B1. [retrieved: Jun. 2014]

[9] L. Schellen, W. van Marken Lichtenbelt, M. Loomans, J. Toftum, and M. De Wit, “Differences between young adults and elderly in thermal comfort, productivity, and thermal physiology in response to a moderate temperature drift and a steady-state condition,” Indoor air, vol. 20, iss. 4, 2010, pp. 273-283.

[10] R. Hwang, and C. Chen, “Field study on behaviors and adaptation of elderly people and their thermal comfort requirements in residential environments,” Indoor air, vol. 20, iss. 3, 2010, pp. 235-245.

[11] K. Collins, and E. Hoinville, “Temperature requirements in old age,” Building Services Engineering Research and Technology, vol. 1, iss. 4, 1980, pp. 165-172.

[12] D. DiLaura, The lighting handbook, 1st ed. New York, NY: Illuminating Engineering Society of North America, 2011.

[13] D. Cawthorne, “Daylighting and occupant health in buildings,” 1995. Ph.D. University of Cambridge.

[14] Climatemps, Climate of Changsha, China Average Weather, 2014. [Online]. Available from: http://www.changsha.climatemps.com/. [retrieved: Jun. 2014]

[15] Bureau of Labor Statistics. American Time Use Survey. [Online]. Available from: http://www.bls.gov/tus/. [retrieved: Jun. 2014]

Thickness Reduction Controller Design for Flying Gauge Change in a Cold Strip Mill

Tomoyoshi Ogasahara
Instrument and Control Engineering Research Dept.
JFE Steel Corp.,
1 Kokan-cho, Fukuyama, Hiroshima, Japan
t-ogasahara@jfe-steel.co.jp

Abstract—**This paper proposes a new flying gauge change (FGC) scheme for a continuous cold strip mill to achieve thickness reduction in a specified section including a welding point. Because this section is often processed as scrap, thickness reduction is desired as a means of reducing scrap weight. However, this is not possible with conventional FGC, which considers only one thickness change point, because two thickness change points exist simultaneously in the mill due to the short length of the specified section. The proposed scheme utilizes the transition pattern of the position of the two thickness change points during FGC to calculate the set values of the roll gap and roll speed. Thickness reduction is achieved by applying these set values to the rolling mill according to the tracking signals of the two thickness change points. The validity of the proposed method was verified by simulation and experiments. Application of this technology has improved cold strip yield.**

Keywords-control; flying gauge change; cold rolling; pysical model; actuator model; model based design.

I. Introduction

In a continuous cold strip mill, as shown in Figure 1, a coil is uncoiled by the pay-off reel, and its lead end is welded to the tail end of the preceding strip. Here, the two strips differ in hardness and dimensions, such as thickness and width, and the specified thickness after rolling also differs. Hence, the rolling mill must change the thickness of the strip at the welding point. After rolling, the strips are cut at the welds and coiled separately by the tension reel. Flying Gauge Change (FGC) was developed in order to achieve the necessary thickness change [1][2][3]. Although the most important issue in FGC is obviously thickness accuracy, it is also necessary to minimize strip tension deviations in order to prevent rolling trouble such as strip breakage. In Yamashita et al. [1], the responses of the actuators, which consist of the roll gap and roll speed, are synchronized during FGC, thereby minimizing strip tension deviations. In [2], the set values of the roll gap and speed are also synchronized. Furthermore, a dynamic setup model based on the measured value of strip thickness is proposed to compensate for setup model error. Hol et al. [3] explicitly introduces strip tension and thickness to the controller objective function and directly optimizes the time-domain response, and the inherent variation of the forward slip and friction conditions is mentioned.

On the other hand, a specified section including a welding point is often processed as scrap due to the mechanical properties of the strip or constraints of the downstream manufacturing processes. Therefore, as shown in Figure 2, thickness reduction in that section is desired as a means of reducing scrap weight, and thereby, improving yield. FGC must be applied in order to achieve this thickness reduction in a cold rolling mill. However, thickness reduction is not possible with the conventional FGC method, which considers only one thickness change point, because two thickness change points exist simultaneously in the mill due to the short length of the specified section. Figure 3 shows this situation.

This paper presents a new FGC method which makes it possible to achieve the prescribed two thickness changes. To simplify the problem, the size and hardness of the welded preceding and succeeding strips are the same. While this situation often occurs in actual operation, this can also be eliminated by a few modifications to the technology presented in this paper.

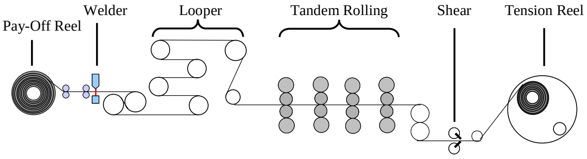


Figure 1. Outline of a Cold Strip Mill

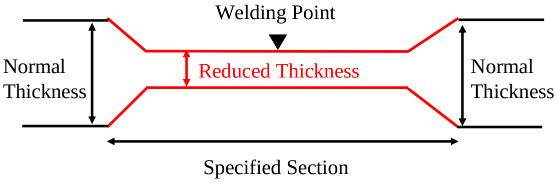


Figure 2. Thickness Reduction for Yield Improvement

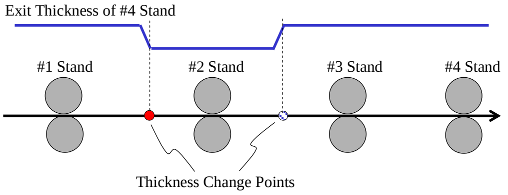


Figure 3. Two Thickness Change Points in the Mill

The outline of this paper is as follows. In Section II, a dynamical model and the principle of the conventional FGC are shown. In Section III, the details of the proposed FGC algorithm are explained. In Section IV, simulation and experimental results are presented. Finally, the conclusions are summarized in Section V.

II. DYNAMICAL MODEL OF A COLD STRIP MILL AND A CONVENTIONAL FGC METHOD

A dynamical model of a cold strip mill and a conventional FGC method are explained in this section. The model is used in designing the FGC controller.

A. *Dynamical model of a cold strip mill*

The model consists of physical models and actuator models. Each model is shown below, and the variables in the models are summarized at TABLE I. Note that the suffix i in the variables indicates the i^{th} stand.

- Rolling Force

$$P_i = wk_iK_t\sqrt{R_i'(H_i - h_i)}Q_i.$$

where,

$$K_t = (1-\sigma_{bi}/k_i)\left(1.05 + 0.1\frac{1-\sigma_{fi}/k_i}{1-\sigma_{bi}/k_i} - 0.15\frac{1-\sigma_{bi}/k_i}{1-\sigma_{fi}/k_i}\right),$$

$$R_i' = R_i\left(1 + \frac{16(1-\nu^2)}{E\pi}\frac{P_i}{w(H_i - h_i)}\right),$$

$$Q_i = 1.08 + 1.79r_i\mu_i\sqrt{\frac{R_i'}{H_i}} - 1.02r_i.$$

- Forward Slip Model

$$f_i = \tan^2\left(\frac{1}{2}\arctan\left(\sqrt{\frac{r_i}{1-r_i}}\right) - \frac{1}{4\mu_i}\log(\frac{H_i}{h_i}\frac{1-\sigma_{fi}/k_{fi}}{1-\sigma_{bi}/k_{bi}})\sqrt{\frac{h_i}{R_i'}}\right).$$

- Strip Tension Model

$$T_{i-i+1} = \frac{E}{L}\int(V_{i+1} - v_i)dt. \qquad (1)$$

- Thickness Model

$$h_i = S_i + P_i/M_i. \qquad (2)$$

- Roll Speed Actuator Model

$$V_{Ri}(s) = \frac{1}{T_i^v s + 1}V_{Ri}^{ref}(s). \qquad (3)$$

- Roll Gap Position Actuator Model

$$S_i(s) = \frac{1}{T_i^s s + 1}S_i^{ref}(s). \qquad (4)$$

- Transport Delay Model

$$H_{i+1} = h_i\exp(-L/v_i s). \qquad (5)$$

- Conservation Law of Mass Flow

$$V_i H_i = v_i h_i. \qquad (6)$$

In these models, rolling force and forward slip can be calculated according to the models in [4][5]. The strip tension between the i^{th} stand and $i+1^{th}$ stand is represented as (1). This model means that strip tension fluctuates when the velocity of the entry strip at the $i+1^{th}$ stand differs from that of the exit strip at the i^{th} stand. This is an important characteristic for FGC. In the thickness model shown in (2), thickness is determined by the roll gap S_i and elastic deformation of a work roll, which is represented as P_i/M_i. The actuators of the mill are electric motors and hydraulic cylinders. The former are for controlling the roll speed, and the latter for controlling the roll gap position. These actuators can be modeled as first order transfer functions like (3) and (4). The transport time delay is variable, depending on the strip velocity, and is modeled as (5). The conservation law of mass flow is described as (6).

TABLE I. PARAMETERS IN THE MODEL

Symbol	Notes
P_i	Rolling force
w	Strip width
k_i	Mean deformation resistance
k_{bi}	Entry deformation resistance
k_{fi}	Exit deformation resistance
K_t	Tension correction term
R_i'	Flattened work roll radius
H_i	Entry thickness
h_i	Exit thickness
Q_i	Influence of friction coefficient
σ_{bi}	Unit tension of entry side of a stand
σ_{fi}	Unit tension of exit side of a stand
r_i	Reduction rate
μ_i	Friction coefficient
ν	Poisson's ratio of work roll
f_i	Forward slip
T_{i-i+1}	Strip tension between i^{th} stand and $(i+1)^{th}$ stand
E	Young's modulus
L	Length between stands
V_i	Entry strip velocity; $V_{Ri}(f_i+1)h_i/H_i$
v_i	Exit strip velocity; $V_{Ri}(f_i+1)$
S_i	Roll gap
M_i	Mill modulus
s	Laplace operator
V_{Ri}	Roll speed
V_{Ri}^{ref}	Reference of roll speed
T_i^v	Time constant of roll speed actuator
S_i^{ref}	Reference of roll gap position
T_i^s	Time constant of roll gap position actuator

B. *Conventional FGC method*

The conventional FGC method is explained here. Figure 4 shows an example of the time chart of the action of the actuators during FGC. The roll gap is being closed when the thickness change point passes each of the stands. On the other hand, the roll speed of the i^{th} stand does not decrease when the point passes the i^{th} stand, but decreases when the point passes the $i+1^{th}$ stand. The reason for this can be explained by (1) and (6). From (6), we obtain

$$V_{i+1} = v_{i+1}h_{i+1} / H_{i+1}. \qquad (7)$$

Because h_{i+1} becomes smaller by closing the roll gap, V_{i+1} in (7) also becomes smaller. To minimize the deviation of strip tension, $v_i = V_{i+1}$ is required from (1). For this, it is necessary to decrease the speed of the i^{th} stand. Synchronization of the responses of the roll gap and speed to minimize the imbalance of the mass flow during the transition period is described in [1].

In order to calculate the strip velocity during FGC, information on the thickness and initial strip velocity are required. The reference thickness after FGC is determined by the setup calculation of the mill, and the initial strip velocity is usually estimated by $V_{Ri}(f_i+1)$. Finally, the roll speed during FGC is calculated by using the prescribed strip velocity and forward slip after FGC based on the forward slip model. The roll gap is calculated by (2).

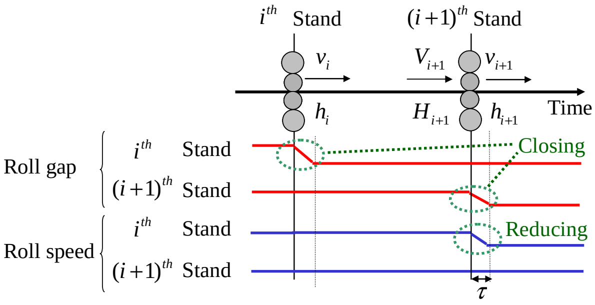


Figure 4. Time Chart of the Action of Actuators during FGC

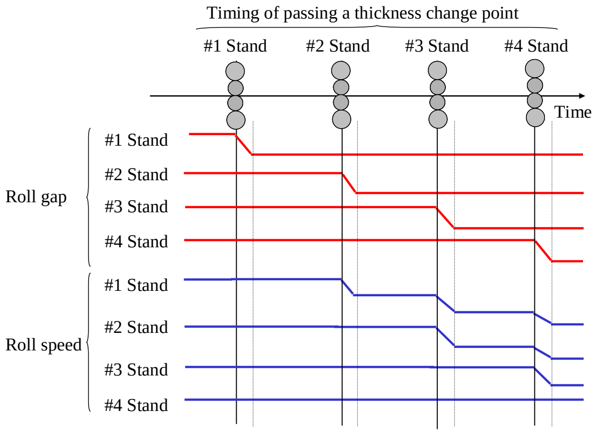


Figure 5. FGC Time Chart of a Conventional Method

FGC is achieved by applying these set values to the rolling mill according to the tracking signal of the thickness change point. This situation is shown in Figure 5. The above is the outline of the conventional method.

III. Thickness Reduction Controller Design

The thickness reduction controller design is presented in this section. The proposed method consists of three steps. The 1st step is to classify the transition pattern of the position of the two thickness change points during the FGC. The 2nd step is to calculate the set values of the roll gap and the roll speed based on the results of the 1st step. The 3rd step is to apply these set values to the rolling mill according to the tracking signals of the two thickness change points.

A. *Classification of the transition pattern*

The volume of the specified section between two thickness change points is invariant. Based on this fact, the transition pattern of the position of the two thickness points can be classified under several cases. Here, A denotes the first thickness change point, B denotes the second thickness change point, M is the volume of the specified section and h_i^{II} is the reference of the exit thickness of the i^{th} stand of the specified section.

Let us consider the pattern shown in Figure 6. Firstly, A exists between No. 1 stand and No. 2 stand in state-1. Then, A moves to between No. 2 stand and No. 3 stand in the following state-2. Because M is larger than $h_1^{II} \times L$, B cannot enter between No. 1 stand and No. 2 stand at that time. In the following state-3, B enters between No. 1 stand and No. 2 stand because M is smaller than $(h_1^{II} + h_2^{II}) \times L$. State-4 shows that A moves to between No. 3 stand and No. 4 stand. The reason for this is explained as follows: When state-1 exists, the inequality $h_2^{II} \times L \le h_1^{II} \times L \le M$ is obtained. Thus, B cannot enter between No. 2 stand and No. 3 stand in this state. In the following state-5, B enters between No. 2 stand and No. 3 stand since M is smaller than $(h_2^{II} + h_3^{II}) \times L$. The following state-6, state-7, and state-8 are obvious. Thus, the transition pattern is classified based on the volume of the specified section.

A summary of the transition pattern is shown in Figure 7. The transition pattern is classified under four cases. The subscripts to A and B indicate that the thickness change point passes the stand indicated by the numbers.

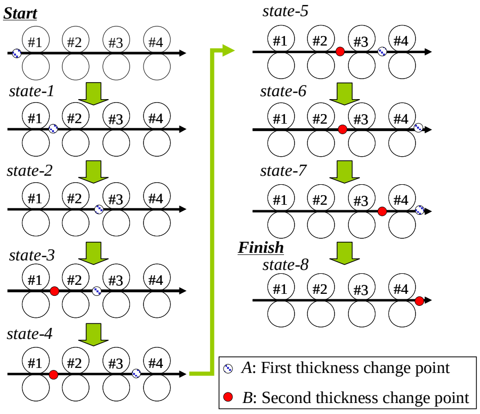


Figure 6. An Example of a Transition Pattern of the Position of the Two Thickness Change Points

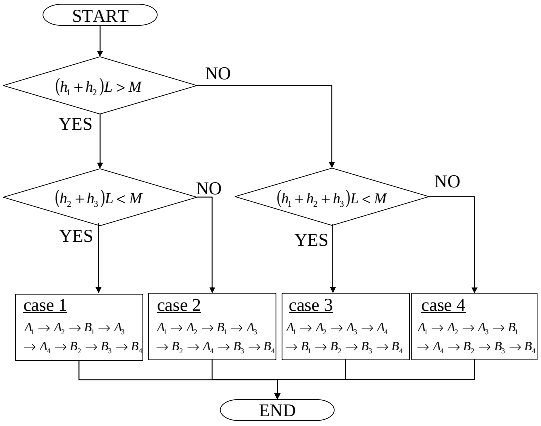

START
$(h_1+h_2)L>M$
NO
YES
$(h_2+h_3)L<M$
NO
YES
$(h_1+h_2+h_3)L<M$
NO
YES
case 1
$A_1 \to A_2 \to B_1 \to A_3 \to A_4 \to B_2 \to B_3 \to B_4$
case 2
$A_1 \to A_2 \to B_1 \to A_3 \to B_2 \to A_4 \to B_3 \to B_4$
case 3
$A_1 \to A_2 \to A_3 \to A_4 \to B_1 \to B_2 \to B_3 \to B_4$
case 4
$A_1 \to A_2 \to A_3 \to B_1 \to A_4 \to B_2 \to B_3 \to B_4$
END

Figure 7. The Logic of Classifying the Transition Pattern

B. Calculation of the set values of the roll gaps and the roll speed

The set values of the roll gap and the roll speed must be changed to achieve thickness reduction when the thickness change points pass each stand. As is described in Section II, the roll gap can be obtained by (1) and the roll speed can be calculated from the thickness information and the forward slip and initial roll speed. Classification of the pattern is performed in order to obtain the thickness information when the thickness change points pass each stand.

An example of a state for roll speed calculation is shown in Figure 8. Here, h_i^I denotes the initial exit thickness of the i^{th} stand and h_i^{II} means the reduced exit thickness of the i^{th} stand. The roll speed of the final stand, described as V_{R4}^I, is fixed. The variables, V_{R1}^*, V_{R2}^* and V_{R3}^* must be calculated. f_i^I is the forward slip of the i^{th} stand in the initial rolling condition, and f_i^{II} is the forward slip of the i^{th} stand in the reduced thickness rolling condition. The calculation is executed as follows.

From the conservation of mass flow in No. 4 stand, we obtain $V_{R3}^*(1+f_3^{II})h_3^I = V_{R4}^I(1+f_4^I)h_4^I$. This formula is deformed as

$$V_{R3}^* = V_{R4}^I \frac{1+f_4^I}{1+f_3^{II}} \frac{h_4^I}{h_3^I}. \tag{8}$$

We obtain $V_{R2}^*(1+f_2^{II})h_2^{II} = V_{R3}^*(1+f_3^{II})h_3^{II}$ from the conservation of mass flow in No. 3 stand. By using (8), this formula is deformed as

$$V_{R2}^* = V_{R3}^* \frac{1+f_3^{II}}{1+f_2^{II}} \frac{h_3^{II}}{h_2^{II}} = V_{R4}^I \frac{1+f_4^I}{1+f_2^{II}} \frac{h_4^I}{h_3^I} \frac{h_3^{II}}{h_2^{II}}. \tag{9}$$

From the conservation of mass flow in No. 2 stand, we obtain $V_{R1}^*(1+f_1^I)h_1^{II} = V_{R2}^*(1+f_2^{II})h_2^{II}$. This formula is deformed as

$$V_{R1}^* = V_{R2}^* \frac{1+f_2^{II}}{1+f_1^I} \frac{h_2^{II}}{h_1^{II}} = V_{R4}^I \frac{1+f_4^I}{1+f_1^I} \frac{h_4^I}{h_3^I} \frac{h_3^{II}}{h_1^{II}}. \tag{10}$$

Thus, the set values of roll speed are obtained as (8), (9) and (10). The roll speed should be obtained for all states. The results are shown in TABLE II.

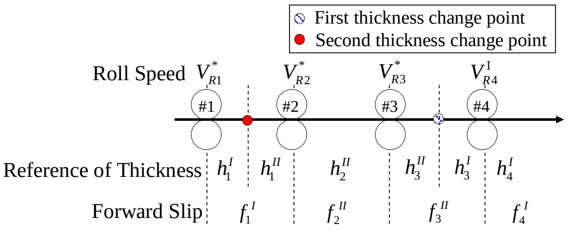


Figure 8. An Example of a State for Roll Speed Calculation

TABLE II. ROLL SPEED SETTINGS

State	V_{R1}^*	V_{R2}^*	V_{R3}^*
1	$V_{R4}^I \frac{1+f_4^I}{1+f_1^{II}} \frac{h_4^I}{h_1^I}$	$V_{R4}^I \frac{1+f_4^I}{1+f_2^I} \frac{h_4^I}{h_2^I}$	$V_{R4}^I \frac{1+f_4^I}{1+f_3^I} \frac{h_4^I}{h_3^I}$
2	$V_{R4}^I \frac{1+f_4^I}{1+f_1^{II}} \frac{h_4^I}{h_2^I} \frac{h_2^{II}}{h_1^{II}}$	$V_{R4}^I \frac{1+f_4^I}{1+f_2^{II}} \frac{h_4^I}{h_2^I}$	$V_{R4}^I \frac{1+f_4^I}{1+f_3^I} \frac{h_4^I}{h_3^I}$
3	$V_{R4}^I \frac{1+f_4^I}{1+f_1^I} \frac{h_4^I}{h_1^{II}} \frac{h_2^{II}}{h_2^I}$	$V_{R4}^I \frac{1+f_4^I}{1+f_2^{II}} \frac{h_4^I}{h_2^I}$	$V_{R4}^I \frac{1+f_4^I}{1+f_3^I} \frac{h_4^I}{h_3^I}$
4	$V_{R4}^I \frac{1+f_4^I}{1+f_1^I} \frac{h_4^I}{h_3^I} \frac{h_3^{II}}{h_1^{II}}$	$V_{R4}^I \frac{1+f_4^I}{1+f_2^{II}} \frac{h_4^I}{h_3^I} \frac{h_3^{II}}{h_2^{II}}$	$V_{R4}^I \frac{1+f_4^I}{1+f_3^{II}} \frac{h_4^I}{h_3^I}$
5	$V_{R4}^I \frac{1+f_4^I}{1+f_1^I} \frac{h_4^I}{h_3^I} \frac{h_3^{II}}{h_2^{II}} \frac{h_2^I}{h_1^I}$	$V_{R4}^I \frac{1+f_4^I}{1+f_2^I} \frac{h_4^I}{h_3^I} \frac{h_3^{II}}{h_2^{II}}$	$V_{R4}^I \frac{1+f_4^I}{1+f_3^{II}} \frac{h_4^I}{h_3^I}$
6	$V_{R4}^I \frac{1+f_4^{II}}{1+f_1^I} \frac{h_4^{II}}{h_2^{II}} \frac{h_2^I}{h_1^I}$	$V_{R4}^I \frac{1+f_4^{II}}{1+f_2^I} \frac{h_4^{II}}{h_2^{II}}$	$V_{R4}^I \frac{1+f_4^{II}}{1+f_3^{II}} \frac{h_4^{II}}{h_3^{II}}$
7	$V_{R4}^I \frac{1+f_4^{II}}{1+f_1^I} \frac{h_4^{II}}{h_3^{II}} \frac{h_3^I}{h_1^I}$	$V_{R4}^I \frac{1+f_4^{II}}{1+f_2^I} \frac{h_4^{II}}{h_3^{II}} \frac{h_3^I}{h_2^I}$	$V_{R4}^I \frac{1+f_4^{II}}{1+f_3^I} \frac{h_4^{II}}{h_3^{II}}$
8	$V_{R4}^I \frac{1+f_4^I}{1+f_1^I} \frac{h_4^I}{h_1^I}$	$V_{R4}^I \frac{1+f_4^I}{1+f_2^I} \frac{h_4^I}{h_2^I}$	$V_{R4}^I \frac{1+f_4^I}{1+f_3^I} \frac{h_4^I}{h_3^I}$

C. Applying the set values to the mill

Thickness reduction is achieved by applying the set values to the rolling mill according to the tracking signals of the two thickness change points. The control system is shown in Figure 9. The state can be judged by the tracking signals, and the appropriate set values are selected. The set values are then sent to the mill controller, and the roll gap and roll speed are controlled.

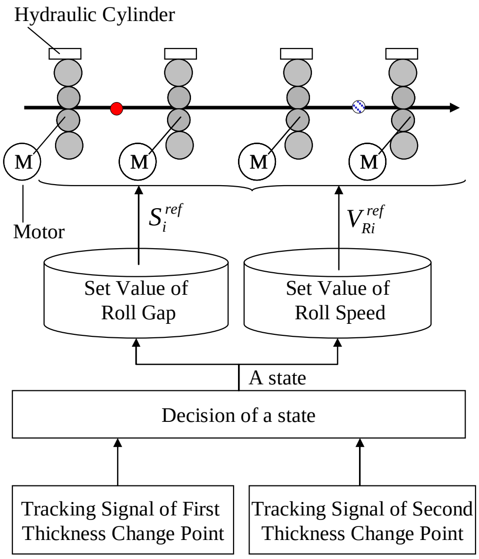


Figure 9. The Proposed Control System

The procedure outlined above is the proposed method.

IV. SIMULATION AND EXPERIMENTS

The results of a simulation and experiments are presented here. In the chart graph of the simulation and experiments, the value on the vertical axis is normalized by the initial value, which is the value before applying thickness reduction control.

A. Simulation

Figures 10 and 11 show the actions of the roll gap and roll speed, respectively. In Figure 10, the roll gap is being closed when the first thickness change point passes each stand, and it is being opened when the second thickness change point passes each stand. Thus, the action of the roll gap is simple and the existence of the two thickness change points in the mill is confirmed by the graph. The action of the roll speed, especially at No. 1 stand in Figure 11, is somewhat complex. It was found that the effect of the two thickness change points appears in the upstream stand speed. The strip tension will deviate unless the roll speed is set appropriately.

Figure 12 shows the strip tension deviations. In order to maintain unit tension, which is the tension per cross sectional area of the strip, the set value of tension is decreased. Strip deviation is within 20% in this graph. In actual operation, strip tension deviation within 30% is desired in order to prevent rolling trouble. Hence, this method achieves tension deviation within the reference value.

Figure 13 is a thickness chart. A thickness reduction of nearly 50% is achieved by using the proposed method.

Thus, these simulation results verify the validity of the proposed method from the viewpoints of thickness and strip tension deviations.

B. Experimental results

The experimental results are shown in Figure 14 and Figure 15. In this control method, the reference of thickness reduction is 11% compared to the thickness before application. Strip tension deviation within 10% was obtained in this experiment. Hence, this method satisfies the condition for actual operation. Thickness fluctuated at the welding point at 19 [sec] due to the difference of the deformation resistance of the welded strips.

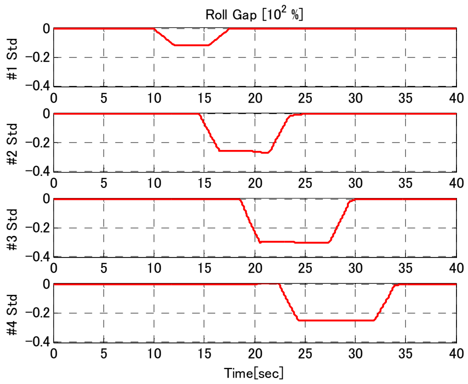


Figure 10. Roll Gap (Simulation)

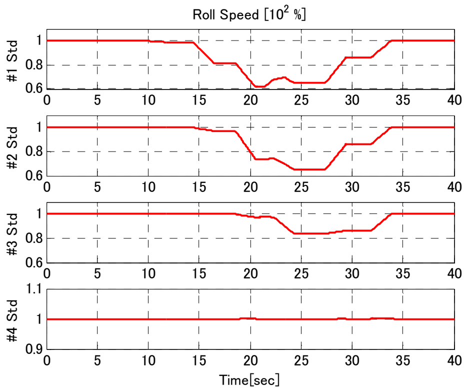


Figure 11. Roll Speed (Simulation)

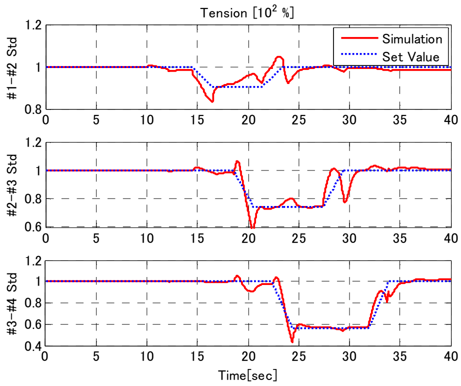


Figure 12. Strip Tension (Simulation)

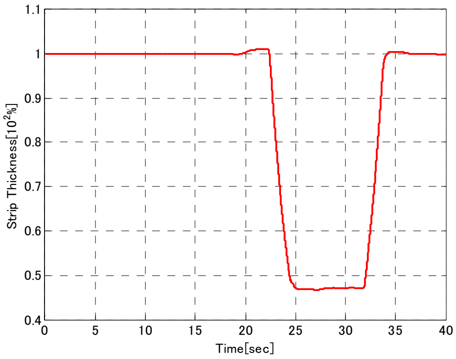


Figure 13. Exit Thickness of Final Stand (Simulation)

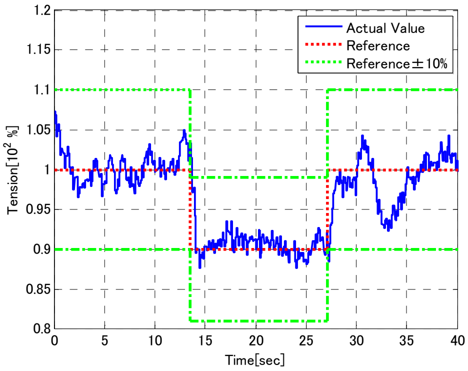


Figure 14. Strip Tension between No. 3 Stand and No. 4 Stand (Experiment)

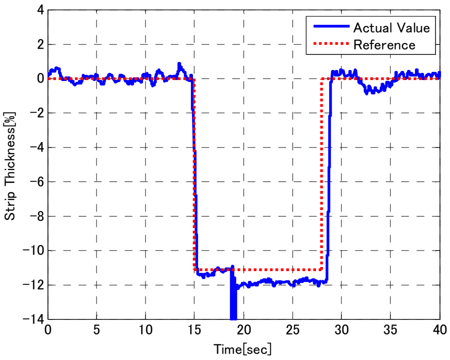


Figure 15. Exit Thickness of Final Stand (Experiment)

However, in total, thickness accuracy is controlled to within 1%, demonstrating accurate thickness control. Thus, the validity of this method was also verified experimentally.

V. Conclusions

In a continuous cold strip mill, a specified section including a welding point is often processed as scrap. In order to reduce scrap weight and improve product yield, reduction of the thickness of this section is desired.

A new FGC method which achieves two thickness changes is proposed in this paper. The proposed method classifies the transition pattern of the two thickness change points in the mill and calculates the set values of the roll gap and the roll speed, thereby simultaneously achieving the desired thickness reduction and reducing tension deviations.

In a simulation, a thickness reduction of nearly 50% and strip deviation within 20% were obtained. Experimental results also showed that thickness reduction and low tension deviation are achieved by this FGC method.

Thus, the validity of the proposed method was verified by both simulation and experiments. Application of this technology has resulted in improved cold strip yield.

References

[1] M. Yamashita, I. Yarita, H. Abe, T. Mikuriya, and F. Yanagishima, “Technologies of flying gauge change in fully continuous cold rolling mill for thin gauge steel strips,” IRSID Rolling Conf, vol. 2, 1987, pp. E.36.1-E.36.11.

[2] J. S. Wang, Z. Y. Jiang, A. K. Tieu, X. H. Liu, and G. D. Wang, “A flying gauge change model in tandem cold strip mill,” Journal of materials processing technology, vol. 204, Issues 1-3, Aug. 2008, pp. 152-161.

[3] C. W. J. Hol, S. Sujoto, M. de Boer, V. Beentjes, M. Price, C. W. Scherer, and A. A. J. van der Winden, “Optimal Feedforward Filter Design for Flying Gauge Changes of a Continuous Cold Mill,” IFAC World Congress, vol. 18. Aug. 2011, pp. 8545-8551.

[4] D. R. Bland and H. Ford, “The Calculation of Roll Force and Torque in Cold Strip Rolling with Tensions,” Proceedings of the Institution of Mechanical Engineers, vol. 159, Jun. 1948, pp. 144-163.

[5] R. Hill, The Mathematical Theory of Plasticity. Oxford University Press, Aug. 1998.

Application of the Butler-Volmer Equation in Mathematical Modelling of Amperometric Biosensor

Dainius Šimelevičius

Faculty of Mathematics and Informatics
Vilnius University, Didlaukio 47
LT-08303 Vilnius, Lithuania
Email: dainius.simelevicius@mif.vu.lt

Karolis Petrauskas

Faculty of Mathematics and Informatics
Vilnius University, Didlaukio 47
LT-08303 Vilnius, Lithuania
Email: karolis.petrauskas@mif.vu.lt

***Abstract*—A computational model of a multilayered amperometric biosensor is presented in this paper. Models of biosensors usually simplifies the electrochemistry of biosensors considering that all electrochemical reactions have infinite rates. While in most cases such approximations are tolerable, at lower electrode potentials these models would not be accurate enough. The model proposed in this paper models the electrochemistry occurring during the biosensor operation more accurately, using the Butler-Volmer equation. Computational experiments showed that models with the Butler-Volmer equation may be valuable tools aiding the analysis of the electrochemistry of amperometric biosensors.**

Keywords–Modelling; Reaction-diffusion; Biosensor; Amperometric; Butler-Volmer Equation.

I. Introduction

A biosensor is an electronic measuring device designed for measuring a concentration of some specific substance (analyte) in a solution. Device specificity for a particular substance is achieved by some biological material, usually an enzyme [1], [2], [3]. An amperometric biosensor assesses concentration of the analyte through measurement of a current on a working electrode [4], [5]. Biosensors are widely used in various applications that require fast quantitative analysis [6], [7], [8], [9], [10].

Manufacturing of a novel biosensor may be very expensive, as it may require a lot of experiments in a laboratory. It is wise to conduct computational experiments prior to physical ones. In order to do that, a mathematical model of the biosensor should be built [11], [12]. Mathematical models of biosensors are built for a few decades already [13], [14], [15].

The biosensor modelled in this paper was modelled previously in two papers. A model in paper [16] assumes rate of an electrochemical reaction as infinite and concentration of an electrochemical reaction reactant on the electrode surface as permanently reduced to zero. An experiment was conducted and another model of the same type of biosensor was presented recently in our paper [17]. A model in paper [17] is augmented with equations representing oxidation of the mediator by oxygen when an experiment is conducted in the aerobic conditions. Computational experiments showed that in the case when N-methylphenazonium methyl sulfate is used as a mediator the mediator oxidation by oxygen may be neglected [17]. The model presented in [17] considers that the rate of an electrochemical reaction as infinite too.

This paper is an extension of our research [17]. This time the model targets electrochemical questions. The model of biosensor presented in this paper considers the fact that the rate of the electrochemical reaction is not infinite. The rate of electrochemical reaction is modelled using the Butler-Volmer equation [18]. The model is not superior with respect to models presented in [16] and [17] but rather it is a helpful tool enabling us to analyse the electrochemical aspect of the amperometric biosensor. We demonstrate that at certain situations the assumption of infinite electrochemical reaction rate may not be a suitable approach.

Behaviour of the biosensor was numerically analysed at various values of input parameters of the model. Influence of an electrode potential, as well as of a standard rate constant on the biosensor response were investigated.

II. Mathematical Model

A. Reaction Scheme

The following reactions take place during operation of this amperometric biosensor [19], [20], [21],

$$\mathrm{GDH_{ox}} + \mathrm{glucose} \xrightarrow{k_1} \mathrm{GDH_{red}} + \mathrm{gluconolactone} \tag{1a}$$

$$\mathrm{GDH_{red}} + \mathrm{PMS_{ox}} \xrightarrow{k_2} \mathrm{GDH_{ox}} + \mathrm{PMS_{red}} \tag{1b}$$

$$\mathrm{PMS_{red}} \rightleftharpoons \mathrm{PMS_{ox}} + 2e^- \tag{1c}$$

where GDH is glucose dehydrogenase and PMS is N-methylphenazonium methyl sulfate.

Further in this paper we use an abstract notation of the chemical species. Glucose is called the substrate S and gluconolactone is called the product P. $\mathrm{E_{ox}}$ is $\mathrm{GDH_{ox}}$, $\mathrm{E_{red}}$ is

$\mathrm{GDH_{red}}$, $\mathrm{M_{ox}}$ denotes $\mathrm{PMS_{ox}}$ and $\mathrm{M_{red}}$ denotes $\mathrm{PMS_{red}}$.

$$\mathrm{E_{ox}} + \mathrm{S} \xrightarrow{k_1} \mathrm{E_{red}} + \mathrm{P} \tag{2a}$$

$$\mathrm{E_{red}} + \mathrm{M_{ox}} \xrightarrow{k_2} \mathrm{E_{ox}} + \mathrm{M_{red}} \tag{2b}$$

$$\mathrm{M_{red}} \rightleftharpoons \mathrm{M_{ox}} + 2e^- \tag{2c}$$

B. Biosensor Principal Structure

The biosensor is comprised of three layers (compartments). Mediums of these layers are made of different materials and so are diffusion properties of species in these layers. The mathematical model includes these three layers plus additional layer which is called the diffusion layer. The diffusion layer is part of a solution in which concentrations of species are different compared to the bulk solution. We use the Nernst model of a diffusion layer. By the definition of this model, the diffusion front is stopped by the convection at a certain distance from the electrode and this distance is equal to the thickness of the diffusion layer [22], [23].

Starting from the electrode surface, layers of the biosensor go in the following order: the enzyme layer, the PVA layer, the terylene membrane layer and the diffusion layer. Thicknesses of these layers are defined d_1, d_2, d_3 and d_4, respectively. Distances between the electrode surface and boundaries of biosensor layers are denoted as a_1, a_2, a_3 and a_4 (see Figure 1).

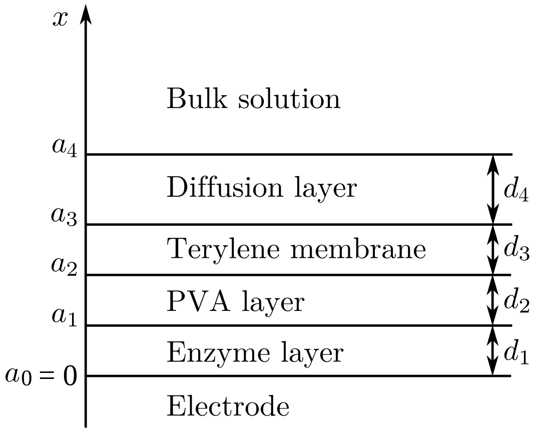


Figure 1. The principal structure of the biosensor.

Enzyme molecules are present in enzyme layer only. As a result biochemical reactions take place in the enzyme layer only. Diffusion of species take place in all four layers of the biosensor. However, enzyme molecules are large and considered as immobile (not influenced by diffusion).

C. Governing Equations

In our model governing equations consist of two parts: kinetics and diffusion. Kinetics is defined according to the law of mass action [24], [15] and diffusion is defined according to the Fick's second law [18] ($0 < x < a_1$, $t > 0$):

$$\frac{\partial e_{\mathrm{ox}}}{\partial t} = -k_1 e_{\mathrm{ox}} s_1 + k_2 e_{\mathrm{red}} m_{\mathrm{ox},1}, \tag{3}$$

$$\frac{\partial e_{\mathrm{red}}}{\partial t} = k_1 e_{\mathrm{ox}} s_1 - k_2 e_{\mathrm{red}} m_{\mathrm{ox},1}, \tag{4}$$

$$\frac{\partial s_1}{\partial t} = D_{\mathrm{S},1} \frac{\partial^2 s_1}{\partial x^2} - k_1 e_{\mathrm{ox}} s_1, \tag{5}$$

$$\frac{\partial m_{\mathrm{ox},1}}{\partial t} = D_{\mathrm{M_{ox}},1} \frac{\partial^2 m_{\mathrm{ox},1}}{\partial x^2} - k_2 e_{\mathrm{red}} m_{\mathrm{ox},1}, \tag{6}$$

$$\frac{\partial m_{\mathrm{red},1}}{\partial t} = D_{\mathrm{M_{red}},1} \frac{\partial^2 m_{\mathrm{red},1}}{\partial x^2} + k_2 e_{\mathrm{red}} m_{\mathrm{ox},1}, \tag{7}$$

where x is the distance from the electrode surface, t is time from the beginning of an experiment, $e_{\mathrm{ox}}(x,t)$ and $e_{\mathrm{red}}(x,t)$ are concentrations of the oxidized ($\mathrm{E_{ox}}$) and the reduced ($\mathrm{E_{red}}$) enzyme molecules, respectively; $s_1(x,t)$ is a concentration of the substrate in the enzyme layer; $m_{\mathrm{ox},1}(x,t)$ and $m_{\mathrm{red},1}(x,t)$ are concentrations of the oxidized ($\mathrm{M_{ox}}$) and the reduced ($\mathrm{M_{red}}$) forms of the mediator in the enzyme layer and $D_{\mathrm{S},1}$, $D_{\mathrm{M_{ox}},1}$, $D_{\mathrm{M_{red}},1}$ are diffusion coefficients of the corresponding species defined by subscripts. Last numeric subscripts in definitions of concentrations and diffusion coefficients show the number of the biosensor layer, i.e., 1 is the enzyme layer. Equations corresponding to enzyme concentrations, lack diffusion term as enzyme molecules are considered immobile. There is no equation corresponding to the product P, because its concentration does not influence other processes defined by the model.

Governing equations in other three layers are simpler as there are diffusion terms only in these equations ($a_{i-1} < x < a_i$, $t > 0$, $i = 2, 3, 4$):

$$\frac{\partial s_i}{\partial t} = D_{\mathrm{S},i} \frac{\partial^2 s_i}{\partial x^2}, \tag{8}$$

$$\frac{\partial m_{\mathrm{ox},i}}{\partial t} = D_{\mathrm{M_{ox}},i} \frac{\partial^2 m_{\mathrm{ox},i}}{\partial x^2}, \tag{9}$$

$$\frac{\partial m_{\mathrm{red},i}}{\partial t} = D_{\mathrm{M_{red}},i} \frac{\partial^2 m_{\mathrm{red},i}}{\partial x^2}, \tag{10}$$

where $i = 2$ corresponds to the PVA layer, $i = 3$ corresponds to the terylene membrane layer and $i = 4$ corresponds to the diffusion layer.

D. Initial Conditions

We model the case when the biosensor is immersed into a solution which lacks the substrate and the mediator prior to experiment. We assume that experiment starts when the substrate and the mediator appear in the bulk solution. This is defined by the initial conditions ($t = 0$),

$$e_{\mathrm{red}}(x,0) = 0, \quad e_{\mathrm{ox}}(x,0) = e_0, \quad 0 < x < a_1, \tag{11}$$

$$s_i(x,0) = 0, \quad a_{i-1} \le x \le a_i, \quad i = 1,2,3, \tag{12}$$

$$m_{\mathrm{ox},i}(x,0) = 0, \quad a_{i-1} \le x \le a_i, \quad i = 1,2,3, \tag{13}$$

$$m_{\mathrm{red},i}(x,0) = 0, \quad a_{i-1} \le x \le a_i, \quad i = 1,2,3, \tag{14}$$

$$s_4(x,0) = m_{\mathrm{ox},4}(x,0) = 0, \quad a_3 \le x < a_4, \tag{15}$$

$$m_{\mathrm{red},4}(x,0) = 0, \quad a_3 \le x \le a_4, \tag{16}$$

$$s_4(a_4,0) = s_0, \quad m_{\mathrm{ox},4}(a_4,0) = m_0, \tag{17}$$

where e_0 is the total concentration of the enzyme ($e_0 = e_{\mathrm{ox}}(x,t) + e_{\mathrm{red}}(x,t), \forall x,t : x \in (0, a_1), t > 0$), s_0 is the substrate concentration and m_0 is the concentration of the oxidized form of the mediator in the bulk solution.

E. Matching Conditions

Species have different diffusion coefficients in different layers of the biosensor, thus matching conditions have to be defined ($t > 0$, $i = 1, 2, 3$) [22], [23],

$$D_{\mathrm{S},i}\,\frac{\partial s_i}{\partial x}\bigg|_{x=a_i} = D_{\mathrm{S},i+1}\,\frac{\partial s_{i+1}}{\partial x}\bigg|_{x=a_i}, \tag{18}$$

$$s_i(a_i,t) = s_{i+1}(a_i,t), \tag{19}$$

$$D_{\mathrm{M_{ox}},i}\,\frac{\partial m_{\mathrm{ox},i}}{\partial x}\bigg|_{x=a_i} = D_{\mathrm{M_{ox}},i+1}\,\frac{\partial m_{\mathrm{ox},i+1}}{\partial x}\bigg|_{x=a_i}, \tag{20}$$

$$m_{\mathrm{ox},i}(a_i,t) = m_{\mathrm{ox},i+1}(a_i,t), \tag{21}$$

$$D_{\mathrm{M_{red}},i}\,\frac{\partial m_{\mathrm{red},i}}{\partial x}\bigg|_{x=a_i} = D_{\mathrm{M_{ox}},i+1}\,\frac{\partial m_{\mathrm{red},i+1}}{\partial x}\bigg|_{x=a_i}, \tag{22}$$

$$m_{\mathrm{red},i}(a_i,t) = m_{\mathrm{red},i+1}(a_i,t), \tag{23}$$

where $i = 1$ corresponds to the boundary between the enzyme layer and the PVA layer, $i = 2$ corresponds to the boundary between the PVA layer and the terylene membrane layer, whereas $i = 3$ corresponds to the boundary between the terylene membrane layer and the diffusion layer.

The matching conditions define that fluxes of the species exiting the layer are equal to the fluxes entering the neighbouring layer of the biosensor. Additionally, concentrations of species on the surface of one layer are assumed to be equal to concentrations on the surface of the neighbouring layer.

F. Boundary Conditions

Concentrations of the species in the bulk solution are kept constant ($t > 0$),

$$s_4(a_4,t) = s_0, \tag{24}$$

$$m_{\mathrm{ox},4}(a_4,t) = m_0, \tag{25}$$

$$m_{\mathrm{red},4}(a_4,t) = 0. \tag{26}$$

$\mathrm{M_{red}}$ is the reactant of the electrochemical reaction (2c) while $\mathrm{M_{ox}}$ is the product. Stoichiometry of the reaction (2c) suggests that the amount of $\mathrm{M_{red}}$ consumed is equal to the amount of $\mathrm{M_{ox}}$ produced. Thus, the flux of $\mathrm{M_{red}}$ on the electrode surface is equal to the flux of $\mathrm{M_{ox}}$, but in the opposite direction. Also, according to the Faraday's Law the flux is proportional to the current. These relations are expressed by the following boundary condition ($t > 0$),

$$An_eFD_{\mathrm{M_{red}},1}\,\frac{\partial m_{\mathrm{red},1}}{\partial x}\bigg|_{x=0} = -An_eFD_{\mathrm{M_{ox}},1}\,\frac{\partial m_{\mathrm{ox},1}}{\partial x}\bigg|_{x=0} = i(t). \tag{27}$$

The flux of the substrate on the electrode is equal to zero, as the substrate does not take part in any electrochemical reaction ($t > 0$),

$$D_{\mathrm{S},1}\,\frac{\partial s_1}{\partial x}\bigg|_{x=0} = 0. \tag{28}$$

G. Biosensor Response

The measured current is usually assumed as the response of an amperometric biosensor in physical experiments. At pH = 5–9 the electrochemical reaction (2c) is reversible. Two electrons are transferred during the charge transfer [21]. The biosensor current $i(t)$ at time t was expressed by the Butler-Volmer equation [18], [25],

$$i(t) = An_eFk^0\Big[M_{red}(0,t)e^{(1-\alpha)n_ef(E-E_0)} - M_{ox}(0,t)e^{-\alpha n_ef(E-E_0)}\Big], \tag{29}$$

where $i(t)$ is the faradaic current generated by the electrochemical reaction (2c), A is the electrode surface, n_e is the number of electrons involved in a charge transfer at the electrode surface, k^0 is the standard rate constant, E_0 is the standard potential, α is the transfer coefficient, E is the electrode potential, $f = F/RT$; F is the Faraday constant, $F = 96\,486$ C/mol, R is the gas constant, $R = 8.314$ J mol^{-1} K^{-1}, T is the absolute temperature.

We assume that the system approaches a steady state as $t \to \infty$,

$$i_{\mathrm{st}} = \lim_{t\to\infty} i(t), \tag{30}$$

where i_{st} is the steady-state biosensor current.

III. Numerical Simulation

There is no known analytical solution for the problem (3)–(29). Therefore, the problem was solved numerically, using the finite difference technique [26], [27]. An implicit finite difference scheme was built on a uniform discrete grid with 50 points in the space direction for each modelled layer corresponding to a certain time moment. The simulator has been programmed by the authors in C++ programming language [28].

In the numerical simulation, the biosensor response time was assumed as the time when the change of the biosensor current remains very small during a relatively long term. A special dimensionless decay rate, ε, was used,

$$t_{\mathrm{r}} = \min_{i(t)>0}\left\{t : \frac{t}{i(t)}\left|\frac{\mathrm{d}i(t)}{\mathrm{d}t}\right| < \varepsilon\right\}, \quad i(t_{\mathrm{r}}) \approx i_{\mathrm{st}}, \tag{31}$$

where t_{r} is the biosensor response time. The decay rate value $\varepsilon = 10^{-2}$ was used in the calculations.

In all numerical experiments the following values were kept constant if not stated otherwise [17], [29], [30], [31], [21]:

$$\begin{aligned}
&d_1 = 5\times10^{-6}\,\text{m},\quad d_2 = 1\times10^{-6}\,\text{m},\\
&d_3 = 1.2\times10^{-5}\,\text{m},\quad d_4 = 1.5\times10^{-4}\,\text{m},\\
&D_{\text{S},1} = D_{\text{M}_{\text{ox}},1} = D_{\text{M}_{\text{red}},1} = 1.5\times10^{-10}\,\text{m}^2/\text{s},\\
&D_{\text{S},2} = D_{\text{M}_{\text{ox}},2} = D_{\text{M}_{\text{red}},2} = 4.2\times10^{-10}\,\text{m}^2/\text{s},\\
&D_{\text{S},3} = D_{\text{M}_{\text{ox}},3} = D_{\text{M}_{\text{red}},3} = 3.75\times10^{-10}\,\text{m}^2/\text{s},\\
&D_{\text{S},4} = 6.77\times10^{-10}\,\text{m}^2/\text{s},\\
&D_{\text{M}_{\text{ox}},4} = D_{\text{M}_{\text{red}},4} = 4.57\times10^{-10}\,\text{m}^2/\text{s},\\
&e_0 = 1\times10^{-3}\,\text{mol/m}^3,\quad m_0 = 5\times10^{-2}\,\text{mol/m}^3,\\
&s_0 = 4.98\,\text{mol/m}^3,\quad k_1 = 8.1\times10^{2}\,\text{m}^3\,\text{mol}^{-1}\,\text{s}^{-1},\\
&k_2 = 6.7\times10^{4}\,\text{m}^3\,\text{mol}^{-1}\,\text{s}^{-1},\quad n_\text{e} = 2,\\
&k^0 = 10^{-5}\,\text{m/s},\quad \alpha = 0.5,\\
&A = 4.5\times10^{-6}\,\text{m}^2,\quad T = 293\,\text{K}.
\end{aligned}\tag{32}$$

IV. Results and Discussion

A. Model Validation

Experimental calibration curve was compared with simulated calibration curves. Results are depicted in Figure 2.

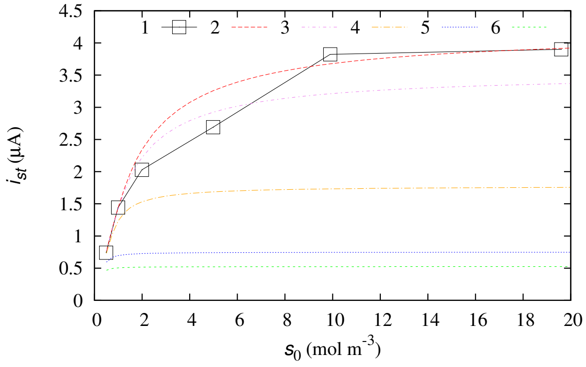


Figure 2. The dependence of the steady-state current on the substrate concentration during physical experiment (1), simulated with the infinite rate of electrochemical reaction (2), simulated at certain electrode potential $(E - E_0)$: 0.1 V (3), 0.05 V (4), 0.02 V (5), 0.01 V (6).

Physical experiments were carried out at potentiostatic conditions, the potential of the working electrode E was set at 0.4 V *vs.* $Ag/AgCl$ [17]. Standard potential E_0 of redox pair M_{red}/M_{ox} is -0.005 V *vs.* $Ag/AgCl$ [32], thus $(E - E_0) = 0.405$ V. In this case the forward electrochemical reaction becomes very fast and the rate of it may be modelled as infinite [17]. The model from paper [17] at anaerobic conditions was used to simulate the biosensor operation with infinite electrochemical reaction rate as depicted by the curve 2.

Using the finite difference technique with the model (3)-(29) becomes impractical at high potential values because modelling very high electrochemical reaction rate requires very small step in time which tremendously increases demand for a computational power. However the model (3)-(29) allows the modelling of lower potentials. Biosensor operation was modelled at potential values $(E - E_0)$ from 0.01 V to 0.1 V. As one can observe from Figure 2 the biosensor response is higher at higher potential values which is consistent with the chemical logic. The model cannot be directly validated with the experimental data but it is evident from Figure 2 that with the higher potential, biosensor response approaches the one observed experimentally.

One more interesting dependence may be observed from Figure 2, the lower the potential, the shorter the range of substrate concentrations in which biosensor may operate. The biosensor is almost entirely insensitive to the change in substrate concentration when the potential is the lowest (0.02 V and 0.01 V). This possibly indicates that the charge transfer is the slowest (limiting) process in this situation.

B. Biosensor Response vs. Electrode Potential

It is important to understand the dependence of the biosensor response on the electrode potential. Thus this dependence was more thoroughly investigated at three different substrate concentrations. Results are depicted in Figure 3.

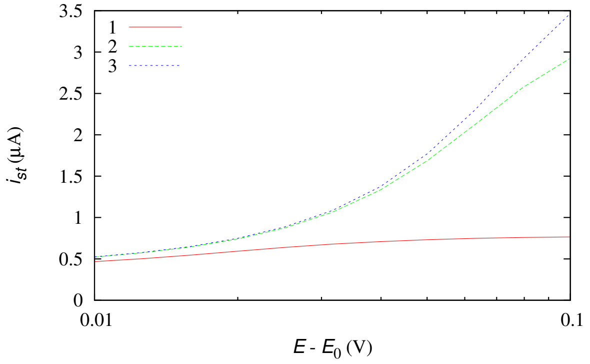


Figure 3. The dependence of the steady-state current on the electrode potential $(E - E_0)$ at different substrate concentrations s_0: 0.5 mol/m^3 (1), 5 mol/m^3 (2), 50 mol/m^3 (3).

At low substrate concentration (curve 1) the biosensor response dependence on the electrode potential is little pronounced. This indicates that in this case the charge transfer is not a limiting process. At higher substrate concentrations (curves 2 and 3), the biosensor response shows stronger dependence on the electrode potential. At these two concentrations charge transfer is possibly a limiting process at lower part of investigated potential range because in this part of the range the curves 2 and 3 coincide. At higher potentials biosensor shows dependence on both the electrode potential and substrate concentration.

C. Biosensor Response vs. Standard Rate Constant

Standard rate constant k^0 is the constant defining the kinetics of electrochemical reaction (2c). This constant shows whether electrochemical reaction reaches equilibrium fast or slow. The value of the constant may range from 10^{-11} m/s for very slugish kinetics to 0.1 m/s for very fast electron-transfer processes [18]. It is important to investigate how standard rate constant of $\text{M}_\text{red}/\text{M}_\text{ox}$ redox couple affects the biosensor response. This dependence was investigated at several values of substrate concentration. Results are depicted in Figure 4.

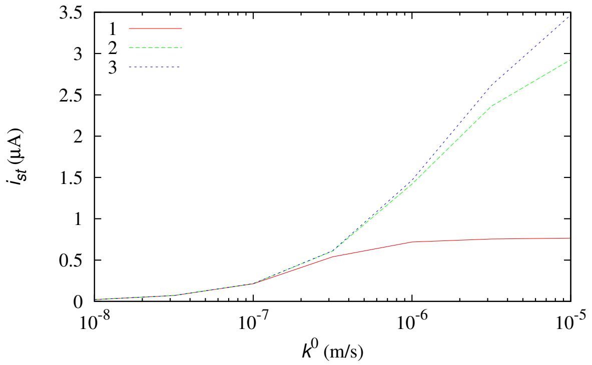


Figure 4. The dependence of the steady-state current on the standard rate constant at different substrate concentrations s_0: $0.5\,\text{mol/m}^3$ (1), $5\,\text{mol/m}^3$ (2), $50\,\text{mol/m}^3$ (3); $E - E_0 = 0.1\,\text{V}$.

Slugish electron-transfer is not desirable in amperometric biosensor because in such a case the electrochemical reaction may be the slowest (limiting) process and solely determine the biosensor response. This is possibly the case when $k^0 \in [10^{-8}\,\text{m/s}..10^{-7}\,\text{m/s}]$ because the biosensor response is not dependent on the substrate concentration even though it is varied by two orders of magnitude (see Figure 4). The biosensor response is dependent on the substrate concentration in the range $[10^{-7}\,\text{m/s}..10^{-6}\,\text{m/s}]$ when concentration is relatively low. Biosensor is not responsive to the change in substrate concentration at higher concentrations though. When $k^0 \in [10^{-6}\,\text{m/s}..10^{-5}\,\text{m/s}]$ the charge transfer is not limiting process, biosensor response is different at all three values of substrate concentration.

Results show that from this point of view the mediator for the biosensor was chosen very successfully as the standard rate constant for the redox couple $\text{M}_\text{red}/\text{M}_\text{ox}$ is quite high ($k^0 = 10^{-5}\,\text{m/s}$) [30].

V. Conclusion

The Butler-Volmer equation may be used in mathematical models of amperometric biosensors. Models with the Butler-Volmer equation may provide valuable information about biosensor behaviour at different values of electrochemical parameters. However numerical simulation using finite difference technique is impractical in cases when electrode potential is high. In such cases models assuming infinite rate of electrochemical reaction should be used.

The models of amperometric biosensors with the Butler-Volmer equation may be useful in the design phase of a biosensor by providing information about the minimum electrode potential at which biosensor may be successfully operated. Also such models may provide insight into which species may be used as mediators prior to the biosensor manufacture.

Acknowledgment

This research was funded by the European Social Fund under the Global Grant Measure, Project No. VP1-3.1-ŠMM-07-K-01-073/MTDS-110000-583.

References

[1] A. P. F. Turner, I. Karube, and G. S. Wilson, Biosensors: Fundamentals and Applications. Oxford: Oxford University Press, 1987.

[2] F. W. Scheller and F. Schubert, Biosensors, ser. Techniques and Instrumentation in Analytical Chemistry. Amsterdam: Elsevier, 1992, vol. 11.

[3] B. D. Malhotra and A. Chaubey, "Biosensors for clinical diagnostics industry," Sens. Actuators B, vol. 91, no. 1-3, 2003, pp. 117–127.

[4] A. Chaubey and B. Malhotra, "Mediated biosensors," Biosens. Bioelectron., vol. 17, no. 6-7, 2002, pp. 441–456.

[5] N. J. Ronkainen, H. B. Halsall, and W. R. Heineman, "Electrochemical biosensors," Chem. Soc. Rev., vol. 39, no. 5, 2010, pp. 1747–1763.

[6] K. R. Rogers, "Biosensors for environmental applications," Biosens. Bioelectron., vol. 10, no. 6-7, 1995, pp. 533–541.

[7] F. W. Scheller, F. Schubert, and J. Fedrowitz, Practical Applications, ser. Frontiers in Biosensorics. Basel: Birkhäuser, 1997, vol. 2.

[8] J. F. Liang, Y. T. Li, and V. C. Yang, "Biomedical application of immobilized enzymes," J. Pharm. Sci., vol. 89, no. 8, 2000, pp. 979–990.

[9] D. Yu, B. Blankert, J.-C. Virè, and J.-M. Kauffmann, "Biosensors in drug discovery and drug analysis," Anal. Lett., vol. 38, no. 11, 2005, pp. 1687–1701.

[10] E.-H. Yoo and S.-Y. Lee, "Glucose biosensors: an overview of use in clinical practice," Sensors, vol. 10, no. 5, 2010, pp. 4558–4576.

[11] J. I. Reyes De Corcuera, R. P. Cavalieri, J. R. Powers, and J. Tang, "Amperometric enzyme biosensor optimization using mathematical modeling," in Proceedings of the 2004 ASAE / Csae Annual International Meeting. Ottawa, Ontario: American Society of Agricultural Engineers, 8 2004.

[12] C. Amatore, A. Oleinick, I. Svir, N. da Mota, and L. Thouin, "Theoretical modeling and optimization of the detection performance: a new concept for electrochemical detection of proteins in microfluidic channels," Nonlinear Anal. Model. Control, vol. 11, no. 4, 2006, pp. 345–365.

[13] L. D. Mell and J. T. Maloy, "A model for the amperometric enzyme electrode obtained through digital simulation and applied to the immobilized glucose oxidase system," Anal. Chem., vol. 47, no. 2, 1975, pp. 299–307.

[14] J.-P. Kernevez, Enzyme Mathematics, ser. Studies in Mathematics and its Applications, J. L. Lions, G. Papanicolaou, and R. T. Rockafellar, Eds. Amsterdam: North-Holland Publishing Company, 1980, vol. 10.

[15] P. N. Bartlett and R. G. Whitaker, "Electrochemical imobilisation of enzymes: Part I. Theory," J. Electroanal. Chem., vol. 224, no. 1-2, 1987, pp. 27–35.

[16] D. Baronas, A. Žilinskas, and F. Ivanauskas, "Computational modelling and validation of a multilayer amperometric biosensor," in Proceedings of the XVIII International Master and PhD Students Conference Information Society and University Studies (IVUS 2013). Kaunas: Vytautas Magnus University, 4 2013, pp. 22–26.

[17] D. Šimelevičius, K. Petrauskas, R. Baronas, and J. Razumienė, "Computational modeling of mediator oxidation by oxygen in an amperometric glucose biosensor," Sensors, vol. 14, no. 2, 2014, pp. 2578–2594.

[18] A. J. Bard and L. R. Faulkner, Electrochemical Methods: Fundamentals and Applications, 2nd ed. New York: John Wiley & Sons, Inc., 2001.

[19] J. Razumiene, A. Vilkanauskyte, V. Gureviciene, J. Barkauskas, R. Meskys, and V. Laurinavicius, "Direct electron transfer between PQQ dependent glucose dehydrogenases and carbon electrodes: An approach for electrochemical biosensors," Electrochim. Acta, vol. 51, no. 24, 2006, pp. 5150–5156.

[20] V. Laurinavicius, B. Kurtinaitiene, V. Liauksminas, A. Ramanavicius, R. Meskys, R. Rudomanskis, T. Skotheim, and L. Boguslavsky, "Oxygen insensitive glucose biosensor based on PQQ-dependent glucose dehydrogenase," Anal. Lett., vol. 32, no. 2, 1999, pp. 299–316.

[21] J. Kulys and A. Malinauskas, "Electrochemical reduction-oxidation of phenazine methosulfate," Lietuvos TSR Mokslu Akademijos darbai, vol. B4, 1979, pp. 41–47.

[22] T. Schulmeister, "Mathematical modelling of the dynamic behaviour of amperometric enzyme electrodes," Sel. Electrode Rev., vol. 12, 1990, pp. 203–260.

[23] R. Baronas, F. Ivanauskas, and J. Kulys, Mathematical Modeling of Biosensors, ser. Springer Series on Chemical Sensors and Biosensors, G. Urban, Ed. Dordrecht: Springer, 2010, vol. 9.

[24] H. Gutfreund, Kinetics for the Life Sciences: Receptors, Transmitters and Catalysts. Cambridge: Cambridge University Press, 1995.

[25] K. J. Vetter, Electrochemical Kinetics: Theoretical and Experimental Aspects. Academic Press, 1967.

[26] D. Britz, Digital Simulation in Electrochemistry, 3rd ed., ser. Lecture Notes in Physics. Berlin Heidelberg: Springer, 2005, vol. 666.

[27] A. A. Samarskii, The Theory of Difference Schemes. New York-Basel: Marcel Dekker, 2001.

[28] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes in C++: The Art of Scientific Computing, 3rd ed. Cambridge: Cambridge University Press, 2002.

[29] A. C. F. Ribeiro, O. Ortona, S. M. N. Simões, C. I. A. V. Santos, P. M. R. A. Prazeres, A. J. M. Valente, V. M. M. Lobo, and H. D. Burrows, “Binary mutual diffusion coefficients of aqueous solutions of sucrose, lactose, glucose, and fructose in the temperature range from (298.15 to 328.15) K,” J. Chem. Eng. Data, vol. 51, no. 5, 2006, pp. 1836–1840.

[30] B. Strehlitz, B. Gründig, W. Schumacher, P. M. H. Kroneck, K.-D. Vorlop, and H. Kotte, “A nitrite sensor based on a highly sensitive nitrite reductase mediator-coupled amperometric detection,” Anal. Chem., vol. 68, no. 5, 1996, pp. 807–816.

[31] J. Kulys, L. Tetianec, and I. Bratkovskaja, “Pyrroloquinoline quinone-dependent carbohydrate dehydrogenase: activity enhancement and the role of artificial electron acceptors,” Biotechnol. J., vol. 5, no. 8, 2010, pp. 822–828.

[32] C. D. Jaeger and A. J. Bard, “Electrochemical behavior of donor-tetracyanoquinodimethane electrodes in aqueous media,” Journal of the American Chemical Society, vol. 102, no. 17, 1980, pp. 5435–5442.

Concept for Geopolitical Crisis Simulation as Assistance During a Decision Making Process

Maria Epp, André Brahmann, Uwe Chalupka, Oliver Kröning and Hendrik Rothe
Institute of Automation Technology,
Measurement and Information Technology
Helmut-Schmidt-University
Hamburg, Germany
emails: {maria.epp, andre.brahmann, uwe.chalupka, oliver.kroening, rothe}@hsu-hh.de

Abstract — **In the aftermath of September 11, the classical term of security is not longer valid and has to be extended. The security forces and decisions makers are faced with new challenges posed by the phenomena of terrorism, piracy, climate change and new economic practical constraints. To cope with this task, they need some support. This paper describes a possible simulation concept of geopolitical crises. As part of the concept the theory of finite element method should be examined on practicability. The starting point is a hypothetical scenario of a fictitious country. Infrastructural, cultural and socio-political aspects will be taken into consideration for the prediction of future crisis developments.**

Keywords-FEM; crisis simulation; crisis management

I. INTRODUCTION

A. *Definition of a Crisis*

To simulate a crisis, one has to be able to understand and explain a crisis at first. In literature, there is no general definition of a “crisis” available; therefore, we will have to give an explanation of the crisis term to make sure, that a common understanding is guaranteed.

In the beginning of the 18th century, a “crisis” was based on the probability of revolutions, counter-revolutions, independence and civil wars [1]. Given the rapid development of peace research, this term was stretched frequently. The changing international environment assigns both stakeholders of the security (i.e., states, international / national organizations, etc.), as well as political decision makers faced with new problems and requirements. Therefore, it is important to derive a new common definition.

A “crisis” is a situation, in which a system or a part of a system is extinguished and destroyed. A system is a group of actors (states, organizations, resources, etc.) that interact together in one or more fixed patters and structures [2]. This increases the likelihood of a war or the usage of force [3]. Furthermore, if a decision-making process is subjected to time pressure to avoid a growing danger of hopelessness [4], a crisis is about to rise.

B. *Crisis Management*

In time of a crisis, citizens need the aid of their leaders. In other words, they are dependent on the leadership of their presidents, local politicians, public managers or religious mentors. These leaders must make decisions to minimize the damage of the crisis and they have to examine ways that will lead their people out of the critical situation. The management of a crisis is often a complex operation, which requires a lot of organization. The decision makers must supervise the whole crisis management process, communicate with stakeholders, discover what went wrong, define possible risks and they have to initiate improvements to overcome the crisis.

To make these important decisions, leaders must have an overview of the possible progress of a crisis. To support the decision making process, a tool for crisis simulation could be of immense value. Such a tool could be used to forecast upcoming events. Consequently, this will provide an edge to the decision makers in managing the crisis.

This paper will start with an introduction of some basic ideas for a simulation that has to be taken into consideration. After that, two basic principles, that seem appropriate for the simulation, will be shown and discussed. The next section will outline a possible simulation prototype to clarify a first simulation concept. Finally, a conclusion is drawn and an outline for further steps is given.

C. *State of the Art*

1) *CASCON*

Computer-Aided System for Handling Information on Local Conflicts (CASCON) is a computerized history-based conflict analysis and decision-support system. This system is based on Bloomfield-Leiss model [6] of local conflict, which represents conflict as a dynamic process that moves in time and space [5]. A local conflict is derived from the substantive dispute. The dispute can be caused because of the territory, ideology, power, race, religion, or whatever. It is derived in phases of varying durations. In each phase, there are factors-conditions, perceptions, situations, or relationships that generate conflict-relevant pressure.

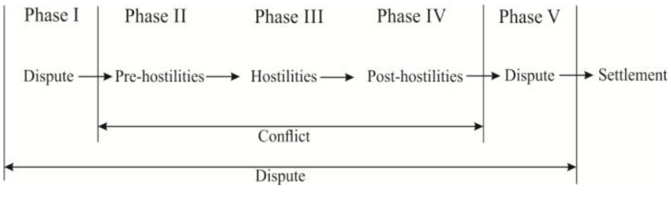
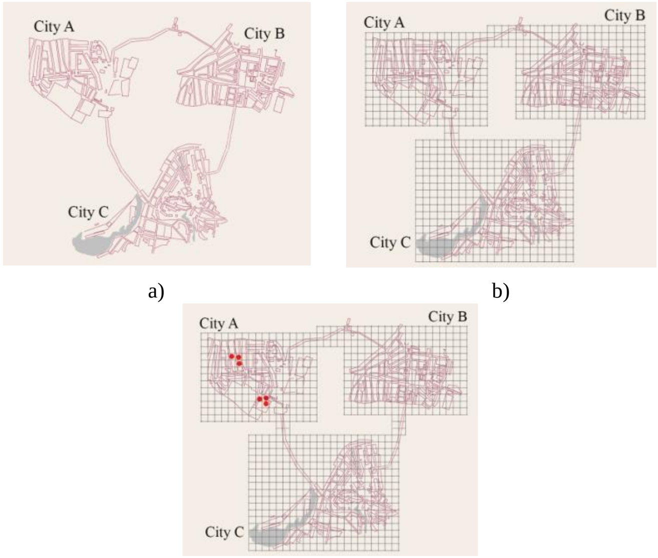


Figure 1. Phases of the Local Conflict Model [5].



a) b)

c)

Figure 2. a) A map of fictitious state. b) A map of fictitious state with a grid. c) Crisis points marked red in the map of fictitious state.

Figure 1 shows the phases of the local conflict based on the Bloomfield-Leiss model. Phase I defines a dispute that will be waged at the polls, in the press, economically, politically without military participation. Phase II defines a conflict (pre-hostilities) with an option of military introduction. Phase III, hostilities, arises when one side actually inserts military force to resolve the dispute. When hostilities are terminated, it skips to phase IV, post-hostilities. Phase V is entered when the dispute is no longer vied in military terms. The conflict is settled, when the dispute ends [5][6].

The users of the CASCON system can access the database. They can record the data about a new case or they can get the information about previous conflicts in a particular phase. The system can recommend steps at the onset of the dispute rather than at the outbreak of hostilities.

This system is very complex. With our simulation, we want to visualize possible spread of the conflict in a region. Actually, we do not need a database with historical data, but rather certain parameters, they must be defined later.

2) Plague Inc.

Plaque Incorporated is a strategic game that is developed by Ndemic Creations [7]. The aim of this game is to destroy the world with a deadly pathogen. The game uses a complex epidemic model to implement a realistic simulation.

Gamers play against the world's population. The earth is modeled as a game map. The map is divided into different states. The states have at its real existing models adapted factors, which can be divided into four areas: economic power, rainfall, temperature, and population density.

The game begins at the current day and then runs on a daily basis. The gamer can select a country, where the pathogen begins. First person will be infected. People can be healthy, infected, and dead. The disease spreads through country infection, ships, aircraft or animals.

If the world's population discovers the disease, it begins with the development of a cure. The more severe the medical condition is, the more money must be invested in the remedy. Economically powerful nations have an advantage for fighting the pathogen. A gamer has possibilities to fight back against the cure, such as the development of a resistance or the formation of a new disease. The player wins, when the world population is completely annihilated before a cure was cultivated. The world wins; if the world population develops a cure or kills the pathogen and support before all people infected [8].

This game would be a good template for a simulation of crisis spreading. But the project is commercial; there is no access to the methods and models that have been applied on the development of this game.

In our research project, we want to develop a crisis simulation based on "Plaque Inc.". We need to examine models and methods, which are descriptive of the crisis spreading more realistic. We want use the CASCON system, to define the influencing factor of crisis spreading. With the aid of Finite-Element-Method (FEM) [9], we want to present the geographic extension of a region, because this method is parallelizable and well scalable in computer power [10].

II. Basic Idea of Simulation

For our investigation, we will use a fictitious country. In Figure 2a, one can see that country, which consists of three cities. Furthermore, the cities are connected via infrastructural objects (i.e., roads). At that point, the development of one or more crises, which were triggered in one or more parts of a city, shall be examined.

To analyze the dispersion of a crisis from one point to another, a grid is placed on the cities and their compounds. The grid can be viewed in Figure 2b. Points that will represent a crisis, are placed on one or more grid locations. In Figure 2c, you can see the crises points marked as red dots. Now, for each part of the grid we could define some characteristic traits. These traits can be used to simulate infrastructure, regional and political states in that particular location. The main idea to model the interaction of every trait with a crisis point. To achieve this, we want to use differential equations. At the next step our work; we want to define these equations. The type and order of the using equations are dependent on the kind of interaction and its complexity. This leads to a convenient way of describing the underlying dependencies. The variables of differential equations could be defined using the CASCON system by providing the single factors.

III. Basic Principles

A. Finite-Element-Method

Taking performance considerations into account, it would not be a good choice to consider the complete map for our simulation. It is possible that the largest part of the observed country is devoid of infrastructure or important objects (forest, grassland, mountains, etc.), but another part is of major interest and only that part is in the need of closer investigating. Hence, the map should be divided into areas. As stated in the previous section, each location has particular properties and a degree of influence on neighboring locations. The fundamental dependencies are simulated via differential equations.

At this point, FEM, as a numeric technique for solving differential and integral equations, could become handy. A region, in which we seek a solution for the underlying differential equations, is divided into a finite number of elements. For each of the elements FEM provides an approximating function. Function values at the boundaries of the elements are the approximated solution of the differential equations and hence the solution of the fundamental problem. The coefficients of the approximating functions are determined by the condition of the values of neighboring functions on the boundaries between the elements. These coefficients are expressed through the values of the functions in the elements [9][11]. At this point, it becomes clear that the basic idea of the simulation and the basic principles of FEM are strongly related. Therefore, it is of high interest to examine this approach as possible solving mechanism of this kind of simulation.

B. Game of Life

Besides FEM, Conways “Game of Life” [12] appears to be a well suited template for this kind of problem. The “Game of Life” is a grid based computer simulator that was invented by Moler [12]. The population spreads over an infinite two dimensional squared field (universe). The population evolves at discrete time steps and depending on the neighborhood they stay alive, die, or bring new life to adjacent locations in the next time step. Every location in the underlying two dimensional grids is called a cell and every cell has two states (population alive or dead).

The fundamental concept of "Game of Life" is based on a very simple set of rules:

1. A dead cell with three living neighbors will become alive in the next time step,
2. A living cell with less than two neighbors dies in the next time step,
3. A living cell with more than three neighbors dies in the next time step,
4. A living cell with two or three neighbors will stay alive.

Though this concept is rather plain, it shows some similarities to the featured idea for crisis simulation. Crises could be regarded as the population and our region of interest is subdivided into cells on a grid. A crisis could appear (cell becomes alive) and it could disappear (cell dies). At this point, it becomes obvious that the set of rules have to be more complex to model the fundamental behavior of a rising or falling crisis and that is the point where a connection between FEM and "Game of Life" could bring a more desirable modeling quality.

IV. Description A Prototype of the Crisis Simulation

In this section, a possible implementation of a simple prototype for the simulation concept is introduced. An arbitrary two dimensional matrix *M* is chosen as a representation of our gridded country map. A second two dimensional matrix *C* is chosen as a representation of the crisis population. It becomes clear, that the fundamental problem is subdivided into multiple layers and each layer has its own matrix. The elements of the matrix *M*, which represent the cities and their infrastructural interconnections, are filled with ones (1). Empty spaces are marked with zeros (0). The same is true for the matrix *C*, a living cell is marked with one and a dead one with zero. Now, rules in favor of our special "Game of Live" represented by Matrix *C* can be applied. For the prototype, the following rules have been prepared:

1) A cell with a living crisis in matrix *C* can spread out (i.e., bring life) to another cell, if the corresponding cell in matrix *M* is defined with a one.
2) A living crisis can cause a new crisis in neighboring (adjacent) cells only.
3) A living crisis can spread out, if at least one neighboring element is also defined as a living crisis.
4) An element can be defined as a living crisis, if this element is not a crisis yet (dead crisis).
5) If a living crisis did not spread to a neighboring element in the previous time step, then it will die.
6) For each element, which is not a living crisis, is made a connection to every neighboring crisis element. To each connection the value 0 or 1 is assigned and that value is random generated with a probability of 50%. If at least one connection has the value 1, then this element will become a living crisis in the preceding time step.
7) The simulation ends after a predetermined number of iteration steps or if all of crisis elements are dead.

V. Conclusion and future work

The crisis management in geopolitical, economical and humanitarian area is a very complex topic. All these areas are highly correlated to each other. To make future statements about the development of a crisis, all factors must be regarded individually and collectively. To determine the dependency factors or the factors that have an influence on a crisis, the collaboration with political, economical and sociological institutes is planned.

It must be analyzed, which factor may affect the spread of a crisis from one to the other area and how a reasonable weighting of each factor with influence could be accomplished. A probable approach for accomplishing that challenge is to examine real crises and their development in the past in existing countries or regions. That is also a mandatory assignment for a validation of the simulation system, to prove correctness of the underlying algorithms.

All political and economic aspects have to be taken into account and neighboring regions must be analyzed. Once the most important factors and parameters are defined, the results can be simulated.

At this moment, an implementation of FEM was neglected in the simulation. That kind of methods will provide a more sophisticated simulation model, but this is only of any value after an interdisciplinary investigation of the whole topic, which is planned for the next prototype.

At last, it must be pointed out that the described prototype is at a very early stage of development. It handles a simple cellular automaton simulation with yet non-justified assumptions only. For further investigations, a validation and a probable reconsideration of the given rules have to be made. Obviously, it is a plain two-dimensional approach that solely relies on geographical distances. To reflect modern world scenarios, a multi-dimensional approach, which could simulate mass transit and electronic communication, has to be taken into consideration.

REFERENCES

[1] M. Jaenicke, “Krisenbegriff und Krisenforschung,” Westdeutsche Verlag, 1973, pp. 11-25.

[2] C. F. Hermann, “Indikatoren internationaler politischer Krisen,” Westdeutsche Verlag, 1973, pp. 44–63.

[3] K. W. Deutsch, “Zum Verständnis von Krisen und politischen Revolutionen,” Westdeutsche Verlag, 1973, pp. 90-100.

[4] A. Boin, P. Hart, E. Stern, and B. Sundelius, “The Politics of Crisis Management: Public Leadership under Pressure,” Cambridge University Press, 2005, pp. 1-2.

[5] L. P. Bloomfield and R. Beattie, “Computers and Policy-Making: The Cascon Experiment,” The Journal of Conflict Resolution, vol. 15, 1971, pp. 33-46.

[6] A. C. Leiss and L. P. Bloomfield, “The Control of The Local Conflict: A design study on arms control and limited war in the developing areas”, U.S. Government Printing Office Washington, 1967, pp. 2-8.

[7] Ndemic Creations, “Plaque Inc.,” http://www.ndemiccreations.com/en/22-plague-inc, [retrieved 09.2014]

[8] C. Buffa, “Plague Inc. iPad Review,” Modojo, 2012 http://www.modojo.com/reviews/plague_inc, [retrieved: 09.2014].

[9] O. C. Zienkiewicz, and K. Morgan, “Finite Elements and Approximation,” A Wilea-Interscience Publication, 1983.

[10] R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski, “Parallel Processing and Applied Mathematics,” Springer-Verlag Heidelberg, 2010, pp. 299-302.

[11] O. C. Zienkiewicz, “The Finite Element Mathod in Engineering Science, ” McGraw-Hill-London, 1971.

[12] C. Moler, “Experiments with MATLAB,” Electronic edition published by MathWorks, Inc., http://www.mathworks.com/moler, [retrieved: 08.2014].

Probabilistic Prognosis of Societal Political Violence by Stochastic Simulation

Using Principal Component Analysis and Support Vector Machines

André Brahmann, Uwe Chalupka, Hendrik Rothe, Torsten Albrecht
Institute of Automation Technology,
Measurement and Information Technology
Helmut-Schmidt-University
Hamburg, Germany
Andre.Brahmann@hsu-hh.de, Uwe.Chalupka@hsu-hh.de, Hendrik.Rothe@hsu-hh.de, albrecto@hsu-hh.de

***Abstract*—The current paper deals with the probabilistic prognosis of societal political violence levels of countries in the context of crisis prevention. The baseline is formed by two freely available datasets, whose relevance gets clarified in the first part of the paper. From these, a classification and a prediction modeling problem can be derived for which a Principal Component Analysis together with Support Vector Machines (SVMs) can be utilized as useful methods. Different SVM kernel functions have been investigated. To further perform the prediction step, a statistic modeling approach has been chosen that includes the computation of occurrence probabilities by stochastic simulation.**

Keywords-stochastic simulation; Major Episodes of Political Violence; principal component analysis; classification; support vector machines.

I. Introduction

Within the United Nations, the European Union and the North Atlantic Treaty Organization (NATO) crisis management, conflict prevention, security of trade routes, humanitarian aid, reconstruction activities as well as international cooperation are some of the main foci nowadays [1][2]. Thus, it is of high value to determine and anticipate possible geographic regions of interest in time [1]. These kind of regions will be defined by the term “hotspots” in the current work and the question arises how to identify and predict them.

A good clue to determine possible hotspots is given by the *Major Episodes of Political Violence* (MEPV) dataset, which is provided by the *Center of Systemic Peace* [10]. In the context of the current work, it is used as reference to characterize hotspot regions, because of its special properties. It summarizes the various occurrences of violence due to intrastate (civil, ethnic) and interstate conflicts on country level and transforms them into normed warfare magnitudes as time series on a yearly basis [10]. Thus, it allows a continuous assessment and objective comparison for each year with the warfare magnitude being proportional to the hotspot level. Other datasets, similar with regards to content, list singular events instead, which makes it hard to perform an continuous analysis on a yearly basis.

Although the usage of this dataset leads to a very convenient way of determining possible hotspots, it causes two distinct issues. The first one is the lack of data for certain regions, so that it is impossible to achieve a sufficient covering of the entire world. The second issue is the unsatisfying feasibility to forecast upcoming hotspots due to insufficient information contained in the MEPV-data for this task.

For that reason, governance indicators provided by the *World Bank* [11] comprising hundreds of underlying societal political factors on a yearly basis are additionally used. These indicators include direct and indirect measures of violence [3]. This makes the dataset basically also applicable within the given scope of hotspot assessment. Thus, the question may arise why not to use these governance indicators as a complete substitution for the MEPV-dataset instead. The reason is, the governance indicators comprise a lot more information and thus span more than one dimension (see Section II.B). This introduces the problem, that one has to decide for which values in what combination a hotspot can be assumed. So, statistical methods and data mining techniques are applied to recognize patterns that are related to certain values in the MEPV-dataset. With this additional information, it is possible to overcome the two issues mentioned above.

In the present paper, the basic approach to connect the level of violence to a certain tuple of indicators is shown and examined to reconstruct missing MEPV-data. Furthermore, a stochastic approach to predict the development of societal political violence by stochastic simulation is presented. Combining the reconstruction and prediction methods, finally it will be shown how MEPV-data can be predicted including the declaration of probabilities. Basically, the applied methodologies are Principal Component Analysis (PCA) and Support Vector Machines (SVM). In the first part of the paper, these methods will be briefly summarized, their relevance clarified and finally explained in which way they are applied for the current problem.

While the presented forecast modeling itself is yet simple, it reflects the underlying methodology, which can be transferred to any higher quality forecasting model. The regarded time frame of the prognosis is targeted for a short term basis of up to five years.

In the next section, the used datasets will be described in more detail. Afterwards, it is explained how and why (at a first glance) a PCA is applied to the current problem. Then, the data classification process and its relevance will be ex-

ISBN: 978-1-61208-371-1

plained, comprised by a motivation and description for the use of SVMs. Finally, the stochastic simulation model will be presented and linked with the PCA and SVM methods. Due to the dependence of the SVM classification quality to the used kernel function [12], different kernels will be investigated for the described datasets.

II. Description of used Data- and Toolsets

A. MEPV-Data

The dataset is provided for free on an annual basis and lists cross-national, time-series data on interstate, societal, and communal warfare magnitude scores (independence, interstate, ethnic, and civil; violence and warfare). The value of interest here is the civil violence ("CIVTOT"-value), also later referred to as "MEPV-level". It is coded from zero to ten using a discrete scale, where zero means no violence and ten means extermination/annihilation [10]. In general, low levels of violence are smaller than four [9]. Currently values are available from 1946 to 2012 for a total of 167 countries [10].

B. Worldwide Governance Indicators

This dataset is distributed by the *World Bank* and comprises a tuple of six indicators as times-series for 215 countries: (1) Control of Corruption, (2) Government Effectiveness, (3) Political Stability, (4) Regulatory Quality, (5) Rule of Law and (6) Voice & Accountability. Each indicator ranges approximately from -2.5 (bad) to 2.5 (good). The most recent set covers a time span from 2002 to 2012 on a yearly basis. The indicators can also be obtained for free, see [11].

C. Toolset

Before any calculation and examination could be done, some decisions regarding the need of specials tools have to be made. For prototyping and for a later verification, all algorithms were implemented in *Matlab R2014a* [14]. It is of great value that *Matlab R2014a* provides convenient implementations of the PCA and SVMs in its statistics toolbox. The first prototype relies exclusively on these two modules. The rest is done using standard functions provided by *Matlab* itself.

Beside the convenience of a *Matlab* implementation, there are also some downsides. First of all, the SVM implementation in *Matlab* does not provide native multiclass classification. Secondly, the creation of a custom kernel modification turned out to be difficult, as not being well documented. Furthermore, it becomes very tedious to maintain the code if a project exceeds a certain size. For these reasons, the working algorithm was implemented in the programming language *C#*. Due to its object oriented paradigm, it is very well suited for projects dealing with structured data and provides a very convenient syntax. While there is no native implementation of an SVM or PCA available in this language, as a downside, mature third party frameworks are available however. For the current application, the *Accord.Net* framework [13] has been chosen. This framework provides a large variety of statistical modules, including multiclass SVM and PCA. The framework is available for free and Open-source. Furthermore, it provides a very useful interface, which allows a convenient way to create any type of a custom SVM kernel.

So, every calculation described in this paper could be reproduced using either *Matlab* or *C#* in conjunction with the *Accord.Net* framework.

III. Application of Principal Component Analysis and Transformation

A. Dimension Analysis

As a first step, the dimensionality of the information contained in the six governance indicators is verified. Therefore, a PCA as described in [1] is applied.

The PCA is basically an approach to transform a tuple of observations, which are potentially correlated, into a set of uncorrelated variables. The result of this transformation, which is essentially a translation and a rotation, are the principal components (PCs) of the given observations. For a useful application of the PCA, the data has to be centered and standardized in a preceding step. Furthermore, to ensure non-corrupt data containing no outliers, an outlier test like the *Wilks*'s test [7] has to be performed.

The next step is the investigation of a possible reduction of the dimension of space. For that stage, the values of all eigenvalues of the correlation matrix have to be examined. The higher a value, the higher is the variance of the related principal component and therefore, the importance for declaring the features of the entire dataset. In Figure 1, the sorted percentages of all eigenvalues are shown. As one can clearly see, the first three principal components explain more than 96% of all observed features in the dataset.

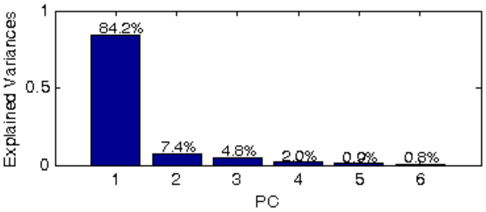


Figure 1. Percentage of all eigenvalues of the correlation matrix

As a first guess, it seems obvious to neglect the three inferior components. To ensure the correctness of this approach, a significance test with an error probability of 1 % on all the eigenvalues has been performed. Finally, this test certifies every eigenvalue as significant, so that a safe dimension reduction cannot be guaranteed.

B. Determining the Principal Components

Though proven with certain significance that there should be no dimension reduction performed, the results of the PCA are further used to compute the PCs of the indicators. The reason for that is their application in the data prediction step, as explained later.

For this, the eigenvectors of the previously calculated correlation matrix are computed. The eigenvectors should be sorted and scaled by the square root of their corresponding eigenvalue. When summarized into a matrix, a transformation matrix M is gained. The final transformation operation then is given by equation (1):

$$PC = Data \cdot M. \tag{1}$$

IV. Data Classification

A. Methodology

As a next step, it is required to develop a systematic scheme that allows to determine to which MEPV-level a certain indicator tuple for a single year and country corresponds. In Figure 2, scatterplots for each pair of the first three PCs are given as example, comprising the corresponding color coded classes in terms of MEPV-level.

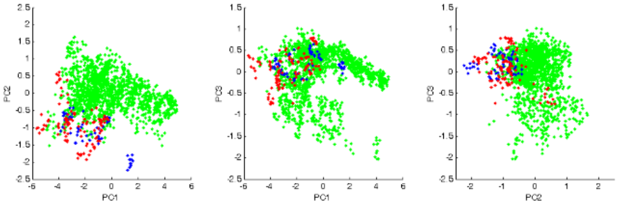


Figure 2. Scatterplots of the first three principal components (left: PC1 & PC2; middle: PC1 & PC3; right: PC2 & PC3) with the corresponding color coded classes: green represents no violence, blue represents a raised violence level and red represents hotspots.

As it can be seen the data seems to be spatially separable by adopting some classification method. In the next section, we take a closer look on possible classifications methods in the context of the given data.

B. Choosing the Classifier

As the simplest solution for solving the classification problem, the PCA could be used to discriminate the dataset. The drawback of using this method is that it can only be used for the classification of linear separable classes. A closer look at Figure 2 discloses that the underlying problem is not linear, therefore the PCA provides only a poor classification quality.

Fortunately, there are currently more advanced methods available. Discriminant analysis [16], neuronal networks (NNs) [15], and SVMs [6] are also known for solving classification problems.

With the discriminant analysis as a multivariate, statistical approach it is possible to classify via a linear, quadratic, *Bayesian* and *Mahalanobis* distance discriminant function. As with the statement above, a linear discrimination function is not appropriate. Choosing the *Mahalanobis* distance as classifier, one gets the best results for this particular problem. The classification error in this case is about 12 %. Although this means that we are getting correct answers in approximately 88 % of all cases, it is worth examining a more refined approach for that particular classification problem.

NNs might be a promising choice as being able to solve non-linear and also complex classification problems like pattern recognition in image processing. Nevertheless, they suffer from their theoretical limitations. For instance, when using back-propagation as learning method, they usually converge to locally optimal solutions. In this particular case, SVMs can offer a major improvement [5]. On top of that, by picking support vectors, SVMs choose the model size on their own [5] and they are developed using a sound theory first [5]. The lucid theoretical foundation of SVMs stands in contrast to the theoretical base of NNs.

SVMs are well explained in [6] and the following description uses this as a foundation. As the starting point for an SVM classification, we need a quantity of training objects whose correct classification is known (this is also true for the discriminant analysis mentioned earlier). An object could be regarded as a vector in a vector space. The SVM tries to fit a hyperplane into the given vector space, which separates the given classes. At the same time, the distance between the nearest vectors and the hyperplane is maximized and these vectors are called the support vectors.

It is apparent that a hyperplane is capable of discriminating linear separable vectors only. At this point, the SVM uses the fact that nonlinear separable vectors become linear separable in higher dimensional space. Therefore, a transformation into higher dimensions has to be performed and the hyperplane could be fitted into the training data set. After that process, the hyperplane must be transformed into the original space, which leads to a nonlinear hypersurface, which could be also non-contiguous.

This algorithm causes one serious issue, namely, the transformation into higher dimensions is computational expensive; therefore, an alternative approach was invented. A method called the "kernel trick" solves this issue. The trick is to use an appropriate kernel function, which describes the hyperplane in higher dimensions. If applied, the for- and backward transformation into and from higher dimensional space can be achieved without computing it directly. Therefore, the art of using SVMs is to choose the correct kernel function, which is well suited for the underlying problem. That discussion will be postponed to Section VI.

With a well suited kernel function, it is possible to distinguish between all MEPV-levels with no error, representing a perfect classification. Anyway, a nearly perfect classification of the training data might cause the problem of an over-classification. This means that the trained classifier gets too specific and hence sensitive to minor changes. This might lead to too many misclassifications when performing the prediction step (this problem also applies to NNs).

So far, the discussion neglected the fact that the underlying problem is a multi-classification problem. The traditional approach is only suited to distinguish between two classes. To overcome this issue, the classification between c classes has to be subdivided into c classifications of two classes. Although this was not explicitly pointed out, it was taken into account, so that the given errors are valid for the generalized classification case of all MEPV-classes.

V. Data Prediction via Stochastic Simulation

A. Approach

As it is intended to perform a prognosis of the societal, political level of violence, a prediction of the development of the indicator data is required. Of course, there is no way to achieve this with reliable results, as the social political processes in countries are too complex.

The most simple prediction approach would be an extrapolation of the indicator data based upon a linear regression model not regarding side effects. Such could be spillover effects between bordering countries with similar cultural and political backgrounds as most recently observed during the *Arab Spring*. A simple extrapolation does not imply the most likely development of the indicators. To consider probabilities, the actually chosen approach is to perform a statistic modelling of the indicator development based upon observed values.

B. *Statistic Modelling*

The first step is to derive the indicators for a specific country by time to obtain their annual slopes. From the distribution of the slopes the likelihood can be obtained for which an indicator will develop into a certain range of directions by integrating the probability density function (PDF) of the underlying distribution. At the other hand, it is possible to compute ranges of slopes for given probabilities. This reflects the approach done for the current application. To formally cover the whole probability range, the underlying PDF is split by n quantiles into equally sized ranges of the probabilities $1/n$. For each quantile range, the corresponding n mean slopes are computed from the PDF. In other words, they represent the n cases how an indicator might develop with identical probabilities. Finally, this step is performed for all six indicators. The predicted indicator values are then constructed for a certain year by using their last known values and the computed slopes, resulting in a set of six by n-indicator-tuples. From these, a total of p possible cases can be constructed on how all indicators might develop. This results from a permutation of each computed possible development of each indicator with

$$p = n^6. \qquad (2)$$

One yet unconsidered issue is that the indicators develop partially dependent on each other. The performed permutation however is only valid if the indicators would be independent from each other. This condition can be met if the PCs of the indicators are used instead of the pure, untransformed ones. All the previously described steps remain the same for the usage of the PCs instead of the original data.

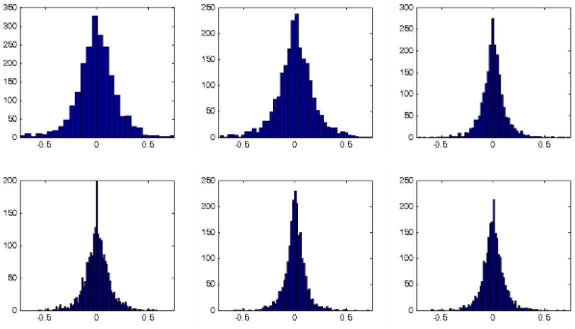


Figure 3. Histogram of the slopes of all PC values (first PC at upper left, sixth PC at lower right)

A remaining question is the type of PDF that is applicable to the current problem. For this, the complete distribution of the slopes for each PC of all countries have been investigated. It has been proven that the underlying PDF can be assumed to be a normal distribution. In Figure 3, the histograms of the slopes of the six PCs are given to exemplify that. The required sigma-value is obtained separately for each country by computing the standard deviation of the slopes of each PC for all years of a current country.

C. *Classification*

So far, p occurrences of indicator tuples per predicted year and country have been computed. Finally, of interest is the probable MEPV-level. Therefore, the classification methodology as described before by using SVMs can be applied. This requires that a SVM with an appropriate kernel is trained in advance by using a time span where both, indicator data and reference MEPV-data are available. The application of trained SVM to the predicted indicator tuples yields p MEPV-level occurrences per predicted year and country. The selection of a proper SVM kernel is addressed later in the next section.

As indicated by equation (2), the computational effort for classification dramatically increases if smaller sized quantile ranges are used. While for tertiles (spanning three ranges with each covering 33 % probability) 729 permutations occur, quintiles for example already produce 15625 permutations. These numbers then equal the count of indicator tuples which need to be classified for every predicted year and country.

Beside the option to avoid too small quantile ranges it is also possible not to include the less significant PCs (e.g., the 5th and 6th PC, see Figure 1) in the permutation process. Thereby, another advantage of using the PCs instead of the untransformed indicator data for classification is given.

TABLE I. MEPV PREDICTION, EXAMPLE OF CHAD

Year	**MEPV-Level Occurrences (Σ = 729)**					
	0	**1**	**2**	**3**	**5**	**4, 6..10**
2011	338 (46.4%)	102 (14.0%)	0 (0%)	289 (39.6%)	0 (0%)	0 (0%)
2012	561 (76.9%)	19 (2.6%)	2 (0.3%)	74 (10.2%)	73 (10.0%)	0 (0%)

In Table I, a prediction example for the country Chad is given for the years 2011 and 2012. The permutation has been performed by usage of all six indicators and probability ranges split by tertiles. The prediction was based on indicators from 2002 to 2010. Together with reference MEPV-data for the same time span a SVM has been trained using the preferred method and kernel function as described in the next section. The reference MEPV-levels for the years 2002 to 2004 have been zero, for 2005 one and for 2006 to 2010 three. The table shows that the predicted indicators of Chad tend to be classified primarily with MEPV-level zero and secondarily with MEPV-level three for the years 2011 and 2012. The real MEPV-levels for these years were zero.

ISBN: 978-1-61208-371-1

VI. Kernel Evaluation

As discussed before, an important aim is to find a well suited SVM kernel function to achieve a proper classification of indicator tuples in terms of MEPV-levels. For this, a separate optimization program has been written in C# instrumenting prior implemented prediction routines.

Five typical kernel functions have been selected for evaluation. Beside the two standard functions for linear and quadratic separation, the *Sigmoid*-, *Cauchy*- and *Hyperbolic Secant*-kernels have been chosen, based upon their principal properties, see [8][13]. These three non-standard kernel functions comprise scaling parameters, which have major influence on the classification results and thus need to be determined within a calibration process. The *Sigmoid*-kernel further represents the learning function of a two-layer-perceptron of a NN.

Because of the prior mentioned problem of over-classification, the kernels should not be evaluated by just using the training dataset as only reference to obtain the classification error. As an example, Table II shows evaluation results of the told kernels, trained with data from 2002 to 2010. The parameters for the three non-standard kernel functions have been optimized by minimizing the classification error of the training data. The table contains the classification errors and the standard deviations with respect to the reference MEPV-data for the years 2002 to 2010. The trained kernels have been used to classify the predicted indicator data for the years 2011 and 2012 using tertiles for the probability ranges of all six PC values. The gained results for both years, in form of the mean predicted MEPV-level for each country, are compared to the real MEPV-levels using their standard deviation. As it can be seen, with the *Cauchy*- and *Hyperbolic Secant*-Kernel it is possible to gain a 100 % fit for the training data, but the classification of the predicted data is worse than for the standard linear kernel. Using the quadratic kernel, the classifier cannot be trained at all.

TABLE II. SVM Kernel Performance
(Optimized for minimum misclassification of Training Dataset)

	Training (2002-2010)				
	linear	***quadratic***	***Sigmoid*** ***(0.05,-2.1)***	***Cauchy*** ***(0.0039)***	***Hyperbolic Secant (2.7)***
Class. Error	11.1 %	n/a[a]	10.0%	0%	0%
Std. Dev.	1.00	n/a[a]	0.98	0	0
	Classification of Prediction (2011-2012)				
Std. Dev.	0.85	n/a[a]	0.89	1.32	1.12

a. SVM training failed to find a solution

So, the goal must be to optimize the kernel function parameters by a minimization of the prediction error. The drawback of this kind of calibration is that the predicted data comprises additional prognosis uncertainties, influencing the accuracy of the found optimum. Due to a yet too small amount of available reference data, investigations on the required amount of years used for training and prediction with respect to the achieved overall prediction performance cannot be faithfully done so far.

In Table III, the results of using the method for minimizing the prediction errors are given. It can be seen that the *Cauchy*- and *Hyperbolic Secant*-kernels perform essentially better in the classification of the predicted values by conserving a small classification error of the training data. Especially the *Hyperbolic Secant*-kernel can be regarded as the kernel of choice for the current problem.

TABLE III. SVM Kernel Performance
(Optimized for minimum misclassification of Predicted Dataset)

	Training (2002-2010)				
	linear	***quadratic***	***Sigmoid*** ***(0.05, -2.4)***	***Cauchy*** ***(0.71)***	***Hyperbolic Secant (0.56)***
Class. Error	11.1%	n/a[a]	10.1%	4.3%	1.3%
Std. Dev.	1.00	n/a[a]	0.96	0.59	0.39
	Classification of Prediction (2011-2012)				
Std. Dev.	0.85	n/a[a]	0.84	0.63	0.61

a. SVM training failed to find a solution

As of the currently quite small data basis, the kernel functions might perform slightly different when data from future years will be available. Further, there even might exist some more suitable, especially custom kernels. Regarding the potentially unlimited function space, an exhausting evaluation is not feasible anyway.

VII. Short Term Prognosis Results

Applying the *Hyperbolic Secant*-Kernel with optimized kernel parameters, a one and a two year prognosis have been computed with respect to the last year of different time spans, which have been used for SVM training. The results are set in relation to the real values as mean and standard deviations through all countries and are listed in Table IV for each time span. Because data is currently available only until 2012, the last time span ends 2010 to remain reference data for up to two years.

TABLE IV. SummArized Results for MEPV-Level Prognosis as Deviations to Real Data

	2002-2005	***2002-2006***	***2002-2007***	***2002-2008***	***2002-2009***	***2002-2010***
+ 1 year	0.19 ±0.93	0.01 ±0.88	0.05 ±0.56	-0.04 ±0.76	0.01 ±0.47	0.06 ±0.59
+ 2 years	-0.16 ±1.03	0.07 ±1.07	0.11 ±0.71	-0.05 ±0.92	0.07 ±0.84	0.03 ±0.62

It can be seen that the mean deviations are all close to zero, while the standard deviations range between 0.5 and 1.0 MEPV-levels. This can be regarded as acceptable by remarking that longer time periods also tend to imply smaller deviations and thus produce more truthful results. Due to too little data, yet undeterminable is at which point the length of the time span used for training can be regarded as sufficient. It can also be observed, that there is only a little increase in the deviations comparing the two with the one year prognosis.

VIII. Conclusion

A method was presented to reconstruct probable societal political violence levels for countries which are not contained in the current MEPV-dataset using SVMs and world governance indicators. Further, the approach was extended to perform a prognosis of near future MEPV-levels. Therefore, a method has been described how to predict the underlying set of indicators by applying stochastic simulation and covering internal correlation effects using a PCA and a PC-transformation. While the forecast model itself is yet simple, as representing some kind of linear extrapolation, it provides an easy mechanism to obtain probabilities by stochastic modeling. As things usually do not develop just linear, a more sophisticated forecast model might provide more convenient results by e. g., also regarding interstate spillover effects which is not regarded so far. Finally, the current solution also delivers a reference model to test future models against.

An investigation of possible SVM kernels showed that the *Hyperbolic Secant*-kernel performed best by providing a classification error for data reconstruction and prediction with a standard deviation of about a half of an MEPV-level with respect to the real levels for the years 2011 and 2012.

As stated in the introduction, the targeted prediction time frame spans five years. Due to yet too few years for which data is available, only a time span of two years has been used for SVM kernel calibration and evaluation. In future, an extension to up to a five year time span is intended. For the same reason, an objective measure of the performance of the proposed method via a validation against past data is only limited possible.

Acknowledgment

Thanks are directed to the chair of *International Politics* from the *Department of Political Science, University of Federal Armed Forces* in Munich for providing guidance on underlying societal political aspects. Special thanks go to *Philipp Klüfers* who suggested the usage of the described datasets.

References

[1] Federal Ministry of Defence (Germany), White Book, 2006.

[2] NATO, "The Alliance’s Strategic Concept," Lisbon, 2010.

[3] D. Kaufmann, A. Kraay, and M. Mastruzzi, "The Worldwide Governance Indicators: Methodology and Analytical Issues," World Bank Policy Research Working Paper No. 5430, September 2010.

[4] J. Shlens, "A Tutorial on Principal Component Analysis," The Computing Research Repository, April 2014.

[5] M. Rychetsky, "Algorithms and Architectures for Machine Learning based on Regularized Neural Networks and Support Vector Approaches," Shaker Verlag GmbH Germany, December 2001.

[6] B. Schölkopf and A. J. Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)," The MIT Press, January 2002.

[7] S.S. Wilks, "Multivariate statistical outliers," Sankhya, Series A, 25, pp. 407–426, 1963.

[8] A. Chaudhuri, K. De, and D. Chatterjee, "A comparative study of kernels for the multi-class support vector machine," Fourth International Conference on Natural Computation, 2008, Vol. 2, pp. 3-7.

[9] O. J. de Groot, M. D. Rablen, and A. Shortland, "Gov-aargh-nance – “even criminals need law and order”, " CEDI Discussion Paper Series 11-01, Centre for Economic Development and Institutions, Brunel University, April 2011.

[10] Center for Systemic Peace, "Major Episodes of Political Violence, 1946-2013," http://www.systemicpeace.org, retrieved: July 2014.

[11] World Bank, "Worldwide Governance Indicators, 1996-2012" http://www.worldbank.org, retrieved: July 2014.

[12] Martin Sewell, "Support Vector Machines (SVMs), " http://www.svms.org/, retrieved: July 2014.

[13] C. R. Souza, “Accord.net framework,” 2013, http://accord-framework.net, retrieved: July 2014.

[14] The MathWorks Inc., “MATLAB R2014a”, Natick, MA, 2000.

[15] S. Samarasinghe, “Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition”, Auerbach Publications, 2006.

[16] G. McLachlan, “Discriminant Analysis and Statistical Pattern Recognition”, John Wiley & Sons, 2004.

Simulation Analysis for Performance Improvements of GNSS-based Positioning in a Road Environment

Nam-Hyeok Kim, Chi-Ho Park
IT Convergence Division
DGIST
Daegu, S. Korea
{nhkim, chpark}@dgist.ac.kr

Soon Ki Jung
School of Computer Science and Engineering
Kyungpook National University
Daegu, S. Korea
skjung@knu.ac.kr

***Abstract*— Global Navigation Satellite Systems (GNSSs), such as the Global Positioning System (GPS) in the USA, the GLObal NAvigation Satellite System (GLONASS) in Russia, and the Galileo in the EU, determine a target position using a satellite signal. They are widely used around the globe at this time. However, there is a critical obstacle when attempting to run a navigation system in a land vehicle. In contrast to aircraft or vessels, which operate in open areas without any obstacles, land vehicles must deal with signal occlusion caused by surrounding buildings, skyscrapers and other objects, especially in urban areas. In order to solve this problem, many researchers have studied many different methods, such as GPS/GLONASS-integrated positioning; pseudolite, which produces a signal similar to that of GPS; and GPS/Vision integrated positioning. These studies have mainly focused on integrated positioning methods. In contrast, this paper focuses on the relationship between the position of a new signal generator and positioning error for high-accuracy positioning in GPS shaded areas using simulation analysis. Through this analysis, we confirmed that horizontal positioning error is the lowest (10m) in the urban canyon when the degrees of geometric stability is the best.**

Keywords- GNSS; Vision; Pseudolite; Simulation.

I. INTRODUCTION

Global Navigation Satellite Systems (GNSSs), such as the Global Positioning System (GPS) in the USA, the GLObal NAvigation Satellite System (GLONASS) in Russia, and the Galileo in the EU, determine a target position using a satellite signal. At present, they are widely used around the globe. Since Selective Availability (SA) was released, the use of such systems has become prevalent in applications ranging from navigation systems for transportation to mobile smart phones. However, there is a critical obstacle when running the navigation system in land vehicles. In contrast to aircraft or vessels, which operate in open areas without any obstacles, a land vehicle must deal with signal occlusion caused by surrounding buildings, skyscrapers and other objects, especially in urban areas. Many researchers have attempted to solve this problem with various methods, such as GPS/GLONASS-integrated positioning [1], pseudolite, which produces a signal similar to that of GPS [2], and GPS/Vision integrated positioning [3]. These studies have mainly focused on integrated positioning methods with a new signal. In contrast, we focus on the relationship between the position of a new signal generator and positioning error for high-accuracy positioning in GPS shaded areas.

For this analysis, we developed a simulator using MATLAB, the configuration of which is described in section 2. The developed simulator generates GPS observation data with a variety of errors, such as ionospheric delays, tropospheric delays and clock errors. Moreover, the simulator filters some signals which are occluded by obstacles such as tall buildings.

Using this simulator, GPS positioning errors were analyzed in diverse road environments, such as housing areas and urban canyons. These results are described in section 3.

The simulator is also able to generate a new signal virtually and then perform integrated positioning using GPS and the new signal data. In section 4, the integrated positioning errors were analyzed according to the new signal generator's position.

Through this simulation analysis, we found that the accuracy of new signals and their degrees of geometric stability should be considered for highly accurate positioning.

This paper starts with the simulator description in section 2, then GPS positioning errors are analyzed in section 3. In section 4, the integrated positioning errors were analyzed. The conclusion of this paper is described in section 5.

II. SIMULATOR

We developed a simulator to perform an error analysis of GPS positioning and integrated positioning with a new signal to enhance the degree of positioning stability.

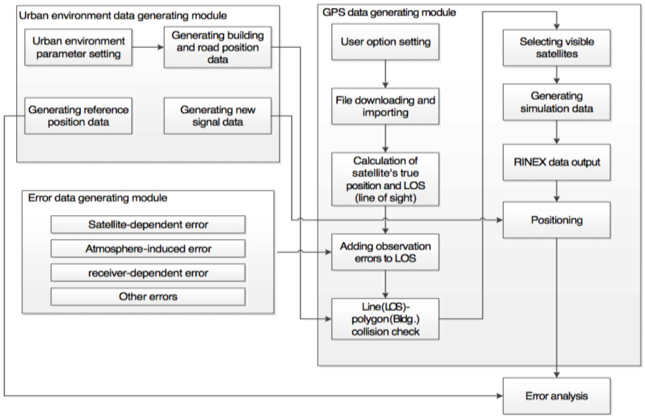
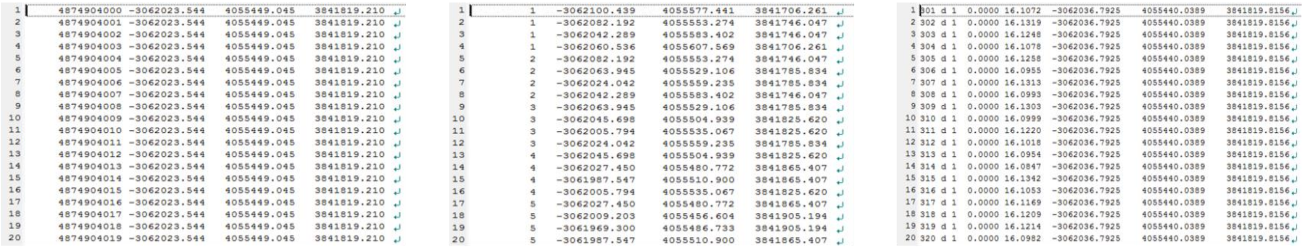

Urban environment data generating module
Urban environment parameter setting
Generating building and road position data
Generating reference position data
Generating new signal data
Error data generating module
Satellite-dependent error
Atmosphere-induced error
receiver-dependent error
Other errors
GPS data generating module
User option setting
File downloading and importing
Calculation of satellite's true position and LOS (line of sight)
Adding observation errors to LOS
Line(LOS)-polygon(Bldg.) collision check
Selecting visible satellites
Generating simulation data
RINEX data output
Positioning
Error analysis

Figure 1. Configuration of the simulator

ISBN: 978-1-61208-371-1



```
      4874904000 -3062023.544   4055449.045   3841819.210
      4874904001 -3062023.544   4055449.045   3841819.210
      4874904002 -3062023.544   4055449.045   3841819.210
      4874904003 -3062023.544   4055449.045   3841819.210
      4874904004 -3062023.544   4055449.045   3841819.210
      4874904005 -3062023.544   4055449.045   3841819.210
      4874904006 -3062023.544   4055449.045   3841819.210
      4874904007 -3062023.544   4055449.045   3841819.210
      4874904008 -3062023.544   4055449.045   3841819.210
      4874904009 -3062023.544   4055449.045   3841819.210
      4874904010 -3062023.544   4055449.045   3841819.210
      4874904011 -3062023.544   4055449.045   3841819.210
      4874904012 -3062023.544   4055449.045   3841819.210
      4874904013 -3062023.544   4055449.045   3841819.210
      4874904014 -3062023.544   4055449.045   3841819.210
      4874904015 -3062023.544   4055449.045   3841819.210
      4874904016 -3062023.544   4055449.045   3841819.210
      4874904017 -3062023.544   4055449.045   3841819.210
      4874904018 -3062023.544   4055449.045   3841819.210
      4874904019 -3062023.544   4055449.045   3841819.210
```

<Reference data>

```
        1    -3062100.439    4055577.441    3841706.261
        1    -3062082.192    4055553.274    3841746.047
        1    -3062042.289    4055583.402    3841746.047
        1    -3062060.536    4055607.569    3841706.261
        2    -3062082.192    4055553.274    3841746.047
        2    -3062063.945    4055529.106    3841785.834
        2    -3062024.042    4055559.235    3841785.834
        2    -3062042.289    4055583.402    3841746.047
        3    -3062063.945    4055529.106    3841785.834
        3    -3062045.698    4055504.939    3841825.620
        3    -3062005.794    4055535.067    3841825.620
        3    -3062024.042    4055559.235    3841785.834
        4    -3062045.698    4055504.939    3841825.620
        4    -3062027.450    4055480.772    3841865.407
        4    -3061987.547    4055510.900    3841865.407
        4    -3062005.794    4055535.067    3841825.620
        5    -3062027.450    4055480.772    3841865.407
        5    -3062009.203    4055456.604    3841905.194
        5    -3061969.300    4055486.733    3841905.194
        5    -3061987.547    4055510.900    3841865.407
```

<Building coordinate data>

```
301 d 1  0.0000 16.1072  -3062036.7925    4055440.0389    3841819.8156
302 d 1  0.0000 16.1319  -3062036.7925    4055440.0389    3841819.8156
303 d 1  0.0000 16.1248  -3062036.7925    4055440.0389    3841819.8156
304 d 1  0.0000 16.1078  -3062036.7925    4055440.0389    3841819.8156
305 d 1  0.0000 16.1258  -3062036.7925    4055440.0389    3841819.8156
306 d 1  0.0000 16.0955  -3062036.7925    4055440.0389    3841819.8156
307 d 1  0.0000 16.1313  -3062036.7925    4055440.0389    3841819.8156
308 d 1  0.0000 16.0993  -3062036.7925    4055440.0389    3841819.8156
309 d 1  0.0000 16.1303  -3062036.7925    4055440.0389    3841819.8156
310 d 1  0.0000 16.0999  -3062036.7925    4055440.0389    3841819.8156
311 d 1  0.0000 16.1220  -3062036.7925    4055440.0389    3841819.8156
312 d 1  0.0000 16.1018  -3062036.7925    4055440.0389    3841819.8156
313 d 1  0.0000 16.0954  -3062036.7925    4055440.0389    3841819.8156
314 d 1  0.0000 16.0847  -3062036.7925    4055440.0389    3841819.8156
315 d 1  0.0000 16.1342  -3062036.7925    4055440.0389    3841819.8156
316 d 1  0.0000 16.1053  -3062036.7925    4055440.0389    3841819.8156
317 d 1  0.0000 16.1169  -3062036.7925    4055440.0389    3841819.8156
318 d 1  0.0000 16.1209  -3062036.7925    4055440.0389    3841819.8156
319 d 1  0.0000 16.1214  -3062036.7925    4055440.0389    3841819.8156
320 d 1  0.0000 16.0982  -3062036.7925    4055440.0389    3841819.8156
```

<New signal data>

Figure 2. Output data example from the urban environment data-generating module

The configuration of the simulator consists of three modules [4]: the urban environment data-generating module, the GPS data-generating module, and the error data-generating module. Figure 1 shows the configuration of the simulator.

The urban environment data-generating module produces building coordinate data based on a reference position, the GPS data-generating module makes the observation data similar to actual GPS observation data, and the error data-generating module calculates observation errors based on models.

A. *Simulator configuration*

The input parameters – the observation time, receiver's reference position coordinates, the road width, the building height and the new signal generator interval for generating simulation data – are set in the urban environment generating module. According to the parameter settings, three-dimensional coordinates of virtual buildings, the new signal generator and the receiver position data (reference data) are generated with an Earth-Centered Earth-Fixed (ECEF) coordinate. Figure 2 shows an example of the output data from the urban environment data-generating module.

The GPS data-generating module generates the Receiver Independent Exchange Format (RINEX) as a GPS observation file and performs GPS positioning and integrated positioning. This module imports the reference data and extracts the observation time and receiver's position coordinates from the data. The satellites orbit information, clock data and other parameters are downloaded from the International GNSS Service (IGS) [5] site to determine the satellite's position and generate the error data. Next, this module calculates the satellite position at every epoch using this data. The Line-Of-Sight (LOS) is calculated based on the satellite and receiver position, and visible satellites are filtered at the receiver's position. Some errors are added to this LOS using the error data-generating module, and the final observation data are generated in the RINEX format after line (LOS)-polygon (building data) collision checking [6]. Using the GPS and new signal data or only GPS data, positioning is performed in this module. Stand-alone L1 Coarse/Acquisition (C/A) code positioning and an integrated positioning algorithm (the least-squares method) were used in this study, and the results are in ECEF coordinates [3].

The error data-generating module simulates errors related to GPS observations. These errors are classified into four types: satellite-dependent errors, atmosphere-induced errors, receiver-dependent errors, and other errors. Table 1 presents the details of the error modeling step. Each error can be modeled or calculated using a model and data files from the IGS and the Center for Orbit Determination in Europe (CODE) sites.

Finally, the positioning error is analyzed compared with the reference data and a plot of a related graph. For directional analysis, the positioning results are converted to North-East-Down (NED) coordinates.

TABLE I. SIMULATED MODELING ERROR

	Error	**Error model**
Satellite-dependent error	GPS orbit	Broadcast ephemerides (IGS orbit)
	Satellite clock error	Final clock file (IGS clock file)
Atmosphere-induced error	Ionospheric delay	IGS TEC (total electron content) map
	Tropospheric delay	Saastamoinen model [7], Chao mapping function [8]
Receiver error	Receiver clock error	Two-state random process model[9]
	Differential Code Bias (DCB)	CODE (center for orbit determination in Europe) DCB file
Other error	Random error	0.3m
	Relativity affecting the earth rotation	Sagnac effect

B. *Simulator verification*

For verification of the developed GPS simulator, a GPS observation file was generated and positioning was performed using C/A code data. The detailed settings are shown in Table 2.

TABLE II. DETAILED SIMULATION SETTINGS FOR VERIFICATION

Observation time	2013. 8. 1. 01:00:00～12:59:59 (12hours, 150 sec. interval, 288 epoch)
Receiver's position	Suwon continuously operating GPS/GNSS reference stations (ECEF coordinates:-3062023.544m, 4055449.045m, 3841819.210m)
Cut-off angle	15 degree
Adjustment computation model	Gauss-Markov model
Code random error	0.3m

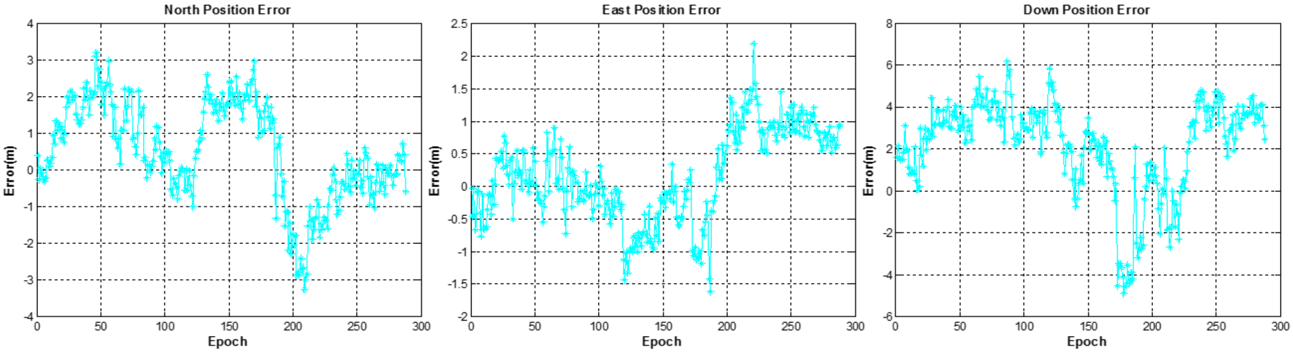


Figure 3. GPS positioning error for simulator verification

TABLE III. GPS POSITIONING ERROR FOR SIMULATOR VERIFICATION

	MAX	MIN	MEAN	STD	RMS
North(m)	3.22	-3.27	0.51	1.34	1.43
East(m)	2.19	-1.62	0.15	0.7	0.71
Down(m)	6.17	-4.9	2.17	2.23	3.12

Table 3 and Figure 3 show the error analysis by the simulator. Generally, a horizontal error in GPS positioning using C/A code data is 2-3 m and the vertical error is twice as much [10]. Through the error analysis result, the simulator is verified.

III. GPS POSITIONING ERROR ANALYSIS IN A DIVERSE ROAD ENVIRONMENT

Diverse road environments were formulated by the proposed simulator. The number of visible satellites, the number of estimated positions, the directional error, and other factors are analyzed in this section. Four types of simulation environments were set. CASE 1 assumed an open sky environment without any buildings as a reference for comparison with other scenarios. CASE 2 was set as a housing area that has two-story buildings (height 5 m) and a road width of 16 m. A commercial area was assumed in CASE 3, which has ten-story buildings (25 m) and a road width of 36 m. Finally, CASE 4 was set as an urban canyon that is surrounded by thirty-story buildings (height 75 m) and a road width of 68 m. For convenient analysis, some conditions were fixed in all cases: the road was assumed to run from south to north and the buildings were built next to the road (west and east) [4].

The above scenario's observation period was from 2013.09.01 00:00:00-11:59:59 and the time interval was 150 seconds. Since the orbital period of GPS satellites are about 12 hours, the observation time should be 12 hours for a reliable analysis. The receiver's position was identical to that of the Suwon continuously operating GPS/GNSS reference stations, which is the national reference station.

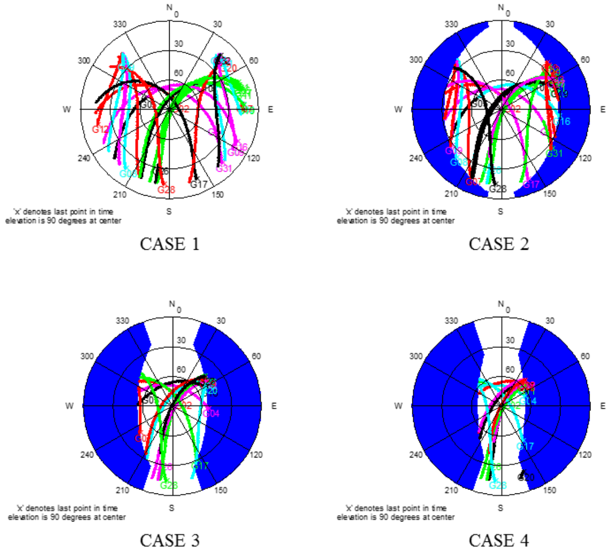
The results of the simulation analysis are as shown below.

TABLE IV. GPS POSITIONING ERROR ANALYSIS IN THE SIMULATED ENVIRONMENTS

		MAX	MIN	MEAN	STD	RMS
CASE 1	*North (m)*	3.39	-3.79	0.36	1.34	1.39
	East (m)	2.42	-0.99	0.56	0.63	0.84
	Down (m)	6.43	-9.71	1.97	2.83	3.45
	Number of estimated positions: 288 / 288					
CASE 2		**MAX**	**MIN**	**MEAN**	**STD**	**RMS**
	North (m)	8.89	-4.24	0.42	1.72	1.77
	East (m)	3.88	-1.12	0.68	0.91	1.13
	Down (m)	13.16	-36.71	1.19	6.28	6.39
	Number of estimated positions: 287 / 288					
CASE 3		**MAX**	**MIN**	**MEAN**	**STD**	**RMS**
	North (m)	10.83	-25.21	-1.31	6.74	6.87
	East (m)	27.67	-30.27	-0.87	9.87	9.9
	Down (m)	15.21	-62.66	-4.93	15.79	16.55
	Number of estimated positions: 159/ 288					
CASE 4		**MAX**	**MIN**	**MEAN**	**STD**	**RMS**
	North (m)	1328.4	-59.28	14.25	180.57	181.14
	East (m)	46.74	-1487	-20.6	201.22	202.29
	Down (m)	2314.6	-55.13	63.92	310.67	317.29
	Number of estimated positions: 56/ 288					

In CASE 2 (the housing area), the positioning error did not increase compared to CASE 1 (open sky). However, CASE 3 (the commercial area) and CASE 4 (the urban canyon) had greater positioning errors than CASE 1. In particular, the positioning of CASE 4 was virtually impossible with the number of estimated positions at 56 during the observation time

Figure 4 shows sky plots of all scenarios. The sky plots express the satellite's azimuth angle and elevation angle based on the receiver's position. The buildings that occlude the signals are illustrated in blue masking. The visible satellites of CASE 2 are similar to those of CASE 1. Satellites are visible above 60-degree elevation angles in the worst environment, which is CASE 4.



CASE 1

CASE 2

CASE 3

CASE 4

Figure 4. Sky plots for all of the scenarios

Figure 5 shows the number of satellites in the four cases.

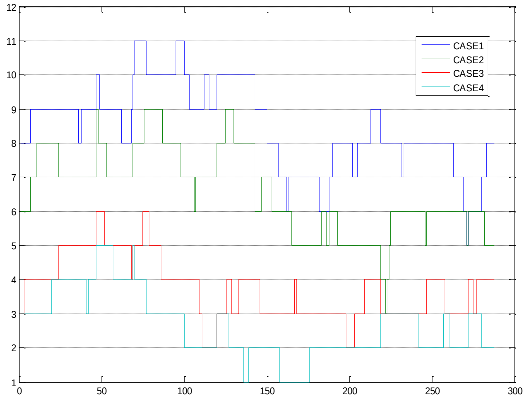


Figure 5. Number of visible satellites in each environment

The number of satellites ranges from 0 to 5 in CASE 4, indicating that the CASE 4 environment is the worst. Although the number of satellites occasionally exceeds four, it is less than four during most observation periods. Therefore, the position coordinate cannot be calculated and GPS positioning is useless in the urban canyon environment. Hence, new signals that offset GPS for stable positioning should be installed.

IV. Integrated Positioning Error Analysis in a Diverse Road Environment

Using the results from section 3, we discuss integrated positioning in this section. Unlike other studies, this study focused on the relationship between the signal generator's position and the signal's error instead of the integrated positioning algorithm. Pseudolite positioning and vision-integrated positioning both use distance data from the generator (landmark) to the receiver (camera) as observation values when using the least-square model. Therefore, the generator's placement or the landmark's placement has the same effect on the integrated positioning error. Hence, a new signal is used as a representative term.

Because CASE 4 is the worst environment, we assumed a situation in which the signal generator is installed in that environment. Under this assumption, an integrated positioning error analysis depending on the generator position was performed.

The position of the new signal generator was assumed to be on the building's roof with a height of 75 m, and it was installed from 10 m to 200 m at 10 m intervals. Because the minimum number of satellites is zero, four new signal generators needed to be installed for stable positioning. The position was set on both sides of the receiver. The signal had a 10% systemic error according to its distance, and random error of 0.3 m.

Because the total number of observation signals was always four in this case, the possibility of integrated positioning was 100% (288/288 epoch). Its directional error according to the generator's installation interval is shown in Figure 6.

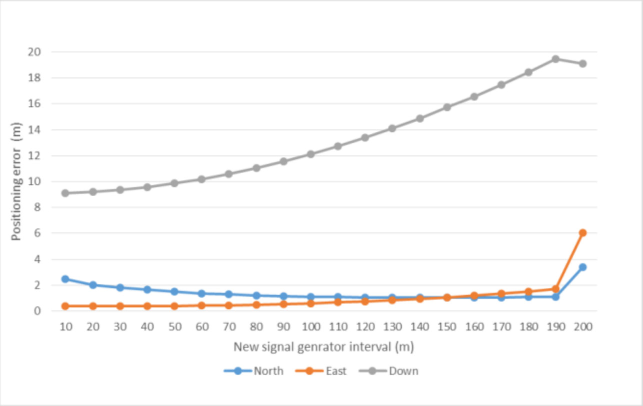
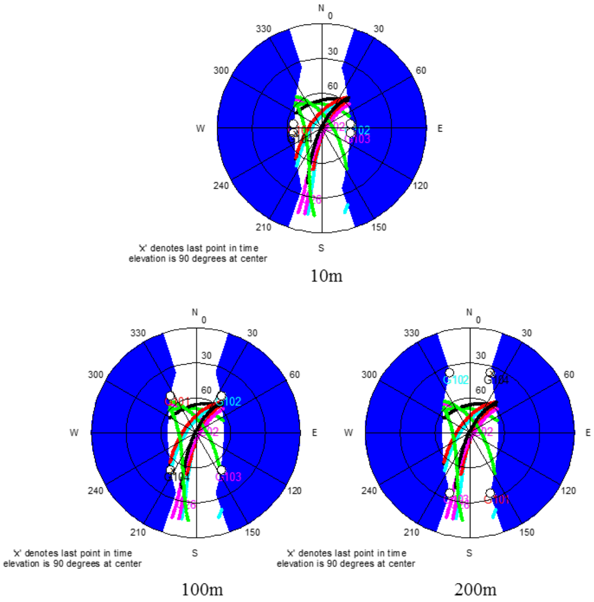


Figure 6. Directional error of the integrated positioning

The horizontal error was worse at the 10 m interval than the others, though the signal error was lowest. A 100 m interval resulted in the best performance in the simulation. This was caused by geometric stability; but the positioning error started increasing at the 200 m interval on account of the increase in the new signal error. This analysis is confirmed by Figure 7. Figure 7 shows a sky plot, with the generator positions of the new signals illustrated as white dots. This figure confirms that the geometrical placement of the interval at 10 m is very unstable. When the interval increases, the geometrical stability improves.



10m

100m

200m

Figure 7. A sky plot and the generator positions of new signals

Consequentially, the degree of geometrical stability and the amount of signal error should be considered at the same time during the installation of a new signal generator. This will guarantee the best positioning performance and stability.

V. CONCLUSION AND FUTURE WORK

In this paper, for GPS, which is a general GNSS, positioning errors were analyzed according to diverse environments for land vehicles using a custom-made simulator. GPS positioning was impossible in some epochs, or the errors were too large to use it in areas with buildings over ten stories. Especially in road environments surrounded by thirty-story buildings, it was almost impossible to calculate the position. In such areas, new signal generators were installed from 10 m to 200 m at 10 m intervals in a simulated environment, and the integrated positioning was performed using the new signals and GPS. The simulator generated the new signals with a 10% systemic error rate. Our results confirm that a 100 m interval gives the best performance in this type of simulation. This is due to the feasible geometric stability, but the positioning error started increasing at a 200 m interval on account of the new signal error.

Through this simulation analysis, we confirmed that the accuracy of signals and their degrees of geometric stability should be considered simultaneously when attempting to solve GPS shaded areas. The proposed simulator can be used in the planning step for solving systems in GPS shaded areas.

This developed simulator can also be used for analyses of multipath effects because simulation data does not have multipath errors despite the fact that it is used for general positioning error analysis. In the future, phase positioning errors will be analyzed by an upgraded simulator. This study will also be used to investigate diverse GPS environments and integrated positioning errors with new signals.

ACKNOWLEDGEMENT

This work was supported by the DGIST R&D Program of the Ministry of Science, ICT & Future Planning of Korea (14-IT-01).

REFERENCES

[1] H. S. Lee, K. D. Park, D. S. Kim, and D. H. Sohn, "Analysis of Integrated GPS and GLONASS Double Difference Relative Positioning Accuracy in the Simulation Environment with Lots of Signal Blockage", Journal of Navigation and Port Research, vol. 36, Aug. 2012, pp. 429-435.

[2] H. Wang, C. Zhai, X. Zhan, and Z. He, "Outdoor Navigation System Using Integrated GPS and Pseudolite Signals: Theoretical Analysis and Simulation", International Conference on Information and Automation, Jun. 2008, pp. 1127-1131.

[3] C. H. Park and N. H. Kim, "Precise and Reliable Positioning Based on the Integration of Navigation Satellite System and Vision System, International Journal of Automotive Technology, vol. 15, Feb. 2013, pp. 79-87.

[4] N. H. Kim, C. H. Park, S. K. Jung, and J. H. Han, "Simulation Analysis of GPS Positioning Accuracy Depending on the Urban Environment", The Korean GNSS Society Conference, Nov. 2013.

[5] International GNSS Service. IGS: IGS Products. [Online]. Available from: http://igscb.jpl.nasa.gov/components/prods.html, 2014.02.20.

[6] H. I. Kim, K. D. Park, and H. S. Lee, "Development and Validation of an Integrated GNSS Simulator Using 3D Spatial Information", Journal of the Korean Society of Surveying, vol. 27, Dec. 2009, pp. 659-667.

[7] J. Saastamoinen, “Contribution of the theory of atmospheric refraction”, B. Geod, pp. 105-106, 1972.

[8] C. C. Chao, “A Model for Tropospheric Calibration from Daily Surface and Radiosonde Balloon Measurements”, Technical Memorandum, Jet Propulsion Laboratory, pp. 391-350, 1972.

[9] B. W. Parkinson and J. J. Spilker, Global Positioning System: Theory and Applications, vol. 1, pp. 389-399, 1997.

[10] B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, Global Positioning System Theory and Practice, 5th ed., Springer, 2001.

Online Heat Pattern Estimation in a Shaft Furnace by Particle Filter Logic

Yoshinari Hashimoto, Kazuro Tsuda
Instrument and Control Engineering Dept.
JFE Steel Corp.,
Kawasaki, Japan
Emails: {y-hashimoto@jfe-steel.co.jp, k-tsuda@jfe-steel.co.jp}

***Abstract*— In steel-making plants, there are many processes, such as the blast furnace, in which the internal state is not directly observable. Automation of such processes based on process visualization is an urgent issue. Because the number of sensors is limited, the state estimation utilizing partial sensor information is necessary. We developed a technique which visualizes the whole temperature distribution of a shaft furnace by combining the sensor information and a nonlinear model calculation. Assuming that the difference between the model calculation and actual data derives from the fluctuation of unknown parameters which cannot be measured online, the parameters are estimated by a particle filter. This way, a robust state estimation logic was established. This technique was implemented and evaluated in a ferro-coke pilot plant at JFE Steel Corporation. As a result, estimation accuracy improved by 30 % compared with the model calculation without the state estimation.**

Keywords-Particle Filter; Data Assimilation.

I. INTRODUCTION

In steel making plants, there are many processes, in which the internal state cannot be measured directly. Such processes are operated manually depending on the operator's ability and experience. Hence, the automation based on the process visualization is an urgent issue.

There have been many approaches to the visualization of internal state by physical model calculation. For instance, complicated models which take into account fluid motion, reaction, and heat transfer have been developed [1]. However, because these models employ fixed parameters, they cannot deal with the transient phenomena caused by fluctuation of unknown parameters of materials characteristics, and so on.

In order to adapt to such situations, many studies have attempted to assimilate the model calculation with the partial information from the sensors, and compensate for modeling errors. Examples of conventional techniques are the Kalman filter [2], which is based on a linear approximation of the model, and the particle filter with simplified models based on a lumped element approximation [3][4]. In the field of process control, there have been a small number of studies that retain the feature of the model, which directly reflects nonlinear and complicated phenomena as a distributed element model, while making the best use of sensor information.

On the other hand, in the field of meteorology, there have been numerous studies of data assimilation in which large scale numerical simulations are assimilated with observation data. Data assimilation can be classified into sequential type and non-sequential type. The former includes the particle filter [5] and the ensemble Kalman filter [6]; an example of the latter is the adjoint method [7]. The former has the advantages that the implementation is relatively easy and probability distribution of the state variable can be obtained. In particular, the particle filter features robustness and a clear physical interpretation.

In this research, nonlinear models assuming various unknown parameters are calculated online, based on the concept of the particle filter. The weight of each model is updated with the degree of coincidence with the actual data, and the unknown parameters are estimated online. In this way, flexible modeling which can follow the plant change with a clear interpretation can be achieved.

The state estimation technique based on the particle filter as outlined above was implemented in a ferro-coke furnace at JFE steel. Ferro-coke is a mixture of coal and iron ore with the ratio of 7 : 3 [8]. Owing to the catalytic effect of metallic Fe, the coke gasification reaction starts at lower temperature, compared with normal coke [9]. This reduces the temperature of thermal reserve zone in blast furnace, enabling low coke ratio operation. The ferro-coke manufacturing process consists of mixing, molding, and coking. The target of this research is the heat pattern of a ferro-coke furnace during the coking process.

There are several constraints on the heat pattern of the furnace, such as the temperature rising rate, coking time, and cooling condition. For example, a higher rising rate enhances the fluidity of coal grain, resulting in better product strength. The holding time in coking zone should be controlled in order to improve the strength and reactivity.

In spite of the necessity of online control of heat pattern, the number of sensors is limited. In addition, unknown parameters exist in the process, such as the heat loss from the oven surface and the specific heat of the solid, which cannot be detected online. Therefore, a state estimation logic based on online parameter estimation with a particle filter is a proper approach.

The framework of this paper is as follows. In Section 2, the mathematical modeling of the ferro-coke furnace and the design of the particle filter are explained. A simulation study of state estimation is presented in Section 3. An evaluation of the estimation accuracy in the actual operation in a pilot plant is explained in Section 4. Finally, we conclude this paper in Section 5.

II. MODELING OF THE SHAFT FURNACE

A. Outline of the ferro-coke furnace

The structure of the ferro-coke furnace at the pilot plant is shown in Figure 1. Several tuyeres for coking and cooling are arranged symmetrically. There are three kinds of tuyeres, which are termed low temperature tuyere, hot tuyere, and cooling tuyere. The low temperature tuyere is used to adjust the temperature rising rate. The holding time in coking zone is achieved by the hot tuyere. The cooling tuyere and discharge device are installed at the bottom of the furnace.

Ferro-coke briquette are charged from the top of the furnace, and heated up by the heat exchange between the solid and the gas. After the coking process near the hot tuyere, the final product is released from the bottom.

Thermocouples TI (1)-TI (5) are arranged on the oven wall, which can monitor the temperature continuously. The thermocouples TI (6)-TI (8) are embedded in a probe, which is inserted in the furnace at appropriate times.

B. Mathematical modeling

For the visualization of temperature distribution, a transient 2D model was developed, which takes into account the reaction, the fluid motion, and heat transfer. The details of the model are as follows. The parameters are listed in Table 1. The coordinate x-y is defined in Figure 1. The material balance of the solid and the gas is presented as the continuity equation,

$$\frac{\partial u_g}{\partial x}+\frac{\partial v_g}{\partial y}=R, \tag{1}$$

$$\rho_s\frac{\partial v_s}{\partial y}=-R. \tag{2}$$

Here, two kinds of reactions were considered as listed in Table 2. One is the gasification of the volatile component of the coal, and the other is the reduction of the iron ore, and they were assumed to be irreversible reaction. The rate of the reactions was a function of solid temperature. For the momentum of the gas flow, Ergun's equation [1] was solved,

$$\left(G_1+G_2\left|\mathbf{u}_g\right|\right)u_g=-\frac{\partial p_g}{\partial x} \tag{3}$$

$$\left(G_1+G_2\left|\mathbf{u}_g\right|\right)v_g=-\frac{\partial p_g}{\partial y} \tag{4}$$

where

$$G_1=150\left(\frac{1-\varepsilon_g}{\varepsilon_g d_p}\right)^2\frac{\mu_g}{\varepsilon_g\rho_g},\quad G_2=1.75\left(\frac{1-\varepsilon_g}{\varepsilon_g d_p}\right)\frac{1}{\varepsilon_g{}^2\rho_g}.$$

In the case of the solid, the flow was simplified as a vertical descent, as described by (2). Next, the heat transfer model was developed to express the heat exchange between the solid and the gas, and the reaction heat. It is noteworthy that the time evolution term of (5) is negligible, because the heat capacity of the gas is much smaller than that of the solid.

$$\begin{aligned}&\rho_g C_g\frac{\partial T_g}{\partial t}+\frac{\partial(C_g u_g T_g)}{\partial x}+\frac{\partial(C_g v_g T_g)}{\partial y}\\&=\alpha\left(T_s-T_g\right)+R\Delta H_R\eta_1\end{aligned} \tag{5}$$

$$\rho_s C_s\frac{\partial T_s}{\partial t}+\frac{\partial(\rho_s C_s v_s T_s)}{\partial y}=\alpha\left(T_g-T_s\right)+R\Delta H_R\eta_2 \tag{6}$$

TABLE I. PARAMETERS IN THE MODEL

Symbol	Notes	Dimension
u_g	Mass velocity of the gas (Horizontal component)	$[kg/m^2\cdot s]$
v_g	Mass velocity of the gas (Vertical component)	$[kg/m^2\cdot s]$
ρ_s	Density of the solid	$[kg/m^3]$
v_s	Velocity of the solid	$[m/s]$
R	Reaction rate	$[kg/m^3\cdot s]$
p_g	Gas pressure	$[Pa]$
T_g	Gas temperature	$[K]$
T_s	Solid temperature	$[K]$
C_g	Specific heat of gas	$[J/kg\cdot K]$
C_s	Specific heat of solid	$[J/kg\cdot K]$
ΔH_R	Reaction heat	$[J/kg]$
η_1	Reaction heat distribution rate (gas)	$[-]$
η_2	Reaction heat distribution rate (solid)	$[-]$
T_{out}	Atmosphere temperature	$[K]$
h	Heat radiation coefficient	$[J/m^2\cdot s\cdot K]$
α	Heat exchange coefficient between gas and solid	$[J/m^3\cdot s\cdot K]$
ε_g	Void ratio	$[-]$
d_p	Diameter of the material	$[m]$
ρ_g	Density of gas	$[kg/m^3]$
μ_g	Viscosity of gas	$[Pa\cdot s]$
c_1	Parameter of reaction	$[-]$
c_2	Parameter of reaction	$[-]$

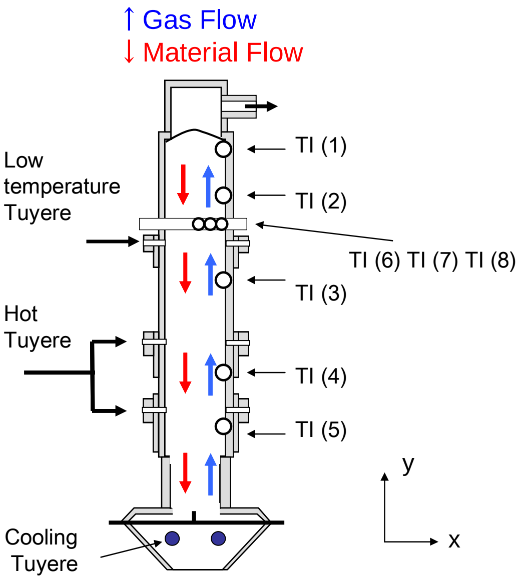


Figure 1. Structure of the ferro-coke furnace.

TABLE II. REACTIONS IN THE MODEL

Reaction	Notes	Dimension
$1/2\,\mathrm{Fe_2O_3} + 3/2\,\mathrm{CO}$ $\to \mathrm{Fe} + 3/2\,\mathrm{CO_2}$	$\dfrac{c_1}{1+\exp(-(T_s-600)/200)}$	$[kmol/m^3\cdot\sec]$
$\mathrm{Coal(s)} \to \mathrm{Coal(g)}$	$\dfrac{c_2}{1+\exp(-(T_s-400)/100)}$	$[kg/m^3\cdot\sec]$

The furnace wall was modeled as a boundary condition (7), where a heat exchange between the gas and the atmosphere occurs. This heat sink was incorporated in the gas heat calculation (5) as source term.

$$q = -h\left(T_g - T_{out}\right) \tag{7}$$

These differential equations are discretized by the finite volume method. The discretization scheme was Hybrid scheme [10]. The gas flow was solved by the SIMPLE algorithm, in which the pressure and the mass velocity are solved in a convergent calculation with (1), (3) and (4). Time marching was modeled by a first order implicit scheme. The time step was 10 minutes, considering the time-scale of the phenomenon in the furnace.

The discretized equations can be expressed in the form

$$\begin{aligned}&\left(\mathbf{T}_s(k+1),\mathbf{T}_g(k+1)\right)\\ &\quad= f\left(\mathbf{T}_s(k),\mathbf{T}_g(k),\mathbf{u}(k),\mathbf{A}(k)\right)\end{aligned} \tag{8}$$

where $\mathbf{T}_s(k)$, $\mathbf{T}_g(k)$ are the temperature distribution of the solid and the gas at time step k, respectively, $\mathbf{u}(k)$ is the model input, such as the inflow gas volume and temperature, and $\mathbf{A}(k)$ is the set of unknown parameters, which were assumed to fluctuate.

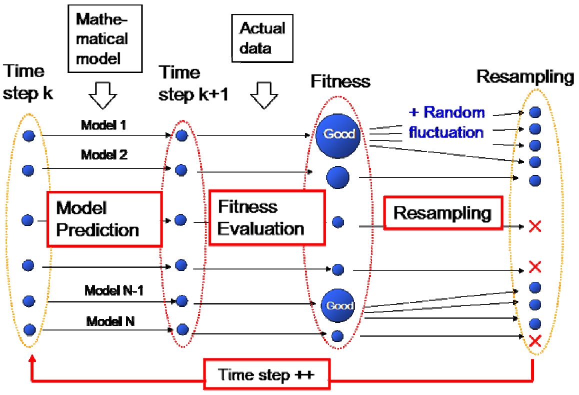


Figure 2. Algorithm of the particle filter.

Organizing the equations in this form is helpful when they are incorporated into the particle filter logic as described in Section 3.

C. Algorithm of particle filter

In this section, the algorithm of the particle filter, which assimilates the sensor information and model calculation, is explained. Summarizing the logic, various models assuming different parameter are prepared, and the temperature distribution is calculated in parallel. Then, the fitness of each model with the actual data is evaluated by the actual sensor data, and the number of copies of the model for the next time step is determined by the fitness. As shown in Figure 2, the procedure consisting of model prediction, evaluation, and making copies is repeated at every time step.

This way, the accuracy of the model is retained. The details of the algorithm consist of five steps, as follows.

Step (1): As the initial guess of the unknown parameters, various sets of the parameters are prepared. Here, we assume that the fluctuating parameters are the specific heat of the solid, and the heat radiation coefficient of the furnace wall. Hereafter, each set of the unknown parameters is to be called “particle”. The number of the particles was 25.

The weight of the particles was set to be equal, that is,

$$w_i(1) = 1/N \tag{9}$$

where $w_i(k)$ is the weight of i th particle at time step k, and N is the number of particles.

Step (2): Based on the unknown parameters of each particle, the temperature distribution is predicted using the transient model as shown in (8),

$$\begin{aligned}&\left(\mathbf{T}_{s,i}(k),\mathbf{T}_{g,i}(k)\right)\\ &\quad= f\left(\mathbf{T}_{s,i}(k-1),\mathbf{T}_{g,i}(k-1),\mathbf{u}(k-1),\mathbf{A}_i(k-1)\right)\end{aligned} \tag{10}$$

where $\mathbf{T}_{s,i}(k)$ is the temperature distribution of the solid at time step k, and $\mathbf{T}_{g,i}(k)$ is the temperature distribution of the solid at time step k, with respect to i th particle.

As this step is conducted independently for each particle, it is helpful to organize the transient model into the form as described in Section 2-*B*.

Step (3): The fitness between the actual data and the predicted temperature at the sensor locations is evaluated for each particle. First, the temperature prediction of the model of the i th particle at the sensor position can be obtained as,

$$\mathbf{y}_i(k) = \mathbf{C}\mathbf{T}_{g,i}(k) \tag{11}$$

where $\mathbf{C}$ is the observation matrix, which extracts the value at the sensor positions from the calculated temperature distribution. In this case, we assumed that the thermocouples measure the temperature of the gas, because the thermocouples were embedded in the wall, and there was no direct contact with the material.

The definition of the fitness was defined as

$$\theta_i(k) = \exp\left(-\frac{\left|\mathbf{y}_i(k) - \mathbf{y}_{act}(k)\right|^2}{\sigma^2}\right) \tag{12}$$

where the suffix *act* means the actual value from the sensors, and σ is the assumed observation error of the sensors.

Step (4): The weight of each particle is updated with the fitness, based on Bayes' theorem [5].

$$w_i(k) \leftarrow w_i(k)\theta_i(k) \tag{13}$$

$$w_i(k) \leftarrow \frac{w_i(k)}{\sum_i w_i(k)}. \tag{14}$$

The weight $w_i(k)$ is updated by multiplication by the fitness $\theta_i(k)$, and normalized so that the total weight is unity. The estimation of the unknown parameters and temperature distribution at time step *k* can be expressed as the weighted average of the particle.

Step (5): The copies of the particles are generated with the probability proportional to the weight of each particle. Thus, a particle with a larger weight has a greater chance of leaving more copies for the next time step. After that, the unknown parameters are slightly adjusted with random numbers to avoid making completely identical particles. In this case, an arbitrary number was selected from 0.995 to 1.005, and the unknown parameters were adjusted by multiplying the number. Then, the weight of each particle is set to be equal.

Online estimation of unknown parameters is possible by conducting Step (2)-Step (5) at each time step. Therefore, higher accuracy is achievable, compared with the calculation by fixed parameters.

III. SIMULATION RESULT

This section describes a simulation study of the assimilation logic, as discussed in Section 2-C. Here, so called twin experiment was carried out as shown in Figure 3. In the virtual plant, the unknown parameters were changed dynamically. Then, using the particle filter, the partial temperature information at the sensor positions in the virtual plant was assimilated with the model calculation without sensing the fluctuation of the parameters. The performance of the particle filter was checked by confirming that the change of the unknown parameters was identified properly, and that the temperature distribution of the virtual plant was estimated.

The unknown parameters are assumed to be the heat radiation coefficient of the oven surface and the specific heat of the solid. These parameters fluctuate in the virtual plant as shown in Figure 4.

The partial information of the temperature distribution at the sensor positions in Figure 1 was utilized as the input of the particle filter logic. The model input, such as the gas inflow temperature and volume are taken from the actual operation data of the pilot plant.

The estimation results of the surface temperature and the internal temperature are shown in Figure 5. The estimation with assimilation corresponds to the temperature visualization result by particle filter in Figure 3, and the estimation without assimilation means the mathematical model calculation without using sensor information in Figure 3. The temperature is scaled by the deviation from the set point. The temperatures of the virtual plant, the estimation result without the data assimilation, and the result with the assimilation are compared. The particle filter logic was set to be effective at time step 250. The upper two of the three rows are the results of the surface temperature, which are observable and used for the input of the particle filter. The bottom row is the result of the internal temperature, which is not observable. At all three positions, the estimated temperature gradually converges on the virtual plant data.

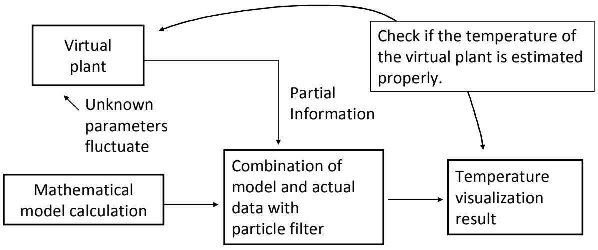
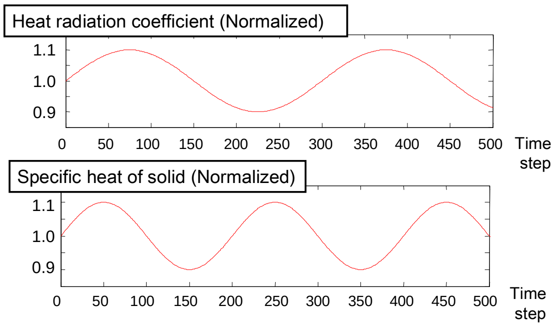

Virtual plant
Check if the temperature of the virtual plant is estimated properly.
Unknown parameters fluctuate
Partial Information
Mathematical model calculation
Combination of model and actual data with particle filter
Temperature visualization result

Figure 3. Validation of the particle filter by simulation.


Heat radiation coefficient (Normalized)

Specific heat of solid (Normalized)

Figure 4. Assumed unknown parameters change.

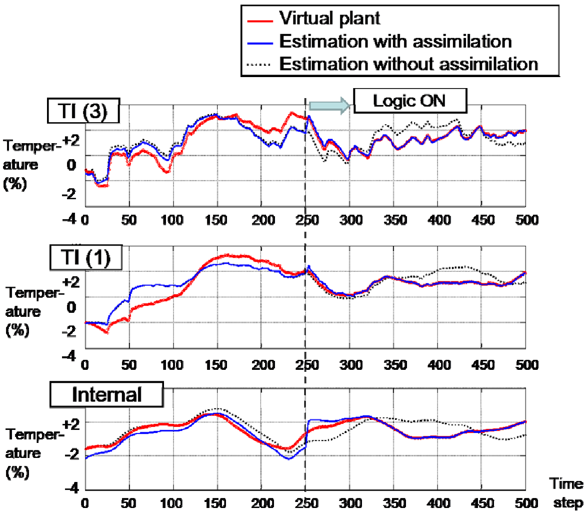


Figure 5. Estimation result of the temperature distribution.

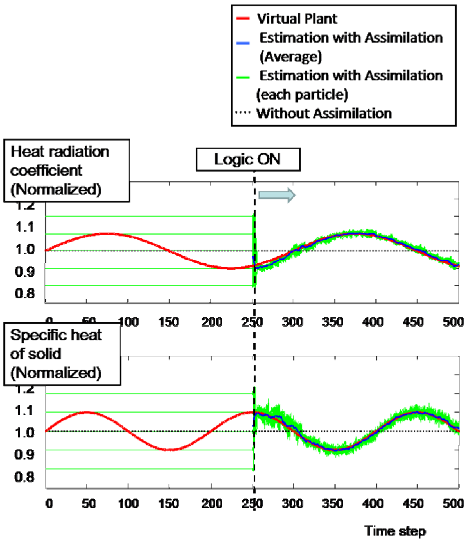


Figure 6. Result of parameter estimation by particle filter.

The result of parameter estimation is shown in Figure 6. The green lines show the result of 25 particles. It can be understood that each particle is searching the unknown parameters around the true value after applying the particle filter.

In this way, it was confirmed that the identification of the unknown parameters, and the estimation of the internal temperature, which cannot be measured online, are achievable with the particle filter logic.

IV. Evaluation in the Actual Operation

In this section, the validation result of the state estimation logic at an actual plant is presented. The particle filter logic was implemented at the operation room of a ferro-coke plant at JFE steel. Four of the five thermocouples, which are TI (1), TI (3), TI (4), TI (5) in Figure 1, were used for the input of the particle filter. The remaining one, TI (2) was used for validation of the logic. As explained in Section 3, we assumed that the unknown parameters were heat radiation coefficient and specific heat of the solid. First, the estimation results of the thermocouples used as the input are shown in Figure 7. The temperature is scaled by the deviation from the set point. For the sake of comparison, this figure shows plots of not only the model calculation with the state estimation, but also the calculation without the assimilation logic, which was conducted offline. The estimation accuracy improved by more than 50 % on average by the assimilation.

Next, the result of the validation thermocouple is explained. This result is shown in Figure 8. Since estimation accuracy improved by 40%, an over-fitting problem does not occur in this case. The estimation result of the internal temperature, which are TI (6)-TI (8) in Figure 1, is presented in Figure 9, where the actual temperature was measured by the probe in Figure 1. The estimated temperature agreed well with the actual temperature. In this case, estimation accuracy improved by 30 %. In comparison with the case of surface temperature, the performance of the assimilation declines because of the phenomenon which cannot be expressed by the assumed unknown factors, such as the fluctuation of the reduction reaction rate. The reaction rate greatly depends on the coal grain size inside the Ferro-coke briquette which cannot be detected online. Another unknown factor is the heat exchange ratio between gas and solid, which is influenced by the shape of the briquette.

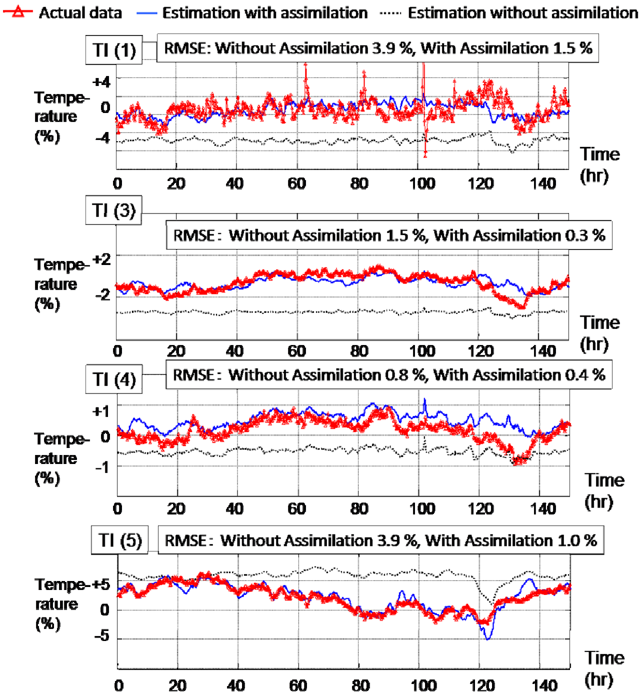


Figure 7. Estimation result of the temperature.

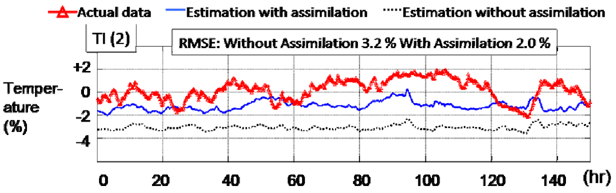


Figure 8. Temperature estimation at the validation point.

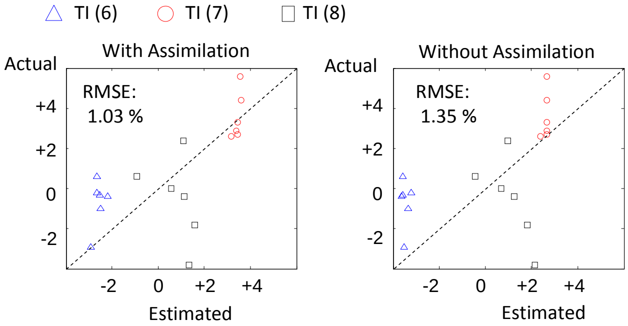


Figure 9. Estimation result of the internal temperature.

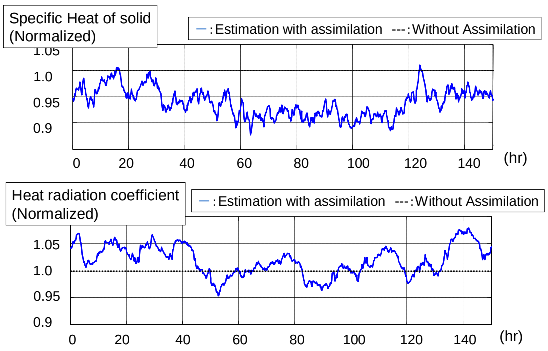


Figure 10. Estimation result of the unknown parameters.

The estimation result of the unknown parameters is shown in Figure 10. The blue line shows the estimation result, and the black dotted line shows the values of offline calculation with fixed parameters. The fluctuation of the specific heat reflects the type of material and the enthalpy of the reaction. The heat radiation coefficient changes due to the contact between the material and oven wall, and the condition of the oven surface.

In this way, the performance of the state estimation technique based on the particle filter was validated in the actual operation. The performance of the particle filter can be enhanced further by adding other unknown factors, such as heat exchange rate, reduction reaction rate, and so on. However, in such case, the number of the particles should be increased, and shortening the calculation time of the physical model is necessary. Considering the limit of the calculation time, identifying the unknown factors which mainly contribute to fluctuation of the temperature distribution in the actual plant is the key element for the practical application of this method.

V. CONCLUSION AND FUTURE WORK

In this research, online state estimation logic based on the particle filter logic was developed. An estimation logic of the whole temperature distribution of a shaft furnace was established by combining the partial sensor information and a nonlinear model calculation. The performance of the estimation logic was confirmed in a simulation study. As a result of an evaluation in actual operation, the estimation accuracy improved by more than 30 % compared with the model calculation without the state estimation.

As future work, the temperature control logic based on state estimation will be developed. The state estimation logic enables the control of inner temperature distribution of the furnace, which cannot be observed directly. Possible manipulated variables are the gas temperature or the gas volume of the tuyeres.

REFERENCES

[1] K. Takatani, T. Inada, and Y. Ujisawa, “Three - dimensional Dynamic Simulator for Blast Furnace,” ISIJ International, vol. 39, 1999, pp. 15-22.

[2] J. Brännbacka and H. Saxén, “Modeling the Liquid Levels in the Blast Furnace Hearth,” ISIJ International, vol. 41, 2001, pp. 1131-1138.

[3] S. Sonoda, N. Murata, H. Hino, H. Kitada, and M. Kano, “A Statistical Model for Predicting the Liquid Steel Temperature in Ladle and Tundish by Bootstrap Filter,” ISIJ International, vol. 52, 2012, pp. 1086-1091.

[4] N. Kaneko, S. Matsuzaki, M. Ito, H. Oogai, and K. Uchida, “Application of Improved Local Models of Large Scale Database-based Online Modeling to Prediction of Molten Iron Temperature of Blast Furnace,” ISIJ International, vol. 50, 2010, pp. 939-945.

[5] T. Higuchi, “Monte Carlo filter using the genetic algorithm operators,” Journal of Statistical Computation and Simulation, vol. 59, 1997, pp. 1-23.

[6] G. Ueno, T. Higuchi, T. Kagimoto, and N. Hirose, “Application of the ensemble Kalman filter and smoother to a coupled atmosphere-ocean model,” SOLA, vol. 3, 2007, pp. 5-8.

[7] M. Kano, S. Miyazaki, and K. Ito, “An adjoint data assimilation method for optimizing frictional parameters on the afterslip area,” Earth, Planets and Space, vol. 65, 2013, pp. 1575-1580.

[8] S. Nomura, H. Terashima, E. Sato, and M. Naito, “Some Fundamental Aspects of Highly Reactive Iron Coke Production,” Tetsu-to-Hagané, vol. 92, no. 12, 2006, pp. 849-855.

[9] S. Ueda, K. Yanagiya, K. Watanabe, T. Murakami, R. Inoue, and T. Ariyama, “Reaction Model and Reduction Behavior of Carbon Iron Ore Composite in Blast Furnace,” ISIJ International, vol. 49, no. 6, 2009, pp. 827–836.

[10] S.V.Patankar, Numerical Heat Transfer and Fluid Flow. Hemisphere: Washington DC/New York, 1980.

Weights Decision Analysis on the Integration of Navigation Satellite System and Vision System for Precise Positioning

Chi-Ho Park, Nam-Hyeok Kim
IT Convergence Division
DGIST
Daegu, S.Korea
{chpark, nhkim}@dgist.ac.

***Abstract*— In this paper, we propose a precise and reliable positioning method for solving common problems, such as a navigation satellite's signal occlusion in an urban canyon and the positioning error due to a limited number of visible navigation satellites. This is an integrated system of the navigation satellites system and a vision system. In general, the navigation satellite positioning system has a fatal weakness in that it cannot calculate a position coordinate when its signal is occluded by some obstacle. For this reason, positioning by the navigation satellites system cannot be used for a variety of applications. Therefore, we propose a method to integrate both the navigation satellites system and a vision system using weights decision analysis for precise positioning.**

Keywords-GNSS; Vision; Integration; Positioning.

I. INTRODUCTION

The Global Positioning System (GPS) was developed in the United States for military purposes. However, in the 1990s, after being opened to the private sector, it has become widely used for vehicle navigation, aircraft, communications, science, agriculture, and exploration. In addition, the Soviet Union's navigation satellite system GLONASS was also opened to the private sector and this has allowed the Global Navigation Satellite System (GNSS) to be utilized for more purposes. Recently, there have been a lot of studies involving GNSS applications [1]-[3]. The advantages of this system include improved position accuracy, convenience, continuity, continuous usability, integrity, and so on. The most important feature of GNSS is its frequency band allocation. Consultations regarding spectrum allocation have assigned the L1 band to 1575.42MHz, the L2 band to 1227.60MHz, and the L5 band to 1176.45MHz in the WRC2000. An increase in the satellites and multi-frequency have solved problems, such as poor accuracy and continuity, however rapidly developing industrialization has increased many of these problems. In particular, the increased position error due to cycle slip and multipath exacerbates these problems. Further, many areas of the earth outdoors are still in a GPS shadow. These factors are the main cause of decreased reliability and continuity [4].

Today’s navigation satellite system provides location information services within around 100 m (drms) accuracy at any time, regardless of the number of users. The system calculates a three-dimensional position by using triangulation, so it can be operated whenever the receiver can take signals from more than 4 satellites. However, locations in a city or in mountainous areas cannot always receive satellite signals because they may be blocked by high buildings or mountains. In such cases, the position error will be bigger or shaded areas will occur. Consequently, to solve these problems, many researchers have proposed techniques using the navigation satellite system and other applications [5]-[6].

One representative research is an integration system involving the navigation satellite system and an Inertial Measurement Unit (IMU). This system is a device used by countries to detect the location of their own submarines, aircraft, missiles, etc. and to drive to a targeted destination. The operating principle involves calculating the moving variance by using accelerometers, after determining cardinal points by gyroscope. The moving object can always calculate its current position and velocity after being input with the initial position. The advantage of IMU is that it is not affected by weather or jamming. However, when moving over a long distance, errors are accumulated. So, GPS correction is needed. In the case that signals are not received from satellites, the errors of the IMU are exponentially increased. Therefore, factors which cause shaded areas and obstacles limit the uses of kinematic positioning [7]. Due to rapid industrialization, the areas and environments where the navigation satellite system can be used is being reduced by the increasing growth of urban canyons. Fig. 1 shows a general situation for navigation satellite system signal reception in an urban canyon.

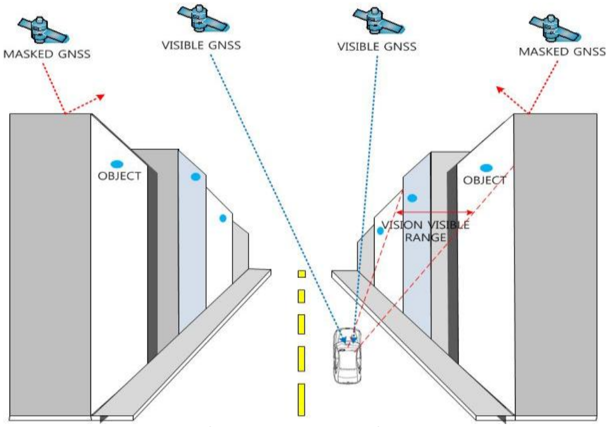


Figure 1. The situation for signal reception from the navigation satellite in an urban canyon.

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-371-1

Recently, a vision system was reported at a DARPA unmanned vehicle conference in the United States, which, in part, dealt with obstacle recognition and detection [8]-[9]. In particular, one team reported that they had used only a vision system for detection and recognition. But the vision system had some constraints, such as a real time processing problem due to the large amounts of data, and a recognition error due to external lighting changes.

Other recent researches using a vision system have made a lot of progress by improving the computer's performance and data processing time through the development of integration technology and the development of a broadband camera [10]-[11].

The advantages of the vision system are its wide range and long detection distance, the ease of data processing due to its similarity with the human visual system, and the provision of a variety of information. A stereo vision system that is capable of solving many of the problems of a monocular vision system uses a stereo matching algorithm for extracting a depth map, and an obstacle detection algorithm based on this depth map [12].

Therefore, in this paper, we propose a method to integrate both the navigation satellite system and the vision system using a weights decision analysis for precise positioning.

This paper explains the problems of the existing fusion research and the limitations of positioning using a satellite navigation system in the Introduction. In the main part it proposes an equation and description for the fusion algorithm of the satellite navigation system and vision system. The performance analysis presents a comparative experiment and analysis of the integration of the navigation satellite system and vision system for precise positioning using weights decision.

II. ALGORITHM

A. Positioning using a navigation satellite system

Usually, the position is obtained in point positioning mode by using a receiver chip that is mounted on the vehicle which receives the L1 C/A (Coarse/Acquisition) code from the navigation satellites. The C/A code observation equation for the navigation satellites system is given as follows.

$$P_{i,1}^k = \rho_i^k + T_i^k + \frac{I_i^k}{f_1^2} + c\left(dt_i - dt^k\right) + e_{i,1}^k$$
$$\rho_i^k = \sqrt{\left(x^k - x_i\right)^2 + \left(y^k - y_i\right)^2 + \left(z^k - z_i\right)^2} \qquad (1)$$

where, *i* and *k* denote receiver and satellite, respectively.

$P_{i,1}^k$: L1 C/A code pseudorange between the receiver and the satellite (m);

ρ_i^k : actual geometric distance between receiver and satellite (m);

T_i^k : tropospheric delay error (m);

$\frac{I_i^k}{f_1^2}$: ionosphereric delay error (m);

c : speed of light (m/s);

dt_i : receiver clock error (sec);

dt^k : satellite clock error (sec);

$e_{i,1}^k$: measurement error.

Ionospheric delay effects and satellite clock errors are removed by a navigation message from the satellites. Tropospheric delay effects are removed by models that account for the dry and wet refractivity at the surface of the Earth. Multipath error is not assumed. Inter-frequency bias is ignored because of its small value. As a consequence, (2) can be used for the observation equation to compute the receiver’s position in 3-dimensional space.

$$PG_{i,1}^k - T_i^k - \frac{I_i^k}{f_1^2} + cdt^k = \rho_i^k + cdt_i + e_{i,1}^k \qquad (2)$$

We denote by $P_{i,c}^k$ in the left side of (2) and linearize (2) because it is a non-linear equation. After that, the Gauss-Markov Model (GMM) [13] is applied. The result is (3). The satellites’ position coordinates are determined by using the navigation message. Unknown factors are the 3-dimensional position and receiver clock error.

$$y = A\xi + e, \qquad e \sim \left(0, \sigma_0^2 P^{-1}\right) \qquad (3)$$

Each item is shown below, $\dot{\rho}$ is calculated by the receiver's initial position $\left(\dot{x}_i, \dot{y}_i, \dot{z}_i\right)$.

$$y = \begin{bmatrix} P_{i,0}^k & - & P_{i,c}^k \\ P_{i,0}^l & - & P_{i,c}^l \\ & \vdots & \\ P_{i,0}^q & & P_{i,c}^q \end{bmatrix} \text{: Observation vector}$$

$$A = \begin{bmatrix} -\frac{x^k - \dot{x}_i}{\dot{\rho}_i^k} - \frac{y^k - \dot{y}_i}{\dot{\rho}_i^k} - \frac{z^k - \dot{z}_i}{\dot{\rho}_i^k} & c \\ -\frac{x^l - \dot{x}_i}{\dot{\rho}_i^l} - \frac{y^l - \dot{y}_i}{\dot{\rho}_i^l} - \frac{z^l - \dot{z}_i}{\dot{\rho}_i^l} & c \\ \vdots & \\ -\frac{x^q - \dot{x}_i}{\dot{\rho}_i^q} - \frac{y^q - \dot{y}_i}{\dot{\rho}_i^q} - \frac{z^q - \dot{z}_i}{\dot{\rho}_i^q} & c \end{bmatrix} \text{: Design matrix}$$

$$\xi = \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \\ dt_i \end{bmatrix} \text{: Unknown parameter vector}$$

$$e = \underbrace{\begin{bmatrix} e_i^k \\ e_i^l \\ \vdots \\ e_i^q \end{bmatrix}}_{n\times 1} \text{: Measurement error vector}$$

The unknown that is calculated in (3) is the increment with respect to the initial value.

$$\hat{\xi} = \underbrace{\left(A^T PA\right)^{-1}}_{N} \underbrace{A^T Py}_{c} \qquad (4)$$

The increment from (4) is added to the receiver's initial position and then the receiver's position is updated. This process is iterated until the increment is under the particular threshold value. After this process, the receiver's position is determined.

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_{update} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_{initial} + \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \end{bmatrix} \qquad (5)$$

The variance component can be computed by using (6). Also, the variance-covariance matrix for the estimates can be obtained using (7).

$$\hat{\sigma}_0^2 = \frac{\tilde{e}^T P \tilde{e}}{n - rkA} \qquad (6)$$

where, $\tilde{e} = y - A\hat{\xi}$, n is the number of observations.

$$D\left\{\hat{\xi}\right\} = \sigma_0^2 N^{-1} \qquad (7)$$

B. Fusion positioning equations

The vision system obtains observation values by recognizing objects. This means that the sizes of the observation vector and design matrix get larger as the number of observations increases. The distance from the receiver to the target object can be computed by using the vision system and the corresponding observation equation is as follows.

$$\begin{aligned} PV_i^a &= \rho_i^a + e_i^a \\ \rho_i^a &= \sqrt{\left(x^a - x_i\right)^2 + \left(y^a - y_i\right)^2 + \left(z^a - z_i\right)^2} \end{aligned} \qquad (8)$$

PV_i^a : distance estimated by the vision system from the specific object to the receiver (m).

ρ_i^a : actual distance from the specific object to the receiver.

x^a, y^a, z^a : three-dimensional position of a specific object.

x_i, y_i, z_i : three-dimensional position of the receiver.

After linearization, (9) can be rewritten as follows.

$$z_0 = K\xi + e_0, \qquad e_0 \sim \left(\sigma_0^2 P_0^{-1}\right) \qquad (9)$$

$$Z_0 = \underbrace{\left[PV_i^a - \rho_i^a\right]}_{1\times 1} \text{: Observation vector}$$

$$A = \begin{bmatrix} -\dfrac{x^a - \dot{x}_i}{\dot{\rho}_i^a} & -\dfrac{y^a - \dot{y}_i}{\dot{\rho}_i^a} & -\dfrac{z^a - \dot{z}_i}{\dot{\rho}_i^a} & 0 \end{bmatrix} \text{: Design matrix}$$

$$\xi = \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \\ dt_i \end{bmatrix} \text{: Unknown parameter vector}$$

The distance measurement from the vision system can be used as an additional observation and then the Gauss-Markov adjustment model with stochastic constraints is applied as shown in (10).

$$\hat{\xi} = \left(N + K^T P_0 K\right)^{-1} \left(c + K^T P_0 z_0\right) \qquad (10)$$

The residual of the distance estimated by the vision system is $e_0 = z_0 - K\xi$ and the estimated variance component is (11).

$$\hat{\sigma}_0^2 = \frac{\tilde{e}^T P \tilde{e} + \tilde{e}_0^T P_0 \tilde{e}_0}{n - m + l} \qquad (11)$$

Here, n is the observed number of navigation satellites, m is the number of unknown parameters (coordinates 3, receiver's clock error 1), l is the number of the distance measurement obtained from the vision system. Also, the variance-covariance matrix for the estimates can be computed by using (12).

$$D\left\{\hat{\xi}\right\} = \hat{\sigma}_0^2 \left(N + K^T P_0 K\right)^{-1} \qquad (12)$$

where, a is the specific object, i is a receiver, and each of the items are as follow.

PV_i^a : distance estimated by the vision system from the specific object to the receiver (m).

ρ_i^a : actual distance from the specific object to the receiver.

x^a, y^a, z^a : three-dimensional position of a specific object.

x_i, y_i, z_i : three-dimensional position of the receiver.

III. PERFORMANCE ANALYSIS

Experiments were conducted to evaluate the reliability and stability of positioning based on the integration of the navigation satellites system and the vision system. The receiver of the navigation satellite system was the DL-V3 Real Time Kinematic (RTK) of Novatel for the base station and the rover. The antenna was GPS-702-GGL of Novatel, the RF modem was the PDL rover kit of 450 MHz, and the Inertial Measurement Unit (IMU) was CG-5100 of KVH. Also, a stereo camera was used as the image sensor. The focal length of the image sensor was 12 mm and the baseline of the stereo camera (x coordinate difference between the two cameras) was 300 mm.

We compared the static and kinematic states of a vehicle in the experiments.

The following are results of experiments in the static state. Figure 2 shows the sky plot of the static state.

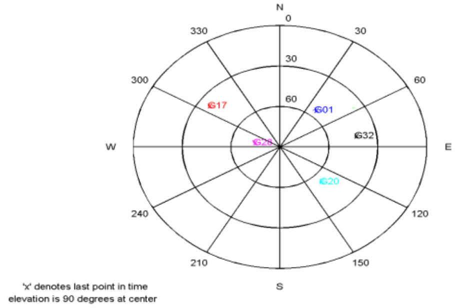


Figure 2. Sky plot.

Tables I-VIII show the Horizontal Positioning Error according to the variance of number of satellites and vision's target object. (In the tables, G means the number of visible navigation satellites and V means the number of target objects for the vision system. For example, G3V1 means that there are 3 navigation satellites and one vision target object.)

Table I shows the Horizontal Position Error when there are three visible navigation satellites, and the number of target objects for the vision system is one.

TABLE I. HORIZONTAL POSITION ERROR

G3+V1		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	19.96
1	2	**19.89**
1	3	20.05
1	5	20.41
1	10	19.87
2	1	**19.83**
3	1	20.30
5	1	20.22
10	1	**19.84**

Table II shows the Horizontal Position Error with three visible navigation satellites, and the number of target objects for the vision system is two.

TABLE II. HORIZONTAL POSITION ERROR

G3+V2		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	13.47
1	2	16.59
1	3	17.62
1	5	18.18
1	10	18.45
2	1	10.29
3	1	9.37
5	1	**8.90**
10	1	**9.10**

Table III shows the Horizontal Position Error with three visible navigation satellites, and the number of target objects for the vision system is three.

TABLE III. HORIZONTAL POSITION ERROR

G3+V3		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	13.17
1	2	16.11
1	3	17.17
1	5	17.85
1	10	18.17
2	1	10.34
3	1	9.06
5	1	**7.69**
10	1	**6.42**

Table IV shows the Horizontal Position Error with four visible navigation satellites, and the number of target objects for the vision system is zero.

TABLE IV. HORIZONTAL POSITION ERROR

G4
Horizontal Error
24.42

Table V shows the Horizontal Position Error with four visible navigation satellites, and the number of target objects for the vision system is one.

TABLE V. HORIZONTAL POSITION ERROR

G4+V1		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	19.83
1	2	20.04
1	3	19.35
1	5	19.62
1	10	20.58
2	1	19.46
3	1	19.42
5	1	**19.26**
10	1	**19.32**

Table VI shows the Horizontal Position Error with four visible navigation satellites, and the number of target objects for the vision system is two.

TABLE VI. HORIZONTAL POSITION ERROR

G4+V2		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	13.67
1	2	18.01
1	3	19.43
1	5	20.52
1	10	21.37
2	1	10.26
3	1	8.64
5	1	**7.77**
10	1	**7.51**

Table VII shows the Horizontal Position Error with four visible navigation satellites, and the number of target objects for the vision system is three.

TABLE VII. HORIZONTAL POSITION ERROR

G4+V3		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	13.61
1	2	15.68
1	3	16.28
1	5	16.73
1	10	21.79
2	1	10.63
3	1	9.10
5	1	**7.60**
10	1	**6.39**

Table VIII shows the Horizontal Position Error with five visible navigation satellites, and the number of target objects for the vision system is zero.

TABLE VIII. HORIZONTAL POSITION ERROR

G5
Horizontal Error
28.15

The following are results of experiments in the kinematic state. Figure 3 shows the kinematic state sky plot.

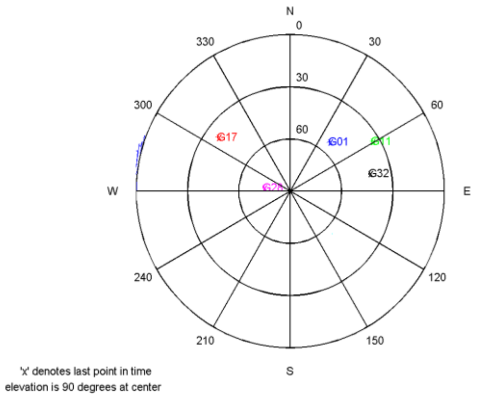


Figure 3. Sky plot of kinematic state.

Tables IX-XVI show the Horizontal Positioning Error according to the variance of number of satellites and vision's target object.

Table IX shows the Horizontal Position Error with three visible navigation satellites, and the number of target objects for the vision system is one.

TABLE IX. HORIZONTAL POSITION ERROR

G3+V1		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	21.62
1	2	21.02
1	3	21.96
1	5	**20.32**
1	10	**20.33**
2	1	21.67
3	1	20.99
5	1	20.50
10	1	**20.34**

Table X shows the Horizontal Position Error with three visible navigation satellites, and the number of target objects for the vision system is two.

TABLE X. HORIZONTAL POSITION ERROR

G3+V2		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	23.95
1	2	19.94
1	3	19.68
1	5	20.06
1	10	20.25
2	1	17.57
3	1	20.65
5	1	**16.68**
10	1	**17.43**

Table XI shows the Horizontal Position Error with three visible navigation satellites, and the number of target objects for the vision system is three.

TABLE XI. HORIZONTAL POSITION ERROR

G3+V3		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	18.07
1	2	18.88
1	3	19.29
1	5	22.05
1	10	19.93
2	1	16.13
3	1	18.96
5	1	13.30
10	1	**12.75**

Table XII shows the Horizontal Position Error with four visible navigation satellites, and the number of target objects for the vision system is zero.

TABLE XII. HORIZONTAL POSITION ERROR

G4
Horizontal Error
20.95

Table XIII shows the Horizontal Position Error with four visible navigation satellites, and the number of target objects for the vision system is one.

TABLE XIII. HORIZONTAL POSITION ERROR

G4+V1		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	**13.86**
1	2	23.18
1	3	17.52
1	5	20.26
1	10	20.78
2	1	24.86
3	1	24.10
5	1	**13.24**
10	1	24.82

Table XIV shows the Horizontal Position Error with four visible navigation satellites, and the number of target objects for the vision system is two.

TABLE XIV. HORIZONTAL POSITION ERROR

G4+V2		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	22.86
1	2	22.94
1	3	21.67
1	5	18.97
1	10	20.66
2	1	21.15
3	1	19.21
5	1	**12.45**
10	1	**11.49**

Table XV shows the Horizontal Position Error with four visible navigation satellites, and the number of target objects for the vision system is three.

TABLE XV. HORIZONTAL POSITION ERROR

G4+V3		
GNSS Weight	***Vision Weight***	***Horizontal Error***
1	1	22.28
1	2	22.64
1	3	21.50
1	5	18.65
1	10	19.53
2	1	19.56
3	1	16.57
5	1	**7.45**
10	1	**6.29**

Table XVI shows the Horizontal Position Error with five visible navigation satellites, and the number of target objects for the vision system is zero.

TABLE XVI. HORIZONTAL POSITION ERROR

G5
Horizontal Error
16.33

IV. CONCLUSION AND FUTURE WORK

Tables I-VII show the Horizontal Positioning Error according to the variance of number of satellites and vision's target object. The experimental results in Tables I-III and VIII-X indicate that the position could be determined with the addition of the vision system when there was less than four visible satellites. In Tables I-III, we can confirm that the results using three visible satellites with one, two, and three visual target objects are better than when using four visible satellites without the vision system. The performance of Horizontal Position Error was improved 19%, 64%, and 74%. In Tables V-VII, we can confirm that the results using four visible satellites with one, two, and three visual target objects are better than when using five visible satellites without the vision system. The performance of Horizontal Position Error was improved 32%, 74%, and 78%. In Tables IX-XI, we can confirm that the results using three visible satellites with one, two, and three visual target objects are better than using four visible satellites without the vision system. The performance of Horizontal Position Error was improved 0.4%, 21%, and 40%. In Tables XIII-XV, we can confirm that the results using four visible satellites with one, two, and three visual target objects are better than when using five visible satellites without the vision system. The performance of Horizontal Position Error was improved 19%, 21%, and 40%.

ACKNOWLEDGEMENT

This work was supported by the DGIST R&D Program of the Ministry of Science, ICT & Future Planning of Korea. (14-IT-01)

REFERENCES

[1] J. B. Tsui , "Fundamentals of Global System Receivers, A Software Approach", John Wiley & Sons, 2. 2000, pp. 10-27.

[2] P. Ramjee and R. Marina, "Applied Satellite Navigation Using GPS, Galileo, and Augmentation Systems", Artech House, 2005.

[3] W. B. Hofmann, H. Lichtenegger, and J. Collins, "GPS Fifth, revised edition", 2001.

[4] D. K. Elliott, "Understanding GPS : Principles and Applications", Artech House, 1996.

[5] W. Jinling, G. Matthew, L. Andrew, J. W. Jack, H. Songlai, and S. David, "Intergration of GPS/INS/Vision Sensors to Navigate Unmanned Aerial Vehicles", The International Archives of the Photogrammetry, Reomote Sensing and Spatial In formation Sciences, 37, pp. 963-969, Beijing, 2008.

[6] S. Godha and M. E. Cannon, "Integration of DGPS with a Low Cost MEMS-based Inerial Measurement Unit(IMU) for Land Vehicle Navigation Application", Proceedings of ION GPA-05, Institute of navigation, pp. 333-345, Long Beach, 2005.

[7] A. Broggi, C. Caraffi, R. I. Fedriga, and P. Grisleri, "Obstacle Detection with Stereo Vision for Off-road Vehicle Navigation", Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, 3, pp. 65-72, 2005.

[8] Y. C. Lim, M. H. Lee, C. H. Lee, S. Kwon, and J. H. Lee, "Improvement of Stereo Vision-based Position and Velocity Estimation and Tracking using a Strip-based Disparity Estimation and Inverse Perspective Map-based Extended Kalman Filter", Optics and Lasers in Engineering, 48, 9, pp. 859-868, 2010.

[9] M. Bertozzi, A. Broggi, M. Cellario, A. Fascoli, P. Lombardi, and M. Porta, "Artificial Vision in Road Vehicles", Proceeding of the IEEE, 90, 7, pp. 1258-1271, 2002.

[10] N. Srinivasa, "A Vision-based Vehicle Detection and Tracking Method for Forward Collision Warning", IEEE Intelligent Vehicle Symposium, pp. 626-631, 2002.

[11] V. Lemonde and M. Devy, "Obstacle Detection with Stereo Vision", Mechatronics and Robotics, Germany, 2004.

[12] E. J. Rossetter, J. P. Switkes, and J. C. Gerfes, "Experimental Validation of the Potential Field Lanekeeping System", International journal of automotive technology, 5, 95, 2004.

[13] R. Havard and H. Leonhard, "Gaussian Markov Random Fields", 2007.

Production-Sales Policies for New Product Diffusion under Stochastic Supply

Ashkan Negahban and Jeffrey S. Smith

Department of Industrial and Systems Engineering
Auburn University
Auburn, AL, USA
Email: {anegahban, jsmith}@auburn.edu

Abstract—**In this study, we highlight the importance of production and sales plans for new products and illustrate the need for explicitly modeling supply uncertainties when making such decisions. We consider the case of variability in the production yield and perform extensive simulation experiments to study its impact on the performance of myopic and build-up policies in terms of the expected profit and risk measures. Managerial implications concerning selection of the production and sales plan are also discussed. The results show that ignoring the production yield variation can result in potentially incorrect decisions on the product launch time. The results also show that the policy selected based on the expected profit does not necessarily minimize risk.**

Keywords*–*Innovation diffusion; Production uncertainty; Production planning; Inventory management; Monte Carlo simulation.

I. Introduction

New products play an important role in today's competitive marketplace. More than 30% percent of overall sales and profit of companies, on average, come from their new products [1][2]. It has long been known that the demand of new products follows diffusion patterns similar to those observed in epidemiology and natural sciences [3]. Once a new product is introduced into the market, some individuals (*innovators*) decide to adopt the product independently of others' decision while the timing of the adoption decision of *imitators* is influenced by word of mouth and pressure for adoption from the social system. Under certain circumstances and in the presence of extensive word of mouth spreading from past sales, if the company starts selling as many units as possible without building an initial inventory (*myopic* policy) the demand for the new product grows rapidly and soon may exceed the firm's capacity resulting in lost sales. To avoid this problem, companies generally delay product launch in order to build sufficient inventory prior to starting sales (*build-up* policy).

Decision making regarding an appropriate production-sales policy requires a deep understanding of the underlying dynamics of diffusion processes and can be difficult even for companies with a lot of experience in successful new product launches. For instance, Sony Electronics Inc. lost $1.8B in its game division and eventually laid off 3% of its workforce due to incorrect over-anticipation of the demand for PlayStation®3 [4]. In another case, Motorola Inc. who manufactured Power Mac G4 chips for Apple Inc., was unable to keep up with the rapid growth of demand for the computer [5]. Bandai Co. faced a similar problem in 1996 when the demand for Tamagotchi™, the first virtual pet, rapidly grew beyond expectations and led to lost sales. The company lost even more money when the demand declined right after they expanded their capacity in 1998 resulting in a $123 million in after-tax losses [6].

These cases prove that an effective production-sales plan is crucial to successful new product introduction. Several studies have addressed supply-restricted diffusion and evaluated various production-sales policies. However, existing literature ignores an important characteristic of any manufacturing environment: production uncertainty. Production systems exhibit significant uncertainties due to machine breakdowns, stochastic processing/tool changeover/setup times, labor availability, and quality uncertainty. These factors directly affect the supply levels and thus can influence the time to market the new product. As an example, in 2001, Microsoft Co. had to postpone the launch of Xbox® in Japan until next year and in the US by a week since they failed to meet the targeted initial inventory [7][8]. In this example, due to production uncertainties, the company needed more time than was expected to fully build the necessary inventory before the product launch. Therefore, the current literature leaves an important question unanswered: *How does uncertainty in production yield affect the company's choice of the production and sales plan?*

This paper aims to answer the above question by investigating supply-constrained new product diffusion in the presence of production uncertainties. More specifically, using a Monte Carlo simulation model, we evaluate the performance of different production-sales policies with respect to three performance measures, namely the expected Net Present Value (NPV) of profit over the product's life cycle and the 25th and 75th percentiles of the NPV of profit as two measures of risk. Through extensive experimentation, we demonstrate that ignoring production uncertainties can lead to an incorrect decision on the *best* production and sales plan under certain circumstances. Furthermore, the results indicate that the cost of making such incorrect decision is expected to increase with higher levels of uncertainty in the production environment.

The remainder of the paper is organized as follows. Section II provides a literature review and highlights the main contributions of this work. Section III describes the Monte Carlo simulation model. Experimental design and results are provided in Sections IV and V, respectively. Finally, conclusions and future research considerations are discussed in Section VI.

II. Literature Review

There are two main related streams of studies in the marketing and manufacturing literature. Both areas have been studied extensively. In fact, early studies on production/inventory management or diffusion models go back to as early as the 1960's [9][10]. The marketing literature primarily focuses on developing diffusion models to enhance demand forecasting while ignoring capacity constraints and assuming unlimited supply [11]. The manufacturing literature, on the other hand, typically involves finding the optimal production plan where the demand is assumed to be either deterministic or stochastic but following a known distribution [12][13]. In the real world, however, production constraints exist and the demand process is not exogenous. Therefore, what these studies fail to capture is the fact that supply constraints affect sales and past sales in turn have an impact on the future demand.

More recently, several studies have addressed the above issue which can be divided into two main categories based on the primary analysis tool used: analytical models and simulation studies. In the context of analytical studies, an equation-based model is proposed by Kumar and Swaminathan [14] to account for supply-constrained diffusion. The proposed diffusion model is used to evaluate the performance of myopic and build-up policies (calculated through numerical experiments) against the optimal policy (obtained by solving a mathematical model). In another study and using a similar diffusion model, closed-form expressions are derived for the demand and sales under supply constraints [15]. These expressions are then used to find the optimal capacity and time to market the new product. Myopic and build-up policies have also been studied in the presence of negative word of mouth from dissatisfied adopters [16]. A few studies also develop mixed-integer optimization models to find the optimal production and sales plan [17] and the optimal configuration of the supply chain for new products [18][19]. The primary performance measure used in all of the analytical studies presented here is the expected discounted profit over the life cycle of the new product. Moreover, none of these studies consider production/supply uncertainties.

The results of a recent survey on the application of Agent-Based Modeling and Simulation (ABMS) by Negahban and Yilmaz [11] indicate that only a few simulation studies have addressed supply-constrained diffusion processes. In [20], an ABMS is used to evaluate the performance of myopic and build-up policies with respect to the NPV of profit and lost sales under a fixed deterministic production level. In another ABMS study, Negahban et al. [21] develop a model where the company adjusts its production level based on forecasts of future demand and a production management strategy. In other words, adjustable production level is used as a substitute for building initial inventory. They evaluate the performance of different production management strategies with respect to the NPV of profit and lost sales under different planning horizons, social network structures, coefficients of innovation and imitation, and discount rates. Although the production level is not fixed, variation in production yield is not considered. Finally, a simulation framework is proposed by Negahban [22] for the newsvendor problem where the demand distribution is first predicted by ABMS while Monte Carlo simulation is used to select the optimal order quantity. The expected profit is the only performance measure while the underlying assumption is that the order will be available in full by the product launch (i.e., no variation in the supplier's production processes).

Our literature review reveals two major gaps in the existing body of knowledge. First, the impact of production uncertainties is not studied. Secondly, existing studies mainly focus only on the *expected* profit to select the best production-sales policy while the risk (which is generally an important factor for managers when making any financial investment) is not considered in the decision making process. The current paper contributes to the literature in three significant ways: (1) to the best of our knowledge, this is the first study that introduces the important notion of production uncertainty into the field of new product diffusion; (2) it uses percentiles to characterize the risk associated with different production-sales policies and shows that the policy with the highest expected profit is not necessarily the best choice under risk considerations; and, (3) it shows that the optimal production and sales plan can change based on the level of uncertainty in the production process.

III. Monte Carlo Simulation Model

A. Conceptual Model: Supply-Restricted Diffusion

The Bass model [10] is perhaps the most fundamental and widely used analytical diffusion model in the literature with the majority of other models being rooted in this model [23]. According to the Bass model, the demand of a new durable product at time period t, $d(t)$, is a function of the coefficient of innovation, p, coefficient of imitation, q, market size, m, and the cumulative number of adopters up to time t, $D(t)$, as follows: $d(t) = p(m - D(t)) + (q/m)D(t)(m - D(t))$. The Bass model was originally developed for diffusion of a product *class* (supposedly produced by many different firms) making supply constraints less relevant. However, here, we consider the case where the new product is produced and marketed by a single company. Therefore, it is possible that due to supply shortage, some adopters will not be able to purchase the product. As a result, the cumulative sales up to time t, $S(t)$, can be different from the cumulative demand, $D(t)$. Assuming that only those customers who have actually made the purchase will spread word of mouth about the product, supply shortage can influence word of mouth and demand growth rate.

Based on the above, we need a model that can capture the effect of supply constraints on the demand dynamics. In this paper, we use a modified Bass model proposed independently in [14] and [15]. The model can be expressed as follows: $d(t) = p(m - D(t)) + (q/m)S(t)(m - D(t))$. Thus, for a market with size m and at any given time t, a proportion of p of the remaining potential adopters (*innovators*) will adopt the product independently of the word of mouth influence. On the other hand, from the remaining potential adopters, the number of *imitators* that will adopt the product is proportional to the cumulative sales up to time t, $S(t)$, representing the effect of word of mouth. It is worth noting that the model is valid regardless of whether unmet demand $D(t) - S(t)$ is lost or backlogged for later fulfillment [14]. The model is also valid for the case of unlimited supply which essentially results in $S(t)$ being equal to $D(t)$ for all t. Therefore, the model is able to capture the impact of past sales on future demand. We will use this model to calculate the demand of the new product in the proposed Monte Carlo simulation model discussed below.

B. Monte Carlo Simulation Algorithm

In this model, there is a single company with a fixed average production level of L that markets the new product. In order to capture the impact of production yield uncertainty, we assume that the production yield varies around the average production level. The magnitude of this variation is determined by the percentage of variation in production yield, v. The actual production yield at each time step t, $y(t)$, is randomly sampled from the interval $[L(1 - v), L(1 + v))]$ based on a uniform distribution. Given $I(t-1)$ as the inventory carried over from

TABLE I. PARAMETER CHOICES FOR SIMULATION EXPERIMENTS

Parameter	Value/Range
Population-related parameters	
Coefficient of innovation, p	0.01, 0.03, 0.05
Coefficient of imitation, q	0.2, 0.4, 0.6
Market size, m	3000
Backlogging percentage, β	0, 0.5, 0.8, 1
Production-related parameters	
Average production level, L	100
Variation in production yield, v	0%, 5%, 10%, 15%
Unit production cost, c	1.0
Unit inventory/holding cost, h	0.001, 0.005, 0.01
Per customer waiting cost, w	0.001, 0.005, 0.01
Unit selling price, π	1.1, 1.2, 1.3
Discount rate, r	0, 0.003, 0.005, 0.01
Parameter related to the production-sales policy	
Number of inventory build-up periods, $T_{Build-up}$	0 – 25

the previous period, the total amount of supply at each time step will be $y(t)+I(t-1)$. The modified Bass model is used to calculate the *new* demand of the product at each time period, $d(t)$. Given there is backlogged demand waiting to be fulfilled from the previous period, $B(t-1)$, the *total* demand for the current time step will be $d(t)+B(t-1)$. Thus, assuming the company has already started selling the product, the instantaneous sales at t, $s(t)$, will be the minimum of supply and total demand, i.e., $min(y(t)+I(t-1), d(t)+B(t-1))$; otherwise, if the sales has not started yet (during periods of build-up), $s(t)$ will be zero. We assume that β percent of customers with unmet demand will wait to purchase the product later. At the end of each time period, the profit is calculated by subtracting the production cost, inventory cost, and cost of waiting customers from the revenue. At the end of the run (once the potential market m is almost entirely exhausted), the net present value of the profit over the diffusion time is calculated based on the given discount rate. The logic of the Monte Carlo model can be summarized in Figure 1.

IV. EXPERIMENTAL DESIGN

The experimental design is summarized in Table I. In order to select reasonable parameter choices and provide a common ground for comparison, the values/ranges are adopted from previous studies in the field [14][21]. It is worth noting that under each parameter configuration, we evaluate all integers between 0 and 25 for the number of build-up periods (with zero representing the myopic policy) and choose the best alternative as the best policy. Therefore, a total of 404,352 scenarios are studied. The simulation model is developed in MATLAB® and simulation experiments are run on a standard Dell™ desktop with a 3.00GHz quad-core CPU and 8GB of RAM.

V. ANALYSIS OF SIMULATION RESULTS

Once the Monte Carlo simulation model is developed and verified, it is used to evaluate the performance of myopic and build-up production-sales policies with respect to three metrics, namely the expected net present value of profit and the 25th and 75th percentiles of the NPV of profit as measures of risk [24]. The best production and sales plan is then the one that has the highest estimated value of the metric under consideration. In this section, we present the results of our extensive simulation experiments and outline the important findings.

```
% Model parameters
% Population-related parameters
Coefficient of innovation, $p$
Coefficient of imitation, $q$
Market size, $m$
Backlogging percentage, $\beta$

% Production-related parameters
Average production level, $L$
Percentage of variation in production yield, $v$
Unit production cost, $c$
Unit inventory/holding cost, $h$
Per customer waiting cost, $w$
Unit selling price, $\pi$
Discount rate, $r$

% Parameter related to the production-sales policy
Number of inventory build-up periods, $T_{Build-up}$

% Pseudo-code for supply-constrained diffusion under production yield uncertainty
% Initialize model variables at time 0
Set initial time step, $t = 0$
Set initial cumulative demand, $D(0) = 0$
Set initial cumulative sales, $S(0) = 0$
Set initial backlogged demand, $B(0) = 0$
Set initial inventory level, $I(0) = 0$

% Iterate by incrementing time step until the market potential is exhausted
While there is demand for the new product
    $t = t + 1$
    Sample production yield, $y(t) \leftarrow U(L(1-v), L(1+v))$
    Demand, $d(t) = p(m - D(t-1)) + (q/m)S(t-1)(m - D(t-1))$
    Update cumulative demand, $D(t) = D(t-1) + d(t)$
    % Calculate sales (do not sell if still in build-up period)
    If $t \leq T_{Build-up}$
        $s(t) = 0$
    Else
        $s(t) = min(y(t) + I(t-1), d(t) + B(t-1))$
    Endif
    Update cumulative sales, $S(t) = S(t-1) + s(t)$
    Remaining inventory, $I(t) = I(t-1) + y(t) - s(t)$
    Backlogged demand, $B(t) = max(0, d(t) + B(t-1) - s(t)) * \beta$

    % Calulcate revenue, total cost, and profit
    Revenue, $R(t) = \pi * s(t)$
    Total cost, $C(t) = c * y(t) + h * I(t) + w * B(t)$
    Profit, $P(t) = R(t) - C(t)$
End While

% Compute net present value (NPV) of profit
$NPV = 0$
For i = 1 to t
    $NPV = NPV + P(t) * (1 + r)^{1-t}$
End
```

Figure 1. Pseudo-code of the Monte Carlo simulation model.

A. Expected Profit vs. Risk

While previous studies only use the expected profit to select the best policy, with the presence of production uncertainty, we can also evaluate the associated risk. The research question under investigation then becomes whether the best policy changes based on risk. In order to answer this question, for

TABLE II. NUMBER OF CASES WHERE THE BEST POLICY IS DIFFERENT BASED ON RISK MEASURES

Risk Measure	Production Variation (v)		
	5%	10%	15%
25th percentile of profit	28	73	109
75th percentile of profit	29	51	90
Total	57	124	199

all parameter configurations, we first find the best policy under each of the above three criteria. We then identify the scenarios where the best number of build-up periods based on the expected NPV of profit is different from the best policy based on the 25th or 75th percentiles. Statistical hypothesis tests are then performed to detect statistical difference between the corresponding percentiles of the resulting two candidate policies. If a statistical difference exists, then we can conclude that the risk associated with the policy selected based on the percentile under consideration is actually more desirable than the policy with the best NPV of profit.

The number of scenarios (out of 3,888 parameter configurations) where a significant statistical difference was detected is provided in Table II. The results show that the policy with the highest NPV of profit does not necessarily have the minimum risk. In other words, the decision on the production-sales plan can change if the primary criterion is risk rather than the expected profit. This is an important finding since existing studies select the best policy only based on the expected profit while the risk has been ignored. High levels of production uncertainty not only increase variability in production cost, but also affect supply which in turn has an impact on future demand. As a result, a higher uncertainty level is expected to increase variability of NPV of profit and have a greater impact on the risk associated with it making consideration of risk measures even more important.

We also investigate the impact of production variation under different diffusion parameters namely coefficients of innovation (p) and imitation (q). Given p, q, and v, there are 432 scenarios. Figure 2 shows the number of cases where the best number of build-up periods chosen based on risk is different (statistically) from the one based on the average NPV. The figure shows that the impact of production uncertainty varies based on p and q. We believe this is mainly due the fact that different combinations of p and q impact the dynamics of the problem in a different way (essentially, p and q affect initial demand and its growth rate). For instance, when both p and q are small, the initial demand levels are low and the demand grows slowly. In such cases and under the best policy, the demand rarely exceeds supply resulting in almost no backlogged or lost demand (supply abundance). On the other hand, when both p and q are large, the initial demand level and its growth rate are both high resulting in high levels of backlogged or lost sales (supply scarcity) which in turn affect the future sales and demand. Therefore, for other combinations of p and q, we expect to see different degrees of these two effects and thus a different impact by production yield variation on the best number of build-up periods as shown in the figure.

It is worth noting that since standard hypothesis testing procedures are not readily available for comparing ordinal values of two populations, nonparametric tests will be necessary when comparing percentiles [25]. In this paper, for all of the statistical tests for comparing the percentiles, we use a nonparametric double bootstrap method which is also based on Monte Carlo simulation sampling; however, since statistical techniques are out of the scope of this paper and for the sake of conciseness, the details are not provided here. The reader is referred to Spiegelman and Gates [26] for more information.

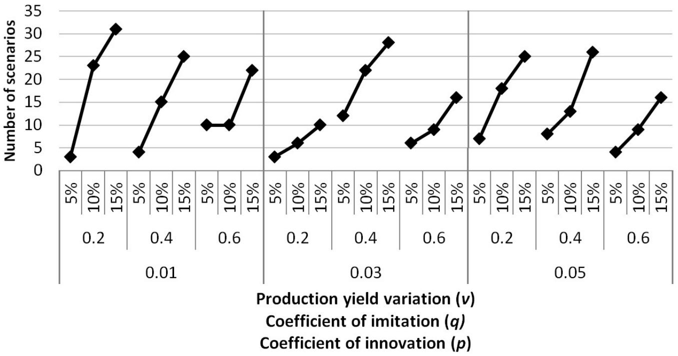


Figure 2. Number of cases where the best policy is different based on risk.

B. Deterministic vs. Stochastic Production Yield

The results presented in this section assess the impact of supply uncertainty on the best production-sales policy. Although the case of Xbox® in 2001 (discussed earlier in the paper) shows that such uncertainties matter, no theoretical work has been done in this area. The research question to be answered is whether the best policy can change if variation in production yield is explicitly considered. In order to answer this question, for each performance measure, we first identify the scenarios where the best number of build-up periods for the deterministic case ($v = 0$) is different from the best policy with production yield variation taken into account ($v > 0$). Statistical tests are then performed to test the significance of these differences. For the resulting two candidate policies, Welch's t-test is used to test the difference in average NPV of profit while a double bootstrap method is used for comparing the 25th and 75th percentiles of the profit.

The number of cases (out of 3,888 parameter configurations) where production yield variation has actually changed the best number of build-up periods is provided in Table III. For instance, when the 75th percentile of the profit is used as the primary performance measure, a 15% variation in the production yield results the optimal decision to be different from the deterministic case in 179 parameter configurations. The results are inline with our expectation from empirical findings that supply uncertainties affect the number of build-up periods needed to build the required initial inventory. Therefore, we have shown that variation in production yield is an important factor that needs to be explicitly modeled when evaluating production-sales policies. The results also suggest that, in general, higher levels of production variation is expected to increase the likelihood of making an incorrect decision on the best number of build-up periods if ignored.

Given p, q, and v, Figure 3 shows the number of scenarios (out of 432) where the best number of build-up periods is different (statistically) from the deterministic case for at least one of the three performance measures. As expected, for

TABLE III. NUMBER OF SIGNIFICANT STATISTICAL DIFFERENCES FROM THE DETERMINISTIC CASE

Performance Measure	Production Variation (v)		
	5%	10%	15%
Average NPV of profit	64	53	79
25th percentile of profit	83	88	101
75th percentile of profit	75	108	179
Total	222	249	359

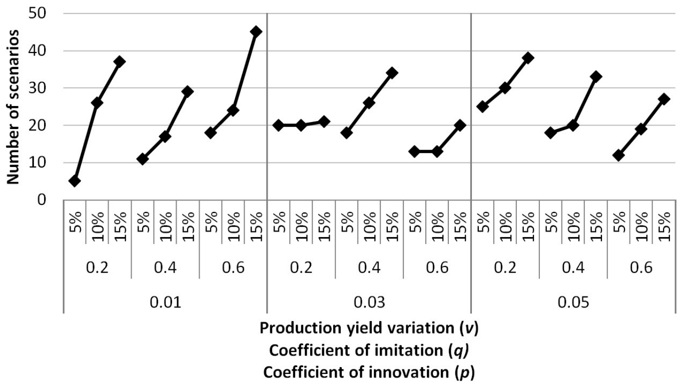


Figure 3. Number of statistical differences from the deterministic case.

all combinations of p and q, we see an increasing trend in the number of changes in the best production-sales plan. Furthermore, similar to the results of the previous section, due to the complex supply-sales-demand inter-dependencies and also the nonlinear effect of the coefficient of imitation, we expect the impact of production yield variation on the best policy to vary for different levels of p and q.

C. Practical Implications

We have shown that production uncertainties affect the best production-sales plan. Companies can easily overlook these uncertainties and thus make a potentially incorrect decision about the required level of initial inventory and time to market the product. In this section, we present three main aspects of our findings that can provide deeper insights for managers.

Change in the best number of build-up periods: The results show that in the presence of production uncertainties, the length of the build-up period can range from as much as 2 periods shorter to 3 periods longer than the deterministic case, illustrating a significant perspective difference from the case where production variation is ignored (Figure 4). However, in most cases, this difference is only one period. In the presence of variation, if the company gets lucky (if they produce more than the average production level in several consecutive periods), it is possible that they can reach the targeted initial inventory faster enabling the company to market the product earlier. This will move the revenues closer to the present time and reduce inventory costs resulting in higher profits. Therefore, when using the average NPV or the 75th percentile of profit, in most cases, ignoring production uncertainties would yield to a policy where the length of the build-up period is one period longer. On the other hand, in order to guarantee that the initial inventory will be met by product launch, the company will generally need longer periods of build-up to reduce the loss of sales due to insufficient supply which is why for the case of the 25th percentile, if production uncertainty is ignored, it is more likely that we incorrectly select a build-up period that is one period shorter than the actual best value.

TABLE IV. MAXIMUM PERCENTAGE OF DIFFERENCE FROM THE DETERMINISTIC CASE ($v = 0$)

Risk Measure	Production Variation (v)		
	5%	10%	15%
Average NPV of profit	0.22	0.35	0.53
25th percentile of profit	0.30	0.54	0.94
75th percentile of profit	0.39	0.78	1.03

Magnitude of statistical differences: The magnitude of difference from the deterministic case can be thought of as the expected cost of making an incorrect decision by ignoring production uncertainties. This cost is expected to increase with higher levels of variation which also matches the results presented in Table IV. Moreover, the results also indicate that the 75th percentile is more sensitive to these variations than the other two performance measures for all levels of v. Therefore, when the probability of making higher profits is the primary risk factor, considering the uncertainties in the production processes becomes even more important for companies.

Practical difference: The magnitude of difference discussed above is also important when considering the concept of *meaningful practical difference* (δ) defined as the minimum difference that is important to detect. Thus, a difference of less than δ between the performance of two policies is considered *practically* insignificant although a statistical difference is detected. For example, if a difference of d is detected between the performance of the best policies in the deterministic and stochastic cases, the effect of production variability is practically insignificant if $d < \delta$. The minimum practical difference can also be used to determine the number of replications. When comparing two different policies, a sufficiently large sample will suggest a statistical difference at any level of significance since it enables us to detect even minute differences between the two samples. Therefore, after running the model for an initial number of replications, we can stop if a difference of less than δ is detected suggesting that the two alternatives are not practically different. Using the same approach, the number of replications for our experiments was chosen to be 1,000 based on a set of pilot runs.

VI. CONCLUSION AND FUTURE WORK

Through comprehensive experimentation using Monte Carlo simulation, we investigate the effect of production uncertainty on the best production and sales plan for new products and show that the time to market can be affected by variations in production yield. We also compare production-sales policies with respect to measures of risk and show that the policy with the maximum expected profit does not necessarily minimize risk. Finally, we discuss theoretical and practical implications of the findings to provide deeper insights on supply-restricted diffusion for both researchers and practitioners.

Myopic and build-up plans are heuristic policies that may not necessarily lead to the *optimal* policy [14]. The impact of supply uncertainties on the optimal production-sales plan is currently under investigation by the authors. Moreover, the only type of uncertainty considered here is production yield

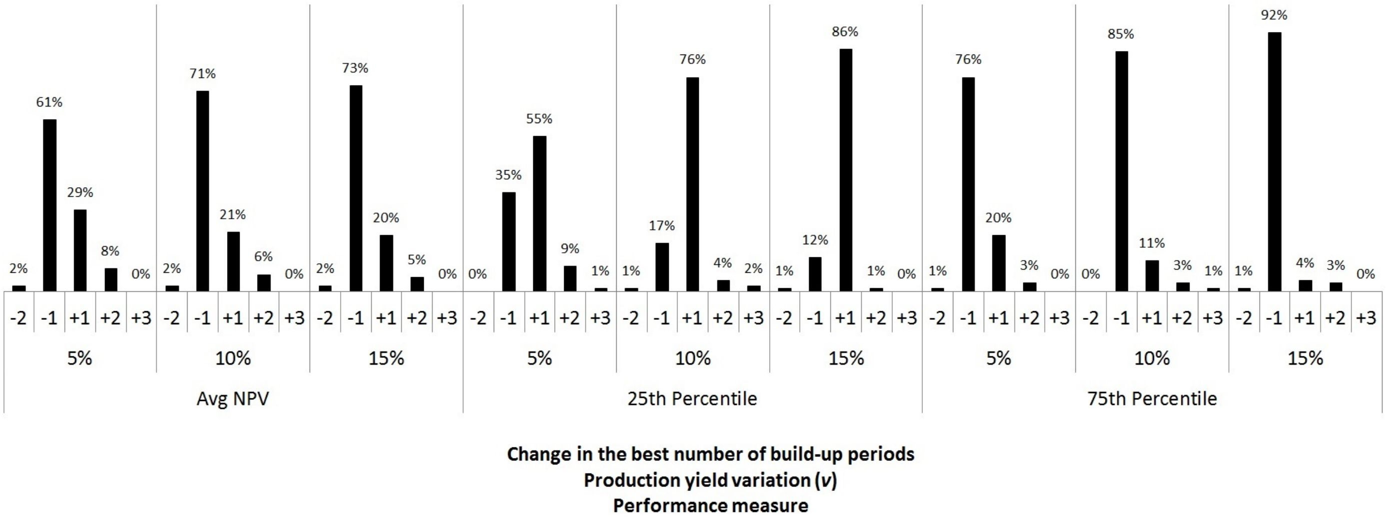


Figure 4. Distribution of the change in the best number of build-up periods from the deterministic case.

variation. Different sources of uncertainties exist in a supply chain making the study of the impact of other types of uncertainties on diffusion dynamics an interesting area for future research. Finally, studying the impact of other distributions for variations in production yield on the findings of this paper would be another interesting extension for future research.

REFERENCES

[1] J. Hauser, G. J. Tellis, and A. Griffin, "Research on innovation: A review and agenda for marketing science," Marketing Science, vol. 25, no. 6, 2006, pp. 687–717.

[2] A. Griffin, "PDMA research on new product development practices: Updating trends and benchmarking best practices," Journal of Product Innovation Management, vol. 14, no. 6, 1997, pp. 429–458.

[3] V. Mahajan, E. Muller, and F. M. Bass, "New product diffusion models in marketing: A review and directions for research," Journal of Marketing, vol. 54, no. 1, 1990, pp. 1–26.

[4] "Sony to cut game workers in US," Los Angeles Times, June 7, 2007.

[5] "Shortage chips Apple net – supplier delays will hold results down," New York Post, September 21, 1999.

[6] T. Higuchi and M. D. Troutt, "Dynamic simulation of the supply chain for a short life cycle product–lessons from the Tamagotchi case," Computers & Operations Research, vol. 31, no. 7, 2004, pp. 1097–1114.

[7] "Microsoft puts off introduction of Xbox game console in Japan," New York Times, August 27, 2001.

[8] "Microsoft delays release of Xbox game system by a week," New York Times, September 22, 2001.

[9] A. J. Clark and H. Scarf, "Optimal policies for a multi-echelon inventory problem," Management Science, vol. 6, no. 4, 1960, pp. 475–490.

[10] F. M. Bass, "A new product growth for model consumer durables," Management Science, vol. 15, no. 5, 1969, pp. 215–227.

[11] A. Negahban and L. Yilmaz, "Agent-based simulation applications in marketing research: An integrated review," Journal of Simulation, vol. 8, 2014, pp. 129–142.

[12] M. Stevenson, L. C. Hendry, and B. G. Kingsman, "A review of production planning and control: The applicability of key concepts to the make-to-order industry," International Journal of Production Research, vol. 43, no. 5, 2005, pp. 869–898.

[13] J. Mula, R. Poler, J. Garcia-Sabater, and F. Lario, "Models for production planning under uncertainty: A review," International Journal of Production Economics, vol. 103, no. 1, 2006, pp. 271–285.

[14] S. Kumar and J. M. Swaminathan, "Diffusion of innovations under supply constraints," Operations Research, vol. 51, no. 6, 2003, pp. 866–879.

[15] T.-H. Ho, S. Savin, and C. Terwiesch, "Managing demand and sales dynamics in new product diffusion under supply constraint," Management Science, vol. 48, no. 2, 2002, pp. 187–206.

[16] Y. Xiaoming, P. Cao, M. Zhang, and K. Liu, "The optimal production and sales policy for a new product with negative word-of-mouth," Journal of Industrial and Management Optimization, vol. 7, no. 1, 2011, pp. 117–137.

[17] M. Cantamessa and C. Valentini, "Planning and managing manufacturing capacity when demand is subject to diffusion effects," International Journal of Production Economics, vol. 66, no. 3, 2000, pp. 227–240.

[18] M. Amini and H. Li, "Supply chain configuration for diffusion of new products: An integrated optimization approach," Omega, vol. 39, no. 3, 2011, pp. 313–322.

[19] H. Li and M. Amini, "A hybrid optimisation approach to configure a supply chain for new product diffusion: A case study of multiple-sourcing strategy," International Journal of Production Research, vol. 50, no. 11, 2012, pp. 3152–3171.

[20] M. Amini, T. Wakolbinger, M. Racer, and M. G. Nejad, "Alternative supply chain production-sales policies for new product diffusion: An agent-based modeling and simulation approach," European Journal of Operational Research, vol. 216, no. 2, 2012, pp. 301–311.

[21] A. Negahban, L. Yilmaz, and T. Nall, "Managing production level in new product diffusion: An agent-based simulation approach," International Journal of Production Research, 2014, http://dx.doi.org/10.1080/00207543.2014.885663.

[22] A. Negahban, "A hybrid simulation framework for the newsvendor problem with advertising and viral marketing," in Proceedings of the 2013 Winter Simulation Conference, R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, Eds. Institute of Electrical and Electronics Engineers, Inc., 2013, pp. 1613–1624.

[23] F. M. Bass, "Comments on "A new product growth for model consumer durables"," Management Science, vol. 50, no. 12, 2004, pp. 1833–1840.

[24] B. L. Nelson, "The MORE plot: Displaying measures of risk & error from simulation output," in Proceedings of the 2008 Winter Simulation Conference, S. Mason, R. Hill, L. Monch, O. Rose, T. Jefferson, and J. Fowler, Eds. Institute of Electrical and Electronics Engineers, Inc., 2008, pp. 413–416.

[25] W. J. Conover, Practical Nonparametric Statistics, 3rd ed. New York, NY, USA: John Wiley Inc., 1980.

[26] C. Spiegelman and T. J. Gates, "Post hoc quantile test for one-way analysis of variance using a double bootstrap method," Transportation Research Record: Journal of the Transportation Research Board, vol. 1908, 2005, pp. 19–25.

Particle Simulation of Granular Flows in Electrostatic Separation Processes

Ida Critelli

Dipartimento di Ingegneria Meccanica
Politecnico di Milano
Milano, Italy 20126
Email: ida.critelli@polimi.it

Alessandro Tasora

Dipartimento di Ingegneria Industriale
Università di Parma
Parma, Italy 43100
Email: alessandro.tasora@unipr.it

Andrea Degiorgi and
Marcello Colledani

Dipartimento di Ingegneria Meccanica
Politecnico di Milano
Milano, Italy 20126
Email: andrea.degiorgi@polimi.it
Email: marcello.colledani@polimi.it

***Abstract*—In waste processing technology, the recent Corona Electrostatic Separation (CES) method is used to separate conductive from non-conductive particles in recycling streams. This paper proposes an innovative simulation approach based on non-smooth dynamics. In this context, a differential-variational formulation is used to implement a scalable and efficient time integrator that allows the large-scale simulation of trajectories of particles with different properties under the effect of particle-particle interactions and frictional contacts. Issues related to performance optimization, fast collision detection and parallelization of the code are discussed.**

Keywords*–*Granular flows, particles, waste processing

I. Introduction

Numerical simulation of waste processing devices is becoming a key factor in the improvement of material recycling: this work focuses on the simulation of the CES processing technology. Such a technology uses electrostatic forces to separate conductor materials from insulating materials within a continuous granular flow.

Particles with a varying degree of materials, such as plastic or copper, are produced by a preprocessing stage where Printed Circuits Boards (PCB) are shred and crushed, then they are dropped on the rotating drum of the ECS device, where corona charging is used to establish a negative charge on particles when passing between two high-voltage electrodes [1].

Research on numerical methods for simulating this class of problems is relatively recent [2][3].

Simulating this type of devices proved to be quite challenging in the past, because it requires the simulation of a so called *dense granular flow*, that is a collection of many particles interacting with the surrounding ones. It is known that only few special cases of granular flows can be simplified into macroscopic models, and in literature there are some example of homogenizations that can lead to simpler Finite Element Analysis (FEA), expecially with spherical particles; but, the case at hand is not one of these. In granular flows each particle, with arbitrary shape and mass properties, exploits three dimensional displacements and rotations, hence six degrees of freedoms are introduced per each particle. If one considers that few seconds of simulation of a waste processing device like our CES can involve hundreds of thousands, if not millions, of particles, this instantly gives an idea of how challenging the problem is from a numerical point of view. On top of this, the fact that particles touch each other, introduces a major complication that cannot be solved with classical tools of ordinary differential equations and that leads to the complex field of non-smooth dynamics.

This motivated our research on a custom software tool that can simulate the CES, and similar separation processes, by means of an efficient solution scheme based on Differential Variational Inequalities (DVI).

Results show that we can obtain reliable simulations with large timesteps and by running a single computer for few minutes. Also, the software tool that we developed can simulate motors, moving parts and other auxiliary devices that might interact with the granular flow.

This article starts with an introductory section about the CES process, then it presents the mathematical background of the DVI simulation algorithm, it discusses how to overcome the difficulty of massive collision detection in granular flows, and finally discusses the computation of electrostatic forces and the software implementation. The last section shows examples of results that can be achieved with this type of simulations.

II. The electrostatic separation process

The CES is a waste processing device that separates conductor materials from insulating materials within a granular flow of fine particles. An example, from our labs, is shown in Figure 1.

The granular flow receives a discharge of electricity, which gives high surface charge to the plastic particles, causing them

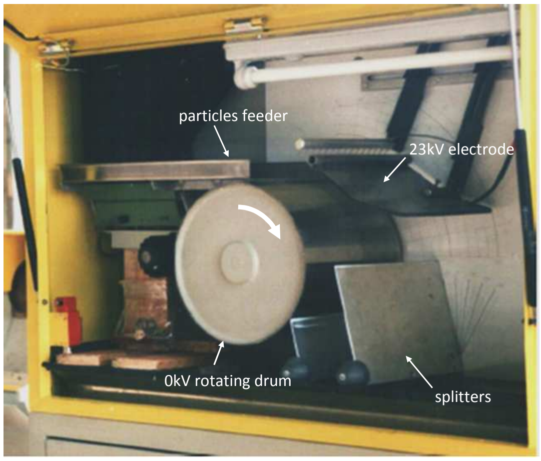


Figure 1: A CES processing device.

ISBN: 978-1-61208-371-1

to be attracted to the rotor surface until they are brushed down into a proper bin, below the drum. On the other hand, metal particles do not remain charged, because their charge rapidly decreases as they touch the rotor, so they fall into the farthest bin. In case a particle is made of a mixture of materials, it drops in intermediate bins. Metal particles are charged only by electrostatic induction, and this causes a slight attraction effect towards the static electrode of opposite sign.

The efficiency of the process is affected by non-controllable parameters, such as particle shape and size. The quality of the separation can be influenced by controllable parameters like electrode voltage and position, drum speed, splitters position, and feed rate.

III. The DVI formulation

Granular flows are complex dynamical phenomena where a large number of particles interact with each other and with the surrounding environment. The fact that interactions are of discontinuous nature, for example because of hard contacts, introduce a relevant difficulty in the time integration of the system.

Most literature deals with this class of problems by means of the so-called Discrete Element Method (DEM), an approach that regularizes the non-smooth contact phenomena by introducing regularized smooth functions [4]. This means that hard contacts between particles in DEM are approximated as spring-dashpot systems, whose stiff and nonlinear nature requires very short timesteps for the integration of the resulting Ordinary Differential Equations (ODE) [5]. If large timesteps are used in DEM, the explicit ODE time integration may lead to divergence.

In search of a faster and more robust way for simulating granular flows, we developed [6] a time stepping method based on the recent theory of DVI. Such an approach is able to handle up to millions of contacts between particles; it directly captures the discontinuous nature of contacts with set-valued functions, it does not require short integration time steps and it has stability properties that are similar to an implicit method for differential equations.

First researches on a special type of DVI, related to contact problems, can be traced back to the Measure Differential Inclusion theory (MDI) developed in [7][8].

In the DVI framework, the right hand side of a differential expression is a set-valued function [9], so it is an inclusion, as a generalization of classical ODEs that have simple equalities. This is useful in mechanics, since non-smooth pehenomena can be described by means of set-valued functions such as the Signorini contact law or the Coulomb friction. Note that no smooth regularization of discontinuities are needed.

One can discretize DVIs in time, even with large timesteps, obtaining a Variational Inequality (VI) problem for each timestep [10], since reaction forces must satisfy set inclusions rather than algebraic equations.

A DVI is defined as the problem of finding the function $\boldsymbol{x}$ on $[0, T]$:

$$\frac{d\boldsymbol{x}}{dt} = f(t, \boldsymbol{x}, \boldsymbol{u}) \tag{1}$$

$$\boldsymbol{u} \in \mathsf{SOL}\left(\mathbb{K}, F(t, \boldsymbol{x}(t), \cdot)\right) \tag{2}$$

where the state of the system is defined by $\boldsymbol{x}$, and $\mathsf{SOL}(\mathbb{K}, F)$ in the solution of the VI problem:

$$\boldsymbol{u} \in \mathbb{K} \quad : \quad \langle F(\boldsymbol{u}), \boldsymbol{y} - \boldsymbol{u}\rangle \quad \boldsymbol{y} \in \mathbb{K} \tag{3}$$

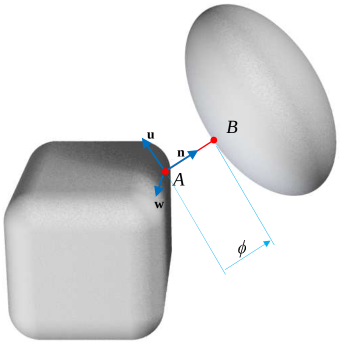


Figure 2: Contact point reference.

given a closed and convex $\mathbb{K} \in \mathbb{R}^n$ set, and given a continuous $F(\boldsymbol{u}) : \mathbb{K} \to \mathbb{R}^n$.

For a problem of particle dynamics, for instance, the state is $\boldsymbol{x} = \{\boldsymbol{q}^T, \boldsymbol{v}^T\}^T$, with positions $\boldsymbol{q}$ and velocities $\boldsymbol{v}$, and $\boldsymbol{u}$ is the set of reaction forces, that must satisfy a VI. For more informations on the DVI theory, see [11]–[15].

In our context of particle dynamics, the granular material is composed of many small particles or large bodies in contact. The system state is defined by the vector of generalized coordinates $\boldsymbol{q} \in \mathbb{R}^{m_q}$ and the vector of generalized speeds $\boldsymbol{v} \in \mathbb{R}^{m_v}$ of the particles. Rotation of each particle is represented with an unit quaternion $\epsilon \in \mathbb{H}_1$ The mass matrix $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{m_v \times m_v}$ includes all the masses and inertia tensors of the particles.

Aside from contacts, in granular flow simulations, particles might be affected by field forces such as gravity, aerodynamic forces, etc. In the case of the CES device described here, also forces caused by electrostatic interactions are added. All these forces will be denoted $\boldsymbol{f}_e(\boldsymbol{q}, \boldsymbol{v}, t)$ in the following. The total force field $\boldsymbol{f}_t(\boldsymbol{q}, \boldsymbol{v}, t) \in \mathbb{R}^{m_v}$ includes also the gyroscopic forces $\boldsymbol{f}_c(\boldsymbol{q}, \boldsymbol{v})$; we remark that for the class of problems at hand, gyroscopic forces could be omitted without compromising the results, in search of a more stable numerical integration (because in free falling slender particles, gyroscopic forces often lead to sharp changes in spin directions, thus leading to potential numerical stiffness).

Bilateral constraints are useful to model hinges and joints in devices. In our CES model, for instance, this class of constraints is used to model the revolute joints of the rotating drum, and the constraints for the movable splitting planes. Bilateral joints are introduced via a set $\mathcal{G}_B$ of scalar constraint equations:

$$\Psi^i(\vec{q}, t) = 0, \quad i \in \mathcal{G}_\mathcal{B}. \tag{4}$$

We introduce $\nabla_q \Psi^i = \left[\partial \Psi^i / \partial \vec{q}\right]^T$ and $\nabla \Psi^{i^T} = \nabla_q \Psi^{i^T} \Gamma(\boldsymbol{q})$, to express the constraint (4) at the velocity level after differentiation:

$$\frac{d\Psi^i(\vec{q}, t)}{dt} = \nabla \Psi^{i^T} \boldsymbol{v} + \frac{\partial \Psi^i}{\partial t} = 0, \quad i \in \mathcal{G}_\mathcal{B}. \tag{5}$$

The term $\frac{\partial \Psi^i}{\partial t}$ is needed only for time-dependent constraints. This is the case, in our CES model, of the motor that imposes a constant angular velocity to the rotating drum.

For the i-th contact between two bodies A and B, let $\mathbf{n}_i$ be the normal at the contact point, directed toward the exterior of A, as in Figure 2. Let $\mathbf{u}_i$ and $\mathbf{w}_i$ be two vectors in the contact plane such that $\mathbf{n}_i, \mathbf{u}_i, \mathbf{w}_i \in \mathbb{R}^3$ are orthogonal vectors, and let Φ_i represent a signed contact distance, assumed to be differentiable at least in vicinity of the contact.

The contact force has a normal component $\boldsymbol{F}_{i,N} = \widehat{\gamma}_{i,n}\mathbf{n}_i$, that must satisfy the Signorini contact rule $\widehat{\gamma}_{i,n} \geq 0 \perp \Phi_i(\cdot) \geq 0$, and a tangential component $\boldsymbol{F}_{i,T} = \widehat{\gamma}_{i,u}\mathbf{u}_i + \widehat{\gamma}_{i,w}\mathbf{w}_i$, caused by friction, along two tangential directions $\mathbf{u}_i, \mathbf{w}_i$, that can represent either the sticking or sliding friction. Within the Amontons-Coulomb theory [16] of dry friction, that we consider adequate for the case of inter-particle friction, the ratio between the normal and the tangential force is limited by friction coefficient μ_i.

Frictional unilateral contacts define a set $\mathcal{G}_A$. For each contact $i \in \mathcal{G}_A$, we introduce the tangent space generators, that can be derived as tangent constraint Jacobians: $\boldsymbol{D}^i_{\gamma_u}$, $\boldsymbol{D}^i_{\gamma_w}$.

We can express the friction model using the maximum dissipation principle, thus leading to a nonlinear program $(\widehat{\gamma}_{i,u}, \widehat{\gamma}_{i,w}) = \underline{\text{argmin}}\ \boldsymbol{v}^T \left(\boldsymbol{D}_{\gamma_u}\widehat{\gamma}_{i,u} + \boldsymbol{D}_{\gamma_w}\widehat{\gamma}_{i,w}\right)$ subject to constraint $\sqrt{\widehat{\gamma}^2_{i,u} + \widehat{\gamma}^2_{i,w}} \leq \mu\widehat{\gamma}_{i,n}$, or equivalently by introducing a multiplier λ_v and writing the Fritz John optimality conditions for such nonlinear program:

$$\begin{aligned}&\nabla_{\gamma_u,\gamma_w}\boldsymbol{v}^T \left(\boldsymbol{D}_{\gamma_u}\widehat{\gamma}_{i,u} + \boldsymbol{D}_{\gamma_w}\widehat{\gamma}_{i,w}\right) + \\ &-\lambda^i_v \nabla_{\gamma_u,\gamma_w}\left(\mu^i\widehat{\gamma}_{i,n} - \sqrt{\widehat{\gamma}^2_{i,u} + \widehat{\gamma}^2_{i,w}}\right) = 0 \end{aligned} \tag{6}$$

$$\mu^i\widehat{\gamma}_{i,n} - \sqrt{\widehat{\gamma}^2_{i,u} + \widehat{\gamma}^2_{i,w}} \geq 0, \quad \perp \quad \lambda^i_v \geq 0. \tag{7}$$

Finally, one can recognize that the following differential model with equilibrium constraints, which expresses the full model of the system including force fields, inertial forces, contacts and bilateral constraints, is a DVI:

$$\begin{array}{rcl} i \in \mathcal{G}_{\mathcal{B}} & : & \Psi^i(\boldsymbol{q}, t) = 0 \\ i \in \mathcal{G}_{\mathcal{A}} & : & \widehat{\gamma}^i_n \geq 0 \quad \perp \quad \Phi^i(\boldsymbol{q}) \geq 0 \\ & & \nabla_{\gamma_u,\gamma_w}\boldsymbol{v}^T \left(\boldsymbol{D}_{\gamma_u}\widehat{\gamma}_{i,u} + \boldsymbol{D}_{\gamma_w}\widehat{\gamma}_{i,w}\right) \\ & & -\lambda^i_v \nabla_{\gamma_u,\gamma_w}\left(\mu^i\widehat{\gamma}_{i,n} - \sqrt{\widehat{\gamma}^2_{i,u} + \widehat{\gamma}^2_{i,w}}\right) = 0 \\ & & \mu^i\widehat{\gamma}_{i,n} - \sqrt{\widehat{\gamma}^2_{i,u} + \widehat{\gamma}^2_{i,w}} \geq 0, \quad \perp \quad \lambda^i_v \geq 0 \\ \dot{\boldsymbol{q}} & = & \Gamma(\boldsymbol{q})\boldsymbol{v} \\ \boldsymbol{M}(\boldsymbol{q})\dfrac{d\boldsymbol{v}}{dt} & = & \displaystyle\sum_{i \in \mathcal{G}_{\mathcal{A}}} \left(\widehat{\gamma}_{i,n}\boldsymbol{D}^i_{\gamma_n} + \widehat{\gamma}_{i,u}\boldsymbol{D}^i_{\gamma_u} + \widehat{\gamma}_{i,w}\boldsymbol{D}^i_{\gamma_w}\right) + \\ & & + \displaystyle\sum_{i \in \mathcal{G}_{\mathcal{B}}} \widehat{\gamma}^i_{\mathcal{B}}\nabla\Psi^i + \boldsymbol{f}_t(t, \boldsymbol{q}, \boldsymbol{v}) \end{array} \tag{8}$$

The DVI model can be discretized in time, using a timestep h. To this end we set $\gamma = h\widehat{\gamma}$, we use an exponential map $\Lambda(\cdot)$ for incremental update of quaternions during the time integration [17][18], and we get:

$$\begin{array}{rcl} i \in \mathcal{G}_{\mathcal{B}} & : & \dfrac{1}{h}\Psi^i(\vec{q}^{(l)}) + \nabla\Psi^{i^T}\vec{v}^{(l+1)} + \dfrac{\partial\Psi^i}{\partial t} = 0 \\ i \in \mathcal{G}_{\mathcal{A}} & : & \gamma_{i,n} \geq 0 \perp \frac{1}{h}\Phi^i(\vec{q}^{(l)}) + \nabla\Phi^{i^T}\vec{v}^{(l+1)} \geq 0 \\ & & \nabla_{\gamma_u,\gamma_w}\boldsymbol{v}^T \left(\boldsymbol{D}_{\gamma_u}\gamma_{i,u} + \boldsymbol{D}_{\gamma_w}\gamma_{i,w}\right) \\ & & -\lambda^i_v \nabla\left(\mu^i\gamma_{i,n} - \sqrt{{\gamma_{i,u}}^2 + {\gamma_{i,w}}^2}\right) = 0 \\ & & \mu^i\gamma_{i,n} - \sqrt{{\gamma_{i,u}}^2 + {\gamma_{i,w}}^2} \geq 0, \perp\lambda^i_v \geq 0 \\ \boldsymbol{q}^{(l+1)} & = & \Lambda(\boldsymbol{q}^{(l)}, \boldsymbol{v}^{(l+1)}, h) \\ \boldsymbol{M}^{(l)}\boldsymbol{v}^{(l+1)} & = & \displaystyle\sum_{i \in \mathcal{G}_{\mathcal{A}}} \left(\gamma_{i,n}\boldsymbol{D}^i_{\gamma_n} + \gamma_{i,u}\boldsymbol{D}^i_{\gamma_u} + \gamma_{i,w}\boldsymbol{D}^i_{\gamma_w}\right) + \\ & & + \displaystyle\sum_{i \in \mathcal{G}_{\mathcal{B}}} \gamma^i_{\mathcal{B}}\nabla\Psi^i + h\boldsymbol{f}_t(t, \boldsymbol{q}, \boldsymbol{v}) + \boldsymbol{M}^{(l)}\boldsymbol{v}^{(l)} \end{array} \tag{9}$$

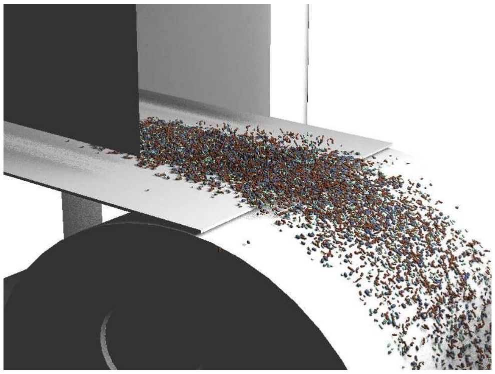


Figure 3: Simulation of the granular flow. Note the different shapes of particles. This example, although simple, leads to half a million of unknown reaction forces in frictional contacts, on average. At each time step, those unknown forces must be solved as a VI problem.

The previous problem is a mixed nonlinear complementarity problem, sub-case of VIs: most results for existence of solutions require monotonicity of the mapping defining the complementarity problem, that also implies convexity of the solution set of the nonlinear complementarity problem [19]. A relaxation of the original problem, that leads to a convex problem whose solution is guaranteed, has been proposed in [20] and it is used in our software.

Once cast to a convex problem, this can be formulated as a second-order Cone Complementarity Problem (CCP), that is also a special type of VI and is also a conically-constrained quadratic optimization problem. As a convex optimization problem, it can be solved by a modification of the Spectral Projected Gradient (SPG) method for the solution of convex-constrained optimization problems, developed by [21]. In the original SPG scheme, the projection operator performs the projection on boxed constraints, whereas we perform projection on the friction cones.

The formulation of (9) can describe only inelastic collisions; but a simple modification can also introduce a restitution coefficient.

We remark that the contact model is based on few parameters (friction coefficient, optionally also restitution coefficient or compliance and damping). Although limited in number of parameters, this simplicity pays back in terms of high computational efficiency and ease of use.

IV. Collision detection

One of the biggest bottlenecks in granular flow simulations is the computation of the collision points between pairs of particles, and between particles and the surrounding environment. There are two main sources of difficulties: one is related to finding the pairs of potential contacting bodies, and the other is computing the geometric location of the contact point(s) between those pairs.

The first difficulty is addressed by the so-called *broad-phase* collision detection stage. Such algorithm consists in pre-filtering the particles in order to detect only the small portion of pairs that potentially could collide, hence avoiding a brute force $O(n^2)$ test for all pairs - clearly inpracticable even for few thousands of particles. Our broadphase stage leverages the Bullet3D open-source library [22].

The second difficulty is related to the fact that there are many types of collision shapes, as shown in Figure 3, and contact points must be created for pairs in this second phase, called *narrow phase*. Ideally on would like to use polihedrons, spheres, cylinders, boxes etc., as well as compounds of those primitives, but then, algorithms for all possible combination between those types must be implemented. A solution could be to use algorithms of the GJK family [23] that requires only a single member function for each class of primitive, i.e., the computation of the support point. This is the default narrow-phase algorithm for the Bullet3D library, however we experienced that the GJK algorithm suffers a noticeable lack of precision when working with primitives whose size is largely different, which is exactly the case of the small particles that come into touch with the rotating cylinder. To overcome this problem, we implemented a closed form solution for computing contacts between spheres and cylinders (interesting enough, GJK is not affected by this issue for the sphere-vs-box case). This works properly; but if one desires particles that are non-spherical, for instance conical, this would require the development of other closed form algorithms for cone-vs-cylinder, cone-vs-cone, etc. To avoid this increase in coding complexity, and yet resolve the issue of low precision in GJK, we decided to model the collision shape of irregular particles by using cluster of spheres, as in Figure 4. This is an approximation of the real shapes (which are known only at a statistical level anyway), but it performs well in terms of algorithmic robustness and precision.

The ability of simulating particles with non-spherical shapes is important because, as discussed by Lu et al. [3], the shape of the particle can affect the global outcome of the separation process; for example, spherical particles tend to fall farther than thin and slender particles.

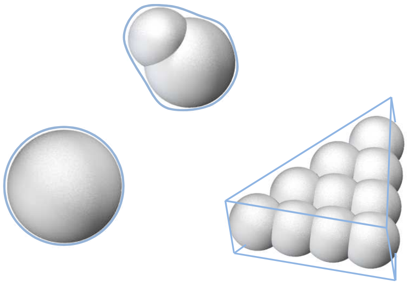


Figure 4: Concept of spherical decomposition.

We experienced that the idea of using cluster of spheres as collision shapes can fit well into parallel computing architectures of the Grpahical Processing Unit (GPU) type, as shown in [24].

V. Forces fields

At each simulation timestep, the force fields acting on each particle must be updated.

In this model, forces are different for metal and non-metal particles [1]. In case of metals, for the j-th particle, forces acting on the particle are: electric force $\boldsymbol{f}_{el,j}$, aerodynamic drag force $\boldsymbol{f}_{a,j}$ and gravity force $\boldsymbol{f}_{g,j}$, plus it can be affected to the contact forces described in the previous paragraphs. Hence, in the DVI of (8), one has:

$$\boldsymbol{f}_{t,j}(\boldsymbol{q},\boldsymbol{v},t) = \boldsymbol{f}_{c,j} + \boldsymbol{f}_{el,j} + \boldsymbol{f}_{a,j} + \boldsymbol{f}_{g,j}$$

On the other hand, forces acting on particles made of non-metal materials are: electric image force $\boldsymbol{f}_{im,j}$, aerodynamic drag force $\boldsymbol{f}_{a,j}$, and gravity force $\boldsymbol{f}_{g,j}$, plus the contact forces if any. In this case it holds:

$$\boldsymbol{f}_{t,j}(\boldsymbol{q},\boldsymbol{v},t) = \boldsymbol{f}_{c,j} + \boldsymbol{f}_{im,j} + \boldsymbol{f}_{a,j} + \boldsymbol{f}_{g,j}$$

An analytical model has been used for computing the electric field $\boldsymbol{E}(x,y,z)$ in the space between the two electrodes; to this end we used the formulas expressed in [2]. The resulting electric field is depicted in Figure 5.

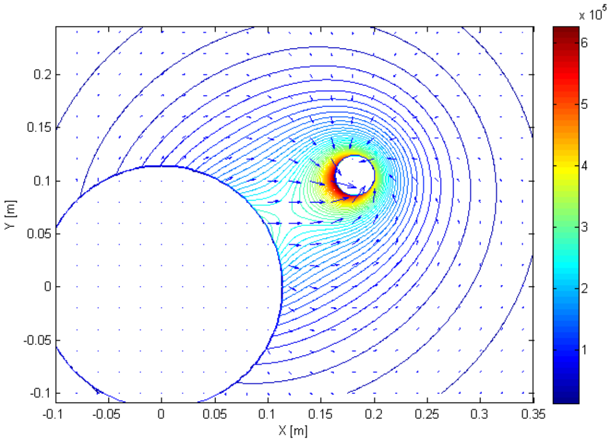


Figure 5: Analytic electric field model, used in run-time computations.

We remark that such analytic model is approximate, and it does not consider the three-dimensional effect of decaying at the sides of the cylinder. For comparison, we performed also a PDE simulation of the electric field using a 3D FEA software, as in Figure 6, and the result is close to the analytical model, at least for the zone of interest.

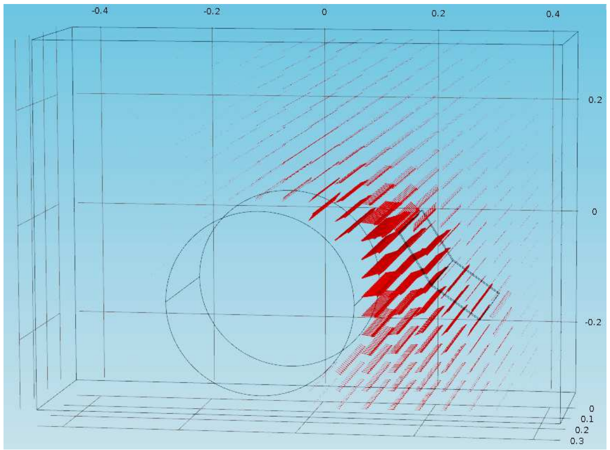


Figure 6: Reference force field, computed with PDE in off-line.

For the sake of brevity, we do not report the formulas for the aereodinamycal drag and for the gravity force, and we focus on forces caused by the electrostatic field. We make the simplifying assumptions that the metal particles discharge instantly as soon as they touch the grounded cylinder, and that charging of the non-metal particles happens instantly as soon they enter the volume in front of the cylinder.

We introduce the vacuum permeability ϵ_0, the relative permeability ϵ_r, the average radius of the particle r. Following [2], the force caused by the electric field acting on metallic particles is:

$$\boldsymbol{f}_{el,j} = 0.832 Q_{\mathrm{metal}} \boldsymbol{E}(x, y, z) \tag{10}$$

where the charge Q_{metal} is

$$Q_{\mathrm{metal}} = \frac{2}{3}\pi^3 \epsilon_0 \epsilon_r r^2 E \tag{11}$$

The non-metal particles are subject to the electric image force:

$$\boldsymbol{f}_{im,j} = \frac{Q^2_{\mathrm{non-metal}}}{4\pi\epsilon_0\epsilon_r(2r)^2} \tag{12}$$

$$Q_{\mathrm{non-metal}} = 3\pi\epsilon_0(2r)^2 E \frac{\epsilon_r}{\epsilon_r + 2} \tag{13}$$

We generate shapes with different densities and different sizes r and different values of ϵ_0 according to a random distribution generator, for instance one can generate flows with 30% of copper particles and 60% of plastic particles, with sizing given by probability distributions. For cases of particles made of microscopic clusters of not-separated materials, we simply apply a weighted average of the forces described above.

VI. Software implementation

The simulation software is implemented in C++ language and uses our Chrono::Engine C++ simulation libraries [25], which implement the DVI method.

To avoid the burden of defining coordinates, sizes and shapes of the collision surfaces, as well motors and joints of the CES machine, we developed a custom add-in in C# language for the SolidWorks software. The add-in allows the designer to model the system with the GUI, as in Figure 7; then one can export the environment into a file that is load and parsed by the simulator before running the simulation. On the other hand, the generation of the particle flow and the simulation loop is controlled directly by C++ statements in the stand-alone simulator, as in Figure 8.

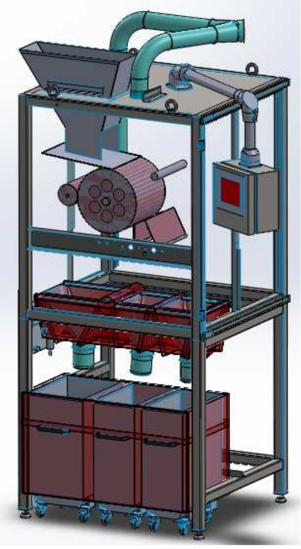


Figure 7: Model of the CES device in the SolidWork parametric CAD. Note that the surfaces that can affect collision detection have been marked as red.

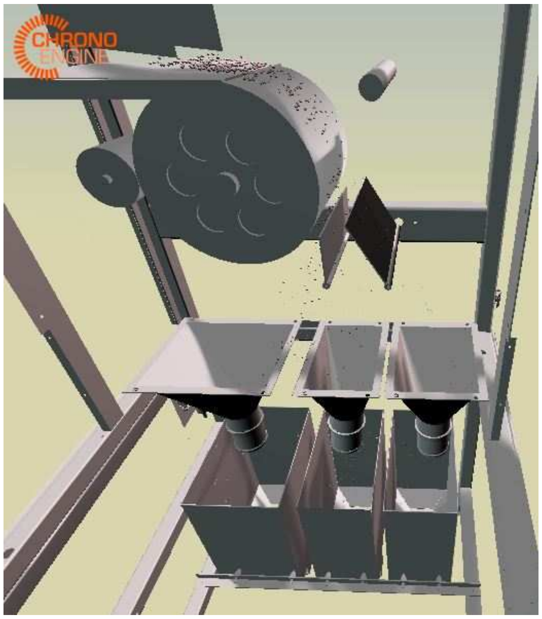


Figure 8: Model of the CES device during the simulation in the OpenGL view of our Chrono::Engine software.

Post-processing of the data is performed via ad-hoc Matlab programs, although the simulator also offers an interactive OpenGL view that can be used to quickly evaluate the results of simulation tests. For the final simulations we use timesteps of 0.001 s, and the number of iterations for the solution of the CCP problem has been limited to 100.

VII. Results

Simulations can be run for different speeds of the drum, for different flow rates, and for different mixtures and size distributions of particles. At each simulation run, particle trajectories are processed by sorting their end point in virtual bins; by splitting the horizontal X direction in evenly spaced bins and by counting the particles that end into such bins, one can build statistical distributions of the output flow, as shown in Figure 9. This can be used to simulate different scenarios of operation of the CES device, and it can be used also to design new versions of the device that could process more material without incurring into inefficiency.

For instance, results in Figures 10 show how the distribution of the outcome, for the same material, increases in variance as the flow rate increases. The reason is that for high

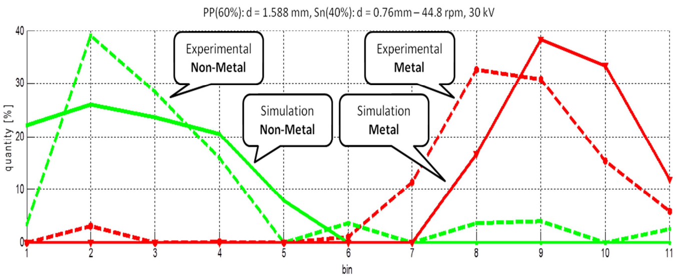


Figure 9: Output distribution showing the separation of metal and plastic particles.

ISBN: 978-1-61208-371-1

flow rate the granular packing starts to be denser, as particles are more likely to be in contact each other, even in stacked layers over the drum, and this introduces more randomness in the process.

VIII. Conclusion and future work

We implemented a custom software, based on our Chrono::Engine simulation middleware, that can perform simulations of granular flows of particles in waste-processing devices. In detail, a CES device has been studied with this tool. The simulation method, being based on an original DVI formulation, permits large timesteps without incurring in numerical instability. Special optimizations have been used in the collision detection algorithms in order to reduce the computational overhead. A custom add-in for the SolidWorks software has been used in order to simplify the workflow, as the user can export collision shapes from the user interface of the CAD.

Future development will address experimental validation and other types of waste processing machines such as the Eddy Current Separator (ECS).

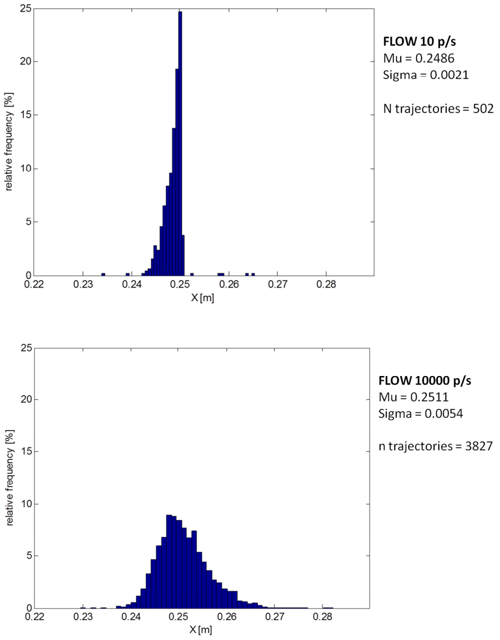


Figure 10: Output distribution of falling particles, for different flow rates.

References

[1] J. Li, Z. Xu, and Y. Zhou, “Application of corona discharge and electrostatic force to separate metals and nonmetals from crushed particles of waste printed circuit boards,” Journal of Electrostatics, vol. 65(4), 2007, pp. 233 – 238.

[2] J. Li, H. Lu, Z. Xu, and Y. Zhou, “A model for computing the trajectories of the conducting particles from waste printed circuit boards in corona electrostatic separators,” Journal of Hazardous Materials, vol. 151, no. 1, 2008, pp. 52 – 57.

[3] H. Lu, J. Li, J. Guo, and Z. Xu, “Movement behavior in electrostatic separation: Recycling of metal materials from waste printed circuit board,” Journal of Materials Processing Technology, vol. 197, no. 13, 2008, pp. 101 – 108.

[4] P. A. Cundall and O. D. L. Strack, “A discrete numerical model for granular assemblies,” Geotechnique, vol. 29, no. 1, 1979, pp. 47–65.

[5] E. Hairer, S. P. Nørsett, and G. Wanner, Solving Ordinary Differential Equations. Springer, 2010.

[6] A. Tasora and M. Anitescu, “A convex complementarity approach for simulating large granular flows,” Journal of Computational and Nonlinear Dynamics, vol. 5, no. 3, 2010, pp. 1–10.

[7] J. J. Moreau, “Standard inelastic shocks and the dynamics of unilateral constraints,” in Unilateral Problems in Structural Analysis, G. D. Piero, F. Macieri, and S. Verlag, Eds. New York: CISM Courses and Lectures no. 288, 1983, pp. 173–221.

[8] ——, “Unilateral contact and dry friction in finite freedom dynamics,” in Nonsmooth Mechanics and Applications, J. J. Moreau and P. D. Panagiotopoulos, Eds. Berlin: Springer-Verlag, 1988, pp. 1–82.

[9] F. Pfeiffer and C. Glocker, Multibody Dynamics with Unilateral Contacts. New York City: John Wiley, 1996.

[10] D. Kinderleher and G. Stampacchia, An Introduction to Variational Inequalities and Their Application. Academic Press, New York, 1980.

[11] D. E. Stewart, “Reformulations of measure differential inclusions and their closed graph property,” Journal of Differential Equations, vol. 175, 2001, pp. 108–129.

[12] D. Stewart and J.-S. Pang, “Differential variational inequalities,” Mathematical Programming, vol. 113, no. 2, 2008, pp. 345–424.

[13] D. E. Stewart and J. C. Trinkle, “An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and Coulomb friction,” International Journal for Numerical Methods in Engineering, vol. 39, 1996, pp. 2673–2691.

[14] M. Anitescu, F. A. Potra, and D. Stewart, “Time-stepping for three-dimensional rigid-body dynamics,” Computer Methods in Applied Mechanics and Engineering, vol. 177, 1999, pp. 183–197.

[15] F. Jourdan, P. Alart, and M. Jean, “A Gauss Seidel like algorithm to solve frictional contract problems,” Computer methods in applied mechanics and engineering, vol. 155, 1998, pp. 31 –47.

[16] C. A. de Coulomb, Théorie des machines simples en ayant égard au frottement de leurs parties et à la roideur des cordages. Bachelier, Paris, 1821.

[17] A. Iserles, H. Z. Munthe-Kaas, S. P. Nrsett, and A. Zanna, “Lie-group methods,” Acta Numerica, vol. 9, 2000, pp. 215–365.

[18] A. Tasora and M. Anitescu, “A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics,” Computer Methods in Applied Mechanics and Engineering, vol. 200, no. 5-8, 2011, pp. 439 – 453.

[19] F. Facchinei and J. Pang, Finite-dimensional variational inequalities and complementarity problems. Springer Verlag, 2003, vol. 1.

[20] M. Anitescu and A. Tasora, “An iterative approach for cone complementarity problems for nonsmooth dynamics,” Computational Optimization and Applications, vol. 47(2), 2010, pp. 207–235.

[21] E. G. Birgin, J. M. Martínez, and M. Raydan, “Nonmonotone spectral projected gradient methods on convex sets,” SIAM J. on Optimization, vol. 10, August 1999, pp. 1196–1211.

[22] E. Coumans. http://bulletphysics.org. [Online]. Available: http://BulletPhysics.org (August 2014)

[23] S. K. E.G. Gilbert, D.W. Johnson, “A fast procedure for computing the distance between complex objects in three-dimensional space,” Robotics and Automation, vol. 4, no. 2, 1988, pp. 193–203.

[24] D. Negrut, A. Tasora, H. Mazhar, T. Heyn, and P. Hahn, “Leveraging parallel computing in multibody dynamics,” Multibody System Dynamics, vol. 27, 2012, pp. 95–117, 10.1007/s11044-011-9262-y.

[25] A. Tasora. Chrono::Engine project, web page. [Online]. Available: www.chronoengine.info (August 2014)

Simulation as a Sensor of Emergency Departments: Providing Data for Knowledge Discovery

Work-in-Progress Paper

Eva Bruballa, Manel Taboada
Tomàs Cerdà Computer Science School
Universitat Autònoma de Barcelona (UAB)
Barcelona, Spain
eva.bruballa@eug.es, manel.taboada@eug.es

Eduardo Cabrera, Dolores Rexachs, Emilio Luque
Computer Architecture and Operating Systems Department
Universitat Autònoma de Barcelona (UAB)
Barcelona, Spain
ecabrera@caos.uab.cat, dolores.rexachs@uab.es,
emilio.luque@uab.es

***Abstract*— Simulation of unusual or extreme situations of Hospital Emergency Departments (ED) makes it possible to get extra knowledge about the behavior of the system which could not be obtained in other way. There is no real data available of such situations, but simulation allows us to obtain this information. In this paper, we show how the data obtained by simulation of scenarios representing special real situations expand the real available data. That provides further information, which will allow us to reach more reliable models of real system behavior, avoiding extrapolation methods, which imply important errors, especially in nonlinear systems as ED. The objective pursued in this ongoing research is to extract the information contained in all this data to observe patterns and translate them into relationships and behavior models about variables which may influence the Hospital Emergency Department's performance and quality of service. A methodology to obtain new knowledge from intensive data, based on the use of the simulator as a sensor of the real system, and so as the main source of data, is proposed. As immediate future work, we intend to prove this methodology in a case of study which aim is to gain knowledge from a specific set of data, obtained through the simulation of a reduced set of scenarios of the real system.**

Keywords—Agent-Based Modeling and Simulation (ABMS); Data-Intensive;Data Mining (DM); Decision Support Systems (DSS); Emergency Department (ED); Knowledge Discovery.

I. INTRODUCTION

A. Simulation as a Decision Support System (DSS)

Hospital Emergency Departments (ED) are a primary care unit in healthcare systems and the main way of patient's admission to hospital. It is one of the most important components of the whole healthcare system and one of the most complex areas of a hospital.

The complexity of the ED service is due to its operation mode, which results from human actions and interactions between the different elements of which it is composed, and simulation based on an Agent-Based Model of the System (ABMS) becomes a powerful tool for its description. Simulation provides a better understanding of the system working way and of the activity of its elements, and it facilitates decision-making to establish strategies for its optimal operation [1][2]. Definitely, it is a way to achieve additional knowledge of the system, by developing inference processes on the variables of interest of the system in order to make predictions about the behavior of these variables under different conditions, based on information obtained from the generated data [3].

B. Simulation as a sensor of the real system

This work in progress aims to obtain knowledge of the system, which could not be obtained without the use of simulation. In fact, the starting idea of this research is to understand the simulation as a "sensor" of the real system [4]. Such idea suggests the hypothesis of the ability to gain knowledge about the ED service behavior from the data provided by the simulation of any possible scenario. Then simulation allows us to obtain data from exceptional, ideal or extreme situations of the ED, such as extraordinary or unusual number of patients or abnormal number of staff (doctors or nurses). This information is very difficult to have in other way because it is impossible to prove these kinds of scenarios in the real system. Data obtained directly from the real system will be complemented with data generated by the simulator. This will allow us to obtain much more refined behavior models of the system, an extra knowledge which would not be possible to obtain without the simulation.

C. An Agent-Based Model of the Emergency Deparment

Previous work carried out by our research group, High Performance Computing for Efficient Applications and Simulation (HPC4EAS), of the Universitat Autònoma de Barcelona (UAB), in collaboration with the ED Staff Team of the Hospital de Sabadell (one of the most important hospitals in Spain, which provides care service to an influence area of 500,000 people, and attends 160,000 patients per year in the ED), has led to the design and development of an agent-based-model of the ED. The model describes the ED's behavior from the actions and interactions between agents, and between them and the ED physical environment. The result is a simplified version of the ED which simulates patients with less acuity level (4 and 5 in the scale of priority and urgency that applies in Spanish hospitals), those causing saturation of the service most of the time. The implementation of the simulator has been done with NetLogo, an Agent-Based Simulation environment well suited for modeling complex systems, and it has been validated and verified, in this its first version [5][6].

The input parameters that characterize each different scenario in the simulation of the real system are the healthcare staff configuration, the number of incoming patients, the derivation percentage of patients and the period of time simulated. In addition, each simulation provides data of the Length of Stay (LoS) of all patients in all ED locations and the number of patients per hour and location [5][6].

D. High Performance Computing (HPC)

There are a great number and variety of simulated agents, and different possible values for the input parameters in the simulator. This results in a large number of different possible scenarios to be simulated. There is no other limitation in the amount of data that can be generated than the computational time needed for the executions of the simulation. Using High Performance Computing (HPC) we can run the simulation for any possible scenario in reasonable computation time. Each execution will generate the corresponding data based on the designed model. Thus, the potential of HPC makes it possible to generate a very large number of data, store this data, process and analyze it to be transformed into information. The effective processing and analysis of this intensive data cannot be done with traditional statistical techniques, and the use of data mining techniques is required to discovering these yet unknown patterns in the data and complete the knowledge discovery cycle. The new knowledge of the system can be manifested through patterns, rules, associations, groups, constraints, trends, etc.

There are previous works in which data mining is applied in different processes within the Hospital Emergency Departments to obtain knowledge about them. But in all cases, the analyzed data comes directly from the real system. These other works can guide us, taking as a reference their objective of study and the algorithms that they have applied in each case. The prediction of the demand for health workers, the influence of different variables on the workload, the expected saturation of the service, the formation of bottle-necks in the connection with the hospital and the creation of a clinical recommender system, or diagnostic support, are the objectives of these referred works [7][8][9][10][11].

The paper is organized as follows: Section II presents the objectives and features of the research, and the proposed methodology; Section III describes a case study which is a reduced example of the proposed working method. Finally, section IV closes the paper with discussion and future work.

II. Objectives and Proposed Methodology

The main purpose of this research is to obtain knowledge from the intensive data generated via simulation of the real system.

The specific objectives to be achieved are:

- To gain knowledge about the system behavior for any possible situation (e.g., epidemics, mass accidents, unexpected increase in the demand for service, etc.).
- To anticipate solutions for unusual situations: Provide prediction models of staff demand and other resources in the service, as a reference for making decisions in exceptional cases.

Our proposal of methodology is represented in Figure 1, which shows the process we will follow in our search from data to knowledge. From bottom to top, the data will be generated by the simulator, processed, transformed and integrated in a data warehouse, and finally analyzed and interpreted using data mining techniques to observe patterns and reach unknown models of system behavior.

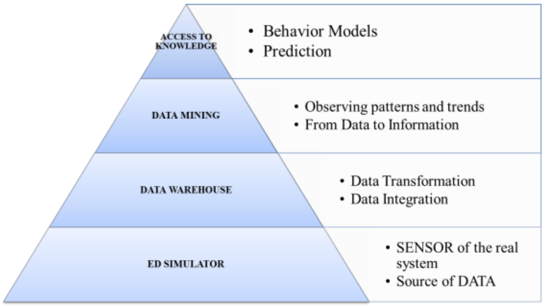


Figure 1. Phases of the proposed methodology

The phases of the proposed work system are described below in more detail.

A. The simulator: Source of Data

Our ED simulator, already verified and validated, allows the generation of data concerning different possible situations, called scenarios, by assigning different values to the input parameters, either representing the more common conditions or other extreme or less likely situations, which are difficult to test in reality. This approach offers the possibility to obtain data for any possible scenario of service.

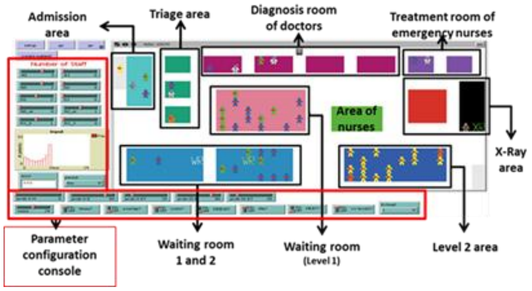


Figure 2. Screenshot with features of the simulator interface.

The present version of the simulator includes six primary areas: admission area, triage boxes, waiting rooms, diagnosis and treatment boxes, and x-ray area; and five different types of active agents: patients, admission staff, nurses, doctors and x-ray technicians (Figure 2). All incoming patients are triaged, but once they have been triaged, only patients identified at triage phase with acuity level 4 and 5 are served at the stage of diagnosis-treatment phase [12].

The input parameters that define each scenario are the staff configuration, the number of incoming patients (workload), the percentage of patient's derivation to other services, and the simulation period. Each possible configuration of the healthcare staff includes: admission personnel, triage nurses, doctors, emergency nurses and X-ray technicians. There is a minimum and maximum number of each type of staff, and two different levels of experience, low and high, labeled as junior and senior, respectively.

Furthermore, different values assigned to the random number generation seed, will affect the way incoming patients arrive into service. The distribution of patients' arrival changes, in each simulated hour, for a specific staff configuration, a given workload and a determined value for patients' derivation. This results in different iterations for the same scenario.

For each of the executions carried out, the simulator provides data concerning healthcare staff configuration, the time required by each of them in each location of the service, the cost of the configuration, and the Length of Stay (LoS) of each patient from their entry into the service until they leave the system, and specifically for each of the areas or places through which it passes. The number of patients per hour, per location, is also registered. Therefore, it is the assigned value for the rest of parameters defining the simulated scenario: the workload, the percentage of derivation for each type of patient, the simulation period and the random seed.

The data obtained directly from the real system is the first approach to obtain information. But, we can generate a much larger number of data using the simulator as a sensor of this real system, and thus, as data source, so that the reliability of the results will be much higher.

B. Transforming Data: The Data Warehouse

After capturing the data, we can try to upgrade its quality, decoding, reformatting and merging it. Extract, Transform and Load (ETL) tools must help us to extract, scrub, transform and integrate all data in a proper way to be stored in a database, data mart or a data warehouse, to make its further analysis possible. Data partitioning, data combining or data summarization is the way to transform data and convert them from operational system format to data warehouse format.

C. Observing Patterns: Data Mining

Stored data contains information that should be treated. We must deal with the analysis of large amounts of data, reason why statistical techniques and traditional methods are inadequate and difficult to apply.

Data mining techniques will be used to analyze such a large amount of data and try to extract the information behind them. Classification, association and clustering methods and algorithms, among others, are some of them. We have to choose which will be the best algorithm to find patterns or trends that will allow us, in turn, make inferences and prediction models through regressions between the observed variables.

D. Acces to knowledge

Finally, this data analysis should lead us to knowledge discovery in terms of behavioral or prediction models. That might be helpful for decision-making in the operational management of the ED by the persons responsible for it.

III. Case Study

In this context and according to the agreed methodology, we will try to obtain knowledge from a set of data already available, obtained from the simulator execution for a reduced set of scenarios, carried out in a previous work within our research group [12].

In this case study, we deal with a set of scenarios characterized by the healthcare staff configuration and the number of incoming patients (workload). Any other input parameter is fixed for all the scenarios. The period of activivy simulated is the equivalent to one day (24 hours), the value for patient derivation percentage is zero, and the random seed is also fixed.

In the referred work [12], the total number of different staff configurations is 28,350. In addition, four different workload scenarios of total incoming patients, during the 24 hours of ED activity, are considered: up to 96, 216, 312, and 408 patients average per day.

In [12], it is described how exhaustive search among all configurations for a given workload was used to find out the optimal ED staff configuration that led to the minimum patient Length of Stay in the ED under a cost configuration constraint (the cost of staff configuration must be less or equal to €3,500) and the amount of human resources (ED Staff) available.

Moreover, in [12], it was implemented an alternative approach to reduce the computation time in simulation executions. This approach consisted of a combination of the Montecarlo method and K-means. After a random configuration execution by the Montecarlo method, the results of the K-means clustering method were that the scenarios were aggregated in some specific regions through the average LoS. Within the region containing the optimal solution, again an exhaustive search was carried out.

Now, we wonder about the possible relation between the set of staff configurations that are aggregated in the same cluster, as a result of this clustering method, applied to all the simulated scenarios. Do the points (configuration) of a specific region follow a specific law? Do they have something in common? Why have these regions and no others appeared? The goal is to observe patterns, similarities, any common feature among the staff configuration of all the scenarios of a same region. Then try to establish relations, and so, new knowledge of the system.

Around 25,000 configurations of the total 28,350 satisfy the considered cost restriction. These are the configurations that were executed for each workload. That is, about 100,000 scenarios were executed. These are the total of scenarios we will deal with for the verification of the proposed methodology, trying to find this possible relation between the aggregated configurations.

The first step in our methodology process has been done, i.e., the data have been generated by the simulator. The second phase will be to transform and integrate these data into a data warehouse format.

This is the point we have reached in this work in progress. We know there is something hidden in the data from these 100,000 scenarios, and we think we have the right methodology to find it. Now, we have to choose the right algorithms and tools to attain this knowledge.

IV. Discussion and Future Work

This ongoing research claims the benefits of simulation as source of data targeting a better understanding of the behavior of ED. Furthermore, the simulator, once tuned with the real system, is able to go beyond purely to replicate reality.

Epidemics, disasters and seasonal fluctuations on the demand of the healthcare service are real examples that could happen any time in the real system, and cause collapse, almost certainly, of the ED. Simulation is a way to obtain knowledge of these kinds of situations that can occur in reality, but cannot be tested in the real system before they happen.

The methodology proposed is supported by the use of high performance computing. HPC allows us to use the simulator as a sensor of the real system, to execute a large number of simulations, each one of them concerning an almost unlimited number of different scenarios, in order to obtain a large amount of data, which in many cases would not be available without the simulation.

Applying data mining techniques on such data will allow us to extract interesting patterns unknown before, thus, information about the real system, and finally knowledge about the behavior of the ED in such situations, which is useful for the decision making process.

As future work, a phase of experimentation with the simulator in its current version will be performed: Execute the simulation in all possible scenarios for different kinds of situations, to obtain intensive data to analyze and prove the proposed methodology to obtain some results.

We have to specify the tools and techniques for the storage and the processing of the data and find the suitable data mining algorithms to analyze it.

Simultaneously, in the same research line, we are working on the model of a new version of the simulator which includes patients with more urgent acuity level, that is, level 1, 2 and 3 in the mentioned scale of priority. Part of the future work is the implementation of this extended version of the simulator, which implies an enhancement that will greatly increase the number of possible scenarios to take into account in our future experimentation. The next experimental phase of the research will consist of executing new simulations with this extended version of the simulator, and again, trying to gain knowledge of the real system from the massive data generated by the improved sensor.

Acknowledgment

This research has been supported by the MINECO (MICINN) Spain under contract TIN2011-24384.

References

[1] A. M. Mancilla, "Simulation: A tool for the study of real systems", Ingeniería y Desarrollo, Universidad del Norte, vol. 6, 1999, pp.104–112. Available from:

http://rcientificas.uninorte.edu.co/index.php/ingenieria/article/view/2226/1443 [Retrieved: July, 2014]

[2] J. Pavón, M. Arroyo, S. Hassan, and C. Sansores, “Simulation of social systems with software agents”, CMPI-2006, Actas del Campus Multidisciplinar en Percepcion e Inteligencia, vol. 1, 2006, pp. 389-400. Available from: http://samer.hassan.name/files/CMPI.pdf [Retrieved: July, 2014]

[3] L. R. Izquierdo, J.M. Galán, J.I. Santos, and R. Del Olmo, “Modeling complex systems using agent-based simulation and system dynamics”, “Modelado de sistemas complejos mediante simulación basada en agentes y mediante dinámica de sistemas”, Empiria: Revista de metodología de ciencias sociales, vol. 16, 2008, pp. 85–112.

[4] A. Choudhary, Northwestern University, USA, Keynote Speaker: “Big Data, exascale systems and knowledge discovery – The next frontier for HPC, Euro-Par Conference, August 2013. Available from: http://www.europar2013.org/upload/Dokumente/Euro-Par-2013-Keynote-Choudhary.pdf [Retrieved: July, 2014]

[5] M. Taboada, E. Cabrera, and E. Luque, “Modeling, simulation and optimization of resources management in hospital emergency departments using the agent-based approach”, Advances in Computational Modeling Research, 2013, pp.1–31.

[6] M. Taboada, E. Cabrera, F. Epelde, M. L. Iglesias, and E. Luque, “Using an agent-based simulation for predicting the effects of patients derivation policies in emergency departments”, Procedia Computer Science, vol. 18, ICCS 2013, pp. 641–650.

[7] C. C. Yang, W. T. Lin, H. M. Chen, and Y.H. Shi, “Improving scheduling of emergency physicians using data mining analysis”, Expert Systems with Applications, vol. 36, no. 2, 2009, pp.3378–3387.

[8] A. Ceglowski, L. Churilov, and J. Wassertheil, “Knowledge discovery through mining emergency department data”, Proceedings of the 38th Annual Hawaii International Conference on System Sciences, IEEE, 2005, pp. 142c–142c.

[9] A. Ceglowski, L. Churilov, and J. Wassertheil, “Combining data mining and discrete event simulation for a value-added view of a hospital emergency department”, Journal of the Operational Research Society, 2006, pp. 246–255.

[10] L. Grigull and W. M. Lechner, “Supporting diagnostic decisions using hybrid and complementary data mining applications: a pilot study in the pediatric emergency department”, Pediatric research, vol.71, no. 6, 2012, pp. 725–31.

[11] W. T. Lin, Y. C. Wu, J. S. Zheng, and M. Y. Chen, “Analysis by data mining in the emergency medicine triage database at a Taiwanese regional hospital”, Expert Systems with Applications, vol. 38, no. 9, 2011, pp. 11078–11084.

[12] E. Cabrera, M. Taboada, M. L. Iglesias, F. Epelde, and E. Luque, “Simulation optimization for healthcare emergency departments”, Procedia Computer Science, vol. 9, ICCS 2012, pp. 1464–1473.

Social Sustainability and Manufacturing Simulation

Defining Social Criteria for a Holistic Sustainability Simulation Approach in Manufacturing Companies

Andi H. Widok
Dept. of Engineering II / Dept. of Modeling and Simulation
Industrial Environmental Informatics Unit
HTW Berlin / University of Hamburg
Berlin / Hamburg, Germany
a.widok@htw-berlin.de

Volker Wohlgemuth
Department of Engineering II
Industrial Environmental Informatics Unit
University of Applied Sciences, HTW Berlin
Berlin, Germany
volker.wohlgemuth@htw-berlin.de

***Abstract* — This paper presents the concept and prototype of a plugin, as part of a software suite (MILAN), aimed to provide technical analysts with the means to simulate various sustainability criteria in manufacturing companies. The plugin is intended to enable analyst to freely define relevant social influence indicators, as well as influence functions and combine them with the existing environmental and economic modeling approach. Various social indicators are, on the one hand still fuzzy, as well as disputed, and on the other hand, dependent on the company's structure. In this regard, the free definition aims to give the modeler the needed flexibility to create a model of his choosing, while also providing him with a structural guideline on how the integration of social criteria is best realized in a holistic sustainability approach. This paper thus addresses the key challenges of the integration of a social perspective in manufacturing simulation and gives an overview over a first implementation of a software that is able to integrate the economic, environmental and social dimension in a single model.**

Keywords – sustainability; discrete event simulation (DES); material flow analysis (MFA); life cycle analysis (LCA); social life cycle analysis (SLCA).

I. INTRODUCTION

From the very start of the modern sustainability debate the idea always included the call for the third pillar of sustainability, i.e., the social pillar, resulting in the term of the triple bottom line [1]. While the classical usages of simulation considering the economical perspective of manufacturing systems and its rather output oriented point of view have already been discussed in much detail [2], the social perspective is, to this day, underrepresented in modern simulation tools. That is, even though many Occupational Health and Safety (OHS) factors, such as ergonomic criteria, or influences through material exposure play an important role in the planning of new and legal compliance of existing entities [3].

Aside from the obvious still debated definition problem of social sustainability [4] the integration of such a perspective is facing other problems, for example the fact that the correlations between humans and their environment are highly dependent on both, which makes a general handling of human resources in a software very difficult and scientifically challenging [5]. Furthermore, the quantification between the relation of the output and the way humans interact with existing production processes may need a great variety of different physical, organizational and psychological algorithms, which indicates a very high modeling effort. This high effort combined with the fact that simulation studies are usually carried out in order to find (economic) optimization potential is contributing to the disregard of social and environmental inclusion. Consequently, in order to promote a more holistic perception, the questions this paper is focusing on are:

- what are possible abstract formulations of social criteria relevant in manufacturing companies,
- how can these criteria be modeled in a way that pays tribute to the great differences between humans and their possible reactions to different strains,
- and how can the modeling effort itself be reduced in order to promote such an integration.

These are the problems this paper addresses and will answer by:

- present the main problems with the integration and related work for this approach (Section II)
- identifying the most relevant aspects of social criteria in producing companies (Section III A),
- categorizing the accorded impacts and deducing criteria for the simulation (Section III B and III C),
- presenting a simulation software for environmental and economic evaluation (Section IV A),
- elaborating the concept and the implementation of a software prototype purposed to integrate the social perspective into the existing simulation software (Section IV B),
- highlight possible results and briefly state strengths and weaknesses of the approach (Section V).

Lastly, an outlook will be given in Section VI.

II. THE PROBLEM IN MORE DETAIL

A. *Motivation*

To understand the driving force behind this paper and the problem with social criteria integration one has to question how producing companies are motivated to produce in a more sustainable way. Aside from intrinsic motivations of given deciders, the two main concepts are legal compliance and the demand of customers, representing top down and bottom up tendencies. Both of these tendencies have inherent difficulties. The main problem with the top down tendency is that regional and international frameworks, as basis for policy decisions, pay tribute to the different regional necessities, hence reflecting the needs and situation of the people in the region. Ultimately the given diversification results in a different prioritization of criteria, which leads to different compliance criteria for the resident entities. These differences allow for a distortion of competition and consequently to a higher prioritization of the economic orientation in order to keep up with the globalized market.

For the bottom up tendency consider that, while time is limited, if a consumer wants to buy a product that is environmentally viable and socially friendly without having to spend too much money, the necessity for an elaborate research develops, in order to find a fitting product. While it is fairly easy to assess the economic value behind the chosen product, when it comes to environmental and social identifying values, it is a difficult task. With regard to firms, and particularly manufacturing companies, reports on the sustainability of their operations rarely include the social dimension. Many companies are issuing corporate reports which stress governance aspects and environmental practices, but tend to overlook the role of the employees or workforce [6][7]. Normally, a detailed analysis of products is impossible to find and thus, in order to make a decision, consumers relay, for example, on brand identification combined with rather current information on how environmental and social friendly the company is or displays itself to be. In other words and from a capitalistic perspective, the steering of the capital by the consumer is not based on the actual environmental and social impacts of the product, but by the little information they can gather about its manufacturing processes and the company itself. The data to assess the environmental and social friendliness of the product itself is not at the consumer's disposition [6]. Naturally a choice regarding the price comes easier than basing the decision on facts that the consumer can hardly evaluate. In order to address the information gap different (modeling and simulation) approaches can be noted, which will be presented in the following.

B. Related Work

Over the last decade the environmental perspective has become more prominent, examples for the focus on the environmental sustainability of production systems can be found in Seliger [8], Andersson [9] and Reinhard et al. [10]. In Thiede [11], one can find a list of simulation tools with a status overview of their features considering sustainability aspects, which gives a broad overview even though the feature list for most tools has already changed. Furthermore, material and energy flow data are under observation in Thiede et al. [12], which can become relevant when considering interaction of human and material, i.e., exposure. The software solutions described were used as references for the meaningful combination of different perspectives.

Most existing simulation software is however not integrating the life cycle approach. It seems that the perception of the system borders of the simulation approach, which logically inhibits the gate to gate focus, is hindering the other. In order to change that and integrate upstream data, two strategies can be observed: on the one hand, through the integration of Life Cycle Analysis (LCA) data (for used material) at least the environmental and some social aspects of the upstream can be integrated, examples can be found in Andersson [9] and Kellens et al. [13], while on the other hand different simulation techniques (for example Discrete Event Simulation (DES) and System Dynamics (SD) and or Agent Based Simulation (ABS)) are combined in order to model and integrate different parts of the life cycle in appropriate and possible detail/granulation. These will logically be integrated once the simulation has finished; see for example Andersson et al. [14]. The combination of these different models is however usually happening via interfaces or meta-models and not integrated in one combined modeling approach, which was the intention when designing the different prototypes of MILAN. In that regard, Widok et al. [15][16] and the elaborations in the following depict the integration of LCA, DES and MFA in a combined modeling approach, where only one model has to be created.

Social criteria are only very rarely elaborated when considering the sustainability of manufacturing system in general [7][17] and when it comes to simulation and or software solutions for these, even less. In Heilala et al. [18], ergonomic criteria are, as part of a social domain, integrated in one simulation approach; Lind et al. [19] displays the findings of the research paper in more detail. The general approach and findings of [18][19] were carefully reviewed for input considering the scope of a possible social domain. The approach defined in Section IV is however intended to go beyond the depicted ergonomic criteria and hence had to be designed more flexible considering very different influences and their respective algorithms. In Makhbul et al. [20] stress at the workplace is analyzed and ergonomic workstation factors categorized. These factors were important for the general handling and served as one reference for the design of the calculation methodology. Implications towards the work performance of following measures can also be found in Yahaya et al. [21]; in the future, it may be possible to integrate the described ideas, even though they only served as reference for the design phase. Detailed analysis of occupational musculoskeletal and mental health with specific focus on production systems can be found in Westgaard and Winkel [3]. They also show an overview over relevant studies and highlight the significance of their findings. The European Agency for Safety and Health at Work Report 2013 is also highlighting OSH risk and trends [22]. A detailed analysis of historic occupational safety measures and trends can be found in Luczak et al. [23]. Examples and guidelines for shift-management and workplace fatigue can be found in [24]. The more general sources [22][23][24] were used as basis for the domain development, while [3] served as guideline for the integration of specific indicators, i.e., which indicator integration was worthwhile and could possibly lead to meaningful results. Furthermore, Sharma [25] presents a case study about a conceptual framework for the improvement of business performance with lean manufacturing and human factors interventions, which served as idea for the post-simulation framework development, i.e., for result interpretation. In addition, a new guideline by the association of German engineers has been published in 2013, depicting the representation and physical strains on humans in virtually modeled manufacturing halls, an analysis is described by Zülch [26] (in German), while it did not influence the development, adaptation in the future may be oriented in order to comply with the formulated standards. Lastly, Zaeh and Prasch [27] are making suggestions for systematic workplace/assembly redesign for aging workforces, which was always considered for future uses of the domain, especially when with regard to the fatigue, ergonomics, skillset and possible differentiations of work performance.

More holistic approaches (needed for the combination of the perspectives) are presented by Omann and Spangenberg [4]. The capital approach for sustainability evaluation is explained in chapter V of the UN report [28], which is also relevant for framework compositions and hence contributed, as Sharma [25] did to the post-simulation framework design for result evaluation. Social capital in relation to quality of life is discussed by Grünberger and Omann [29] and in relation to productivity Reagans and Zuckermann [30], both of these are relevant general evaluation strategies. When trying to incorporate the idea of holistic sustainability approaches one should furthermore consider Gasparatos et al. [31], where the authors list important arguments against the reductionist approach and also directs the attention to possible struggles and problems with their integration. A supplementary holistic design approach is discussed by Spangenberg et al. [32], where most of the already described problems are addressed. While Schneider [17] is giving a specific example for breaking down criteria from a macro management oriented (OECD – Sustainable Development) perspective, to social criteria in firms (see also the OECD report [6] for that matter); this was very important for the general design of the framework composition. Further extensive reviews of social sustainability can be found in Schneider [33] and an extensive literature review of social sustainability assessment methodologies has been published by Benoît and Vickery-Niederman [34]. These papers are also very valuable considering LCA integration and possible SLCA adaptations in the future. Considering SLCA integration further extensive summaries were made by author Jørgensen [35][36], which’s findings will be at the basis for further component development in the SLCA segment.

To sum up, one can note various modeling approaches for social sustainability, but very few actual software implementations when it comes to manufacturing simulation. Furthermore, combinations with different perspectives and focus on holistic perception of social criteria are usually intended for reporting, after careful aggregation of various different sources. Actual implementations are usually conducted with a single focus (for example ergonomic criteria). With this in mind, the following sections will describe what specifically had to be taken into account for the definition of a more holistic oriented implementation of a social domain in the described simulation software.

III. Social Sustainability in the Manufacturing Industry

A. *Understanding Social Sustainability on Company Level, definitions, challenges*

The definition of social sustainability and the deduction of relevant criteria are both far from new; in the last decades various entities have made great efforts to give deciders a stronger foundation on what social sustainability implies. Starting with the World Bank’s sponsored Social Capital Initiative at the beginning of the millennium [37] many international and regional organizations have since created a variety of international policy/reporting guidelines, such as the G4 guidelines of the Global Reporting Initiative (GRI) [7] /monitoring/auditing frameworks and other instruments, such as value chain analysis, social impact assessments and other that all aim for a broad perception and integration of social criteria in sustainability assessment.

When considering these framework approaches and social criteria, the problem arises, that different social criteria will be relevant to different regions, companies and different people. Furthermore, as the sustainability concept has an inherent function to be able to shift in time [16], potential sustainability criteria have to have their qualification flexible in those regards (hinting the change of the model used for their qualification, i.e., iterations of simulations to pay tribute to the change of normative values at the basis of the qualification). In addition to those modeling challenges, Omann and Spangenberg formulated four major challenges on how to assess social sustainability, namely:

- the lack of conceptual clarity (emphasizing definitions to be dependent on countries and entities),
- the complexity (questioning if the concept is even manageable with current organizational and technical means),
- the “bad experience” from the past (1960’s) considering the formulation of normative goals, in order to place social values in relation to economic and environmental goals (see also Colantonio [38]),
- the fact that a stronger integration of social values may question the very foundations of current development models [4], reducing the likeliness of acceptance/introduction.

Many other authors [33][38][39] argue in similar directions, yet the first argument should not be understood as lack of conceptual clarity; this is because the regional/organizational shift of relevant criteria is explainable. If we consider social criteria to be in direct relation to human beings, similar to any human need categorization, regional social sustainability frameworks will represent the state of the needs of the people in that region. This does not necessarily influence the validity of existing frameworks, but only reduces their comparability.

Summarizing one can observe two functions that influence the definition of social sustainability criteria in companies:

- The first differentiation needs to be made considering the people and organizations that are at the basis of the question of what is sustainable (i.e., sustainable for whom, for what, for how long). The definition is thus dependent and pays tribute to the different states of the people and organizations in question.
- The second variance is in relation to manufacturing companies. It is necessary to make a difference between the social impact manufacturing processes have on the people directly involved in them (i.e., the people working for example at a workstation) and the social influences emitted by the production itself.

In addition to these differences, the technology choice needs to be discussed. Consequently confronted with a variety of possible input factors the question poses itself, what are the relevant criteria and how could they be integrated.

B. Categorization of social sustainability aspects on company level

This categorization of social sustainability aspects is oriented on Porter and Kramer's depiction of social impacts of the value chain of companies [40][41]. Extending their description of different criteria and placing them in a manufacturing company perspective (their elaborations are more general), we can note that the main value creating activities for manufacturing companies are operations, inbound and outbound logistics, as well as procurement, while the logistics, procurement and human resource management enable and facilitate the operations in the same way as the firm infrastructure and marketing/sales enable the function of the firm itself.

Given these main branches of the company (including also technology development) it is possible to make a differentiation between:

- the infrastructure, marketing and after sales – being categorized as mainly socio-economic with some socio-institutional aspects,
- the operations, inbound-, outbound logistics, as well as procurement – being categorized as mainly socio-environmental, with natural socio-economic (especially if we consider efficiency) and some social orientation,
- and the human resource management, as well as technology development – being categorized as mainly social orientation with some socio-environmental (due to new technologies).

The according social criteria can be derived from these main categories, as for example energy, water and material usage, emissions and waste, worker safety and labor regulations, hazardous material usage and general ecological impacts, for the operation category. The same derivation (for the other main aspects of the value chain) has already been done a few times and can be reviewed for example in Porter [40]. The idea behind this division is one can now understand where existing simulation approaches have high correlations.

C. Definition of social criteria relevant to the simulation of manufacturing companies

The main thesis that was described in Section III B is that many of the social impacts at operation's level and generally in the primary value creating activities of manufacturing companies (mainly operations, but also inbound/outbound logistics, procurement) have high correlations with existing DES and ABS modeling approaches. This is because the social impacts are almost directly linked to either the materials in usage (socio-environmental orientation and socio-economic if we consider efficiency aspects) or the people working and facilitating the functioning of the workstations (social orientation, OHS aspects). One can thus note, that a limited integration of a social perspective in existing economic, environmental orientated manufacturing simulation models is possible without having to change the model itself drastically, opening the possibility for an integrated holistic modeling approach. In that regard the choice for a first set of resulting criteria was based on the described social impact criteria from these aspects. Also note, that the indicated social impacts are not complete, further elaborations of social impacts at midpoint level can be found in [35]; the given lists were simply intended to demonstrate examples and their general categorization. Furthermore, as social impact criteria have been categorized as socio-environmental, basically representing the original sustainability perception of conservation, and as socio-economic, the correlations between the pillars of sustainability become even more apparent.

While this only considers a limited view on social impacts (reducing the perception to the manufacturing processes), it is important to note, that the life cycle approach can consequently be incorporated through the integration of social life cycle assessment (SLCA) data for the materials in usage and general upstream input data. To clarify this, consider a classical manufacturing model, which depicts the system borders at the in- and output flows before and after the existing manufacturing processes. This model has and produces little life cycle knowledge but only considers the manufacturing aspects (which depending on the used materials make more or less of the overall impact). It is however possible to have a combination of classical DES/ABS manufacturing approaches in combination with life cycle assessment (LCA) upstream data (and possibly even downstream data, depending on the modeling approach), as has been demonstrated among others by Kellens et al. [13], Andersson et al. [14] and Widok et al. [16] for the environmental LCA (ELCA) part. Taking SLCA parallel to ELCA, it can thus potentially be used for two different overall purposes, already discussed in 1997:

- to compare the social impacts of two comparable products or services (or compare a product or service against a standard – which is what we want to achieve in the future),
- to identify hot spots or improvement potentials in the life cycle of the product or service [34].

There are different approaches, which use different simulation techniques in order to model and simulate the bigger picture (apart from the LCA integration), i.e., changes in customer demand (due to marketing for example) or the abstract term of innovation (tech. dev.) can be modeled and simulated using system dynamics, examples can be found in Georgiadis and Besiou [42], as well as, Venkateswaran and Son [43] and then combinations of these can be found in Andersson [9], Rabelo et al. [44] and Jain et al. [45] using also LCA.

Since the integration of social issues into LCA, SLCA methodology now advanced to the point where it is left with many of the same unresolved issues as ELCA (see also Jørgensen [36]). These include:

- the challenges of tracking down site-specific data,
- the challenges of integrating location sensitive information,
- the challenges of integrating information collected at different scale (from general sectors to specific unit processes),
- developing characterization methods [34].

Yet, even though the data situation is always a problem to be taken seriously, the concept of integrating social impacts for the production processes in the simulation model, as was already done for environmental criteria, while integrating social impacts for the different other life cycle stages through SLCA data, was found worthwhile and is at the basis of the depicted prototype that will be elaborated in the following.

IV. Integration of Social Criteria in the DES/MFA/LCA Simulatior

A. The basics of the simulation software MILAN

The software MILAN has its origin in 2001, when the conviction began, that the combination of material flow analysis (MFA) with existing simulation approaches was worthwhile [46].

The concept of combining discrete event simulation and material flow analysis in a component-based approach was then presented in 2006 [47] and its re-implementation on .NET basis was elaborated in 2009 [48]. The integration of DES with the material flow perspective of MFA within a single integrated modeling approach was made possible in order to strengthen the perception of correlations between environmental and economic questions. Based on the dynamic, tactic and strategic character of the simulation approach itself, the perception of material and energy flows, which was at that point not part of the operative level, was intended to be given a more strategic, proactive tendency.

In 2011, a capital measurement approach for a more holistic sustainability perspective was presented, hinting the beginning of the integration of the life cycle approach [15].

In 2012, the ELCA integration was elaborated and the integration of the social perspective was discussed in the outlook [16]. Since then the simulation software has constantly been enhanced with new features and has been used for case studies with companies in Germany and Switzerland (under a different scope with a MILAN core). The most basic components of the software are:

- a simulation core (central simulation service, interfaces and abstract base classes for models),
- a bundle for discrete event simulation (specific for DES, with scheduler, timing aspects, etc.),
- stochastic distributions (e.g., Bernoulli, Exponential, etc., to generate streams of numbers),
- a graph editor (enabling the visual representation and manipulation of models),
- property editors (facilitating the parameterization of model entities and given metadata),
- a reporting suite (creating the simulation results and preparing charts depending on the scope),
- the material management (for the creation, management of materials, batches, bills of materials),
- the material accounting (by its means it is possible to show, save and manage material and energy bookkeeping resulting from the simulation. The bookkeeping is realized using accounting rules, which can be added to all discrete events in combination with relevant model components),
- a LCA browser, which enables an easy, string-based search and the subsequently integration of LCA material data, enabling life cycle inventory (LCI) and LCA in the simulation and the results.

For more information about the technical aspects of the simulation software, see Jahr et al. [48].

B. The social perspective prototype

The main components of the social domain are visualized in Figure 1 below. The first two lines represent the social domain layer, the two lines below that represent sustainability related components, the elements in the fifth line from the top represent DES relevant components, while the last two stand for technical features facilitating the general functioning of the software.

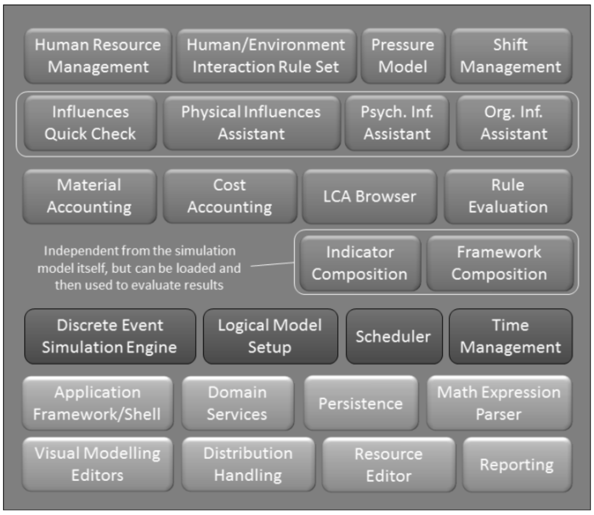


Figure 1. Abstracted overview of the main components of and needed for the social domain

When the social prototype was first conceptualized the two attributes considered to be the most important were:

The component architecture: aside from normal component-development reasons, such as high reusability and the easier understanding of the code, through clear, small packages, this also means that the usage of the social perspective is not enforced, i.e., it is possible to model social aspects through the software, but one does not have to. The software also allows to only build DES simulation models and not integrating MFA or LCA, but if the data is existing and the intention is to have a strong, holistic model, one can use the different techniques combined in one modeling approach and only a single model has to be created, incorporating the methodologies.

The free definition of influences: this is based on the conviction that social criteria, as well as their measurement, are still disputed. Based on this, it was decided that an open definition of different influences would be made possible, with different editors for the most common influences (physical, organizational, psychological), incorporating current knowledge considering the measurement of such criteria and their impact on human resources over time. These impacts however are not validated by the tool itself, i.e., the reasonableness of the defined influences and their impact lays currently with the modeler (except for logically excluding behavior).

The main features of the social component will be elaborated in the following (see Figure 1 for reference).

Human resource management/editor: based on normal resource management approaches a functionality was created to split existing resources into three different resource types, 1) human resources, 2) tools and 3) usable resources. Each of these resource types has a different editor, facilitating for the human resources possibilities to adjust for skill set, integration of distributions considering illness or weaknesses (also usable for the modeling of elderly workers and adjustment of strengths in the following) and many others. Furthermore, a new pooling mechanism was created based on a list of categories attributable to the existing resources, for example one could attribute a human resource different locations, workplaces and others (also at different time steps). The categorization/pooling then manages for example the availability of the resource.

Shift management module: the shift management is basically a standard shift planning tool, which is used for both, the workstations, i.e., one can define if production processes are continuously or with breaks for a period of time. This is of course relevant for the warm up phases and different states of the workstations. Furthermore, the shift management is used to attribute different human resources to their respective work-related entities. These could be different workplaces (although a workplace editor is yet to be integrated). For the moment, these are the respective workstations (i.e., the rather classic DES workstations model entities). In that regard a classical resource usage over time can be calculated and attributed to locations, as well as workstations and other categories that were defined in the resource categorization. In addition, the possibility is given to attribute a type of influence on the resource over time. These possible strains can be either physical, or otherwise, depending on the modeled influences through the different influence editors and the following choice of the modeler.

Social influence layer: in this layer, different editors for different types of influences were developed, the main differentiation is between physical, psychological and organizational influences, where the physical editor guides the definition of a physical influence through possible input choices (strong relation to German OHS guidelines, as in strains for lifting, crouching, carrying, but also general, as in workload dependent, biological interaction, noise, etc.) all of the possible choices are backed up with known formulas for the development of the influence (such as the physical basics of noise development or basics for the development of particulate matter in production processes), as well as known limit values considering the strain on an average human being. The psychological editor does currently have a completely free definition of influences, while different types are suggested, no choices of formulas is, but rather the definition of a type is mandatory, which can subsequently be used in the rule set editor. The same procedure is implemented for the organizational influences. Even though many studies were incorporated in a knowledge basis for these components (a systematic review of occupational musculoskeletal and mental health studies for production systems can be found in [3]), the definition of the non-physical influences was implemented without structural restriction.

Human environmental influences rule set component: this is the second key element for the integration of the social criteria. In this element one can choose from the previously defined social influences and by the usage of a math expression parser and the existing model of shifts and or the production system (i.e., the workstations), combine time with influences to create an impact over time. Different dose concepts were evaluated in that regard, which are also integrated in a knowledge base and selectable (note: the tool is only making a basic validation for reasonable combination choices). Once an influence is attributed to a shift or a workstation, the simulation is then calculating an impact of the indicated influence over time.

V. Possible Results and brief Discussion of strengths and weaknesses

The social component is currently being tested in two use cases, respectively in one plastic processing company and one company that manufactures technical boilers. Aside from the classic results, such as new information on resource usage, failure times, etc. new information considering workload and strains on human resources are expected as results. Different scenarios are still under evaluation (noise, repetition, material exposure influences). What can however be observed, is that the bringing into focus of social aspects, already created ripple effects, considering the perception and the management of social impacts.

In light of the current feedback, we argue that the main weaknesses/challenges of this approach (bad data situation, privacy issues, fear of abuse, wrong evaluations) are manageable and that it is similar as with the environmental sustainability assessment in the past, i.e., that the best way to address the complexity is by making one step at a time, without losing focus of the needed flexibility and adaptability of further models, simulations and their result qualification. This approach is intending to do just that. While others have shown that different social aspects can be integrated in DES manufacturing approaches, it is our intention to create the scientific basis for the step by step integration of new impact criteria, by delivering results of successful integration and evaluation of social criteria through the depicted method in the future. The concept for a worthwhile integration of SLCA criteria is currently being worked on.

VI. Conclusion and Outlook

Last year's Amnesty International Report [49], titled "the dark side of migration" was discussing the exploitation of humans as workforce under inhuman conditions. While this very terrifying problem is less occurring in western countries, it is common sense, that as long we cannot track and measure the social impacts of production processes, it is less likely that consumers will be empowered to choose social friendly created products, and hence not be able to steer their capital accordingly.

Even though we agree with the conclusion of Gasparatos et al. [31], considering methodological pluralism (very simplified: more is not necessarily better), the key idea of the approach in this paper is the attempt of the integration and ability to put different perspectives in correlation. It is clear that the social aspects have yet to mature in their scientific provability, yet potentials can clearly already be indicated. This is what the tool already delivers as result, potentials compared to limit values (i.e., elevated by x%, without qualifying beyond stating that it is a positive or negative tendency and putting it into context).

The main arguments against the integration of social criteria are usually their fuzziness and the fact that every human is different. These points are valid, however the main aspects of human beings are not so different, as a variety of studies suggest (see Westgaard and Winkel [3]). Of course, it is complicated to derive exact numbers, but that is where the free definition of influences comes into play, by allowing for the modeling of workers, as well as the impact on different levels. So while the presented approach is far from scientifically established, its purpose is more to promote the re-integration of social values in existing manufacturing processes. Human development author and activist Max-Neef mentioned in his keynote at Zermatt Summit 2012 that sustainability has been misused to promote rather economical concepts than actually bringing the essence of what sustainability incorporates into prominence, hence it is the intention of this paper to clarify that the deficit of social integration in these regards can be overcome.

References

[1] J. Elkington, "Towards the suitable corporation: Win-win-win business strategies for sustainable development," California management review, vol. 36, iss. 2, 1994, pp. 90-100.

[2] J. Banks, J. Carson, B. L. Nelson, and D. Nicol, Discrete-event system simulation, 4th edition, Upper Saddle River. New Jersey, USA, 2005.

[3] R. H. Westgaard and J. Winkel, "Occupational musculoskeletal and mental health: Significance of rationalization and opportunities to create sustainable production systems - A systematic review," Applied Ergonomics, vol. 42, iss. 2, Elsevier Ltd. and The Ergonomics Society, 2011, pp. 261-296.

[4] I. Omann and J. H. Spangenberg, "Assessing Social Sustainability - The Social Dimension of Sustainability in a Socio-Economic Scenario," Proceedings of the 7th ISEE, Barcelona, Spain, 2002.

[5] A. H. Widok and V. Wohlgemuth, "Simulating Sustainability," Proceedings of the 27nd Int. Conference Environmental Informatics, Shaker Verlag, Hamburg, Germany, 2013, pp. 514-522.

[6] Organization for Economic Cooperation and Development (OECD), "Measuring material flows and resource productivity," OECD Report, Paris, France, 2008.

[7] Global Reporting Initiative (GRI), "G4 Sustainability Reporting Guidelines Implementation Manual," GRI Report, Netherlands, 2013.

[8] G. Seliger, Sustainable Manufacturing, 1st edition, Springer Verlag, Berlin, Germany, 2012.

[9] J. Andersson, Environmental Impact Assessment using Production Flow Simulation, Sweden, 2014.

[10] J. Reinhard, R. Zah, and V. Wohlgemuth, P. Jahr, "Applying Life Cycle Assessment within Discrete Event Simulation: Practical Application of the Milan/EcoFactory Material Flow Simulator," Proceedings of the 27nd Int. Conference Environmental Informatics, Hamburg, Germany, 2013, pp. 532-542.

[11] S. Thiede, Energy Efficiency in Manufacturing Systems. 1st edition, Springer-Verlag, Germany, 2012.

[12] S. Thiede, C. Herrmann, and S. Kara, "State of Research and an innovative Approach for simulating Energy Flows of Manufacturing Systems," Proceedings of the 18th CIRP International Conference on Life Cycle Engineering, 2011, pp. 45-48.

[13] K. Kellens, W. Dewulf, M. Overcash, M. Z. Hauschild, and J. R. Duflou, 2011. "Methodology for systematic analysis and improvement of manufacturing unit process life-cycle inventory (UPLCI)," International Journal of Life Cycle Assessment, vol. 17, iss. 1, 2011, pp. 69-78.

[14] J. Andersson, A. Skoogh, and B. Johansson, "Evaluation of Methods used for Life-Cycle Assessments in discrete event simulation," Proceedings of the 2012 Winter Simulation Conference, Edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, IEEE, Berlin, Germany, 2012, pp. 135-141.

[15] A. H. Widok, V. Wohlgemuth, and B. Page, "Combining Event Discrete Simulation with Sustainability Criteria," Proceedings of the 2011 Winter Simulation Conference, IEEE, Phoenix, USA, 2011, pp. 859-870.

[16] A. H. Widok, L. Schiemann, P. Jahr, and V. Wohlgemuth, 2012. "Achieving Sustainability through the Combination of LCA and DES integrated in a Simulation Software for Production Processes," Proceedings of the 2012 Winter Simulation Conference, IEEE, Berlin, Germany, 2012, pp.264-276.

[17] R. Schneider, "Measuring Social Dimensons of Sustainable Production," Measuring Sustainable Production, Chapter 4, OECD Publishing, Paris, France, 2008.

[18] J. Heilala, S. Vatanen, H. Tonteri, J. Montonen, S. Lind, B. Johansson, and J. Stahre, "Simulation-based Sustainable Manufacturing System Design," Proceedings of the 2008

Winter Simulation Conference, IEEE, USA, 2008, pp. 1922-1930.

[19] S. Lind, B. Johansson, J. Stahre, C. Berlin, A. Fasth, J. Heilala, K. Helin, S. Kiviranta, B. Krassi, J. Montonen, H. Tonteri, S. Vantanen, and J. Viitaniemi, "SIMTER - A Joint Simulation Tool for Production Development," VTT Working Paper, No. 125. Espoo, Finland, 2009.

[20] Z. M. Makhbul, N. L. Abdullah, and Z. C. Senik, "Ergonomics and Stress at Workplace: Engineering Contributions to Social Sciences," Jurnal Pengurusan, vol. 37, iss. 1, 2013, pp. 125-131.

[21] A. Yahaya, N. Yahaya, A. T. Bon, S. Ismail, and T. C. Ing, "Stress level and its influencing factors among employees in a plastic manufacturing and the implication towards work performance," Elixir Psychology, vol. 41, 2011, pp. 5923-5941.

[22] European Agency for Safety and Health at Work (EASHW), "Green jobs and occupational safety and health: Foresight on new and emerging risks associated with new technologies," EU Report, Spain, 2013.

[23] H. Luczak, O. Cernavin, K. Scheuch, and K. Sonntag, "Trends of Research and Practice in Occupational Risk Prevention as Seen in Germany," Industrial Health, vol. 40, iss. 2, 2002, pp. 74-100.

[24] Department of Labour New Zealand, "Managing shift work to minimise workplace fatigue - A Guide for Employers," Crown, Wellington, New Zealand, 2007.

[25] R. Sharma, "Conceptual Framework for Improving Business Performance with Lean Manufacturing and Successful Human Factors Interventions - A Case Study," International Journal for Quality Research, vol. 6, iss. 3, 2012, pp. 259-270.

[26] G. Zülch, „Ergonomische Abbildung des Menschen in der Digitalen Fabrik – Die neue VDI-Richtlinie 4499-4," Simulation in Produktion und Logistik 2013, edited by Dangelmaier W., Laroque C. and Klaas A., Paderborn, Germany, 2013, pp. 53-60.

[27] M. F. Zaeh and M. Prasch, "Systematic workplace and assembly redesign for aging workforces," Production Engineering, vol. 1, iss. 1, 2007, pp. 57-64.

[28] United Nations (UN), Indicators of Sustainable Development: Guidelines and Methodologies, 3rd edition, United Nations. New York, USA, 2007.

[29] S. Grünberger and I. Omann, "Quality of Life and Sustainability. Links between Sustainable Behaviour, Social Capital and Well-being," Proceedings of the 9th Biennial Conference of the European Society for Ecological Economics (ESEE), Istanbul, Turkey, 2011.

[30] R. Reagans and E. W. Zuckermann, "Network, Diversity, and Productivity: The Social Capital of Corporate R&D Teams," Organization Science, vol. 12, iss. 4, 2012, pp. 502-517.

[31] A. Gasparatos, M. El-Haram, and M. Horner, "A critical review of reductionist approaches for assessing the progress towards sustainability," Environmental Impact Assessment Review, vol. 28, iss. 4-5, 2007, pp. 286-311

[32] J. H. Spangenberg, A. Fuad-Luke, and K. Blincoe, "Design for Sustainability (DfS): the interface of sustainable production and consumption," Elsevier Ltd., Journal of Cleaner Production, vol. 18, iss. 15, 2010, pp. 1485-1493.

[33] S. McKenzie, "Social Sustainability: Towards some definitions," Hawke Research Institute, Magil Australia, Working Paper Series, No. 27, 2008.

[34] C. Benoît and G. Vickery-Niederman, "Social Sustainability Assessment Literature Review," White Paper, The Sustainability Consortium, Arizona State University and University of Arkansas, USA, 2010.

[35] A. Jørgensen, A. Le Bocq, L. Narzarkina, and M. Hauschild, „Methodologies for Social Life Cycle Assessment," International Journal of Life Cycle Assessment, vol. 13, iss. 2, 2007, pp. 96-103.

[36] A. Jørgensen, Developing the Social Life Cycle Assessment: Adressing Issues of Validity and Usability, Thesis (PhD), Lyngby, Denmark, 2010.

[37] C. Grootaert, T. van Bastelaer, S. Assaf, T. Rossing Feldman, and G. Ochieng, "The Initiative on Defining, Monitoring and Measuring Social Capital," Social Capital Working Paper Series, Social Capital Initiative Working Paper No. 1. Washington, DC, USA, 1998.

[38] A. Colantonio, "Social sustainability: an exploratory analysis of its definition, assessment methods metrics and tools". In EIBURS Working Paper Series, 2007/01, Oxford Brooks University, UK, 2007.

[39] N. Pelletier, E. Ustaoglu, C. Benoit, and G. Norris, "Social Sustainability in Trade and Development Policy," European Union, Ispra, Italy, 2013.

[40] M. E. Porter, The Competitive Advantage, NY Free Press, 1985.

[41] M. E. Porter and M. R. Kramer, "The Link Between Competive Advantage and Coporate Social Responsibility," In: Harvard Business Review, vol. 84, iss. 12, 2006, pp. 78-92.

[42] P. Georgiadis and M. Besiou, "Environmental and economical sustainability of WEEE closed-loop supply chains with recycling: a system dynamics analysis," Springer Verlag, International Journal of Advanced Manufacturing Technology, vol. 47, iss. 5-8, 2010, pp. 475-493.

[43] J. Venkateswaran and Y.-J. Son, "Hybrid system dynamics - discrete event simulation based architecture for hierachical production planing," Taylor & Francis Group, International Journal of Production Research, vol. 43, iss. 20, 2005, pp. 4397-4429.

[44] L. Rabelo, M. Helal, and A. Jones, H.-S.. Min, "Enterprise Simulation: A Hybrid System Approach," International Journal Computer Integrated Manufacturing, vol. 18, iss. 6, 2005, pp. 498-508.

[45] S. Jain, S. Sigurðardóttir, E. Lindskog, J. Andersson, A. Skoogh, and B. Johansson, "Multi-Resolution Modeling for Supply Chain Sustainability Analysis," IEEE, Proceedings of the 2013 Winter Simulation Conference, 2013, pp. 1996-2007.

[46] V. Wohlgemuth, L. Bruns, and B. Page, „Simulation als Ansatz zur ökologischen und ökonomischen Planungsunterstützung im Kontext betrieblicher Umweltinformationssysteme (BUIS)," in L. M. Hilty; P. Gilgen (Eds.): 15th International Symposium Informatics for Environmental Protection, 2001.

[47] V. Wohlgemuth, B. Page, and W. Kreutzer, "Combining discrete event simulation and material flow analysis in a component-based approach to industrial environmental protection," Elsevier Ltd., Environmental Modeling & Software, vol. 21, iss. 11, 2006, pp. 1607-1617.

[48] P. Jahr, L. Schiemann, and V. Wohlgemuth, "Development of simulation components for material flow simulation of production systems based on the plugin architecture framework EMPINIA," Proceedings of the 23nd Int. Conference Environmental Informatics," 2009, pp. 151-159.

[49] Amnesty International, "The Dark Side of Migration - Spotlight on Quatar's Construction Sector ahead of the World Cup," Amnesty International Report, London, UK, 2013.

An Agent-Based Financial Market Simulator for Evaluation of Algorithmic Trading Strategies

Rui Hu

Quantica Trading
119 King St. West Suite 300
Kitchener, Ontario, Canada
Rui@quanticatrading.com

Stephen M. Watt

Computer Science Department
University of Western Ontario
London, Ontario, Canada
Stephen.Watt@uwo.ca

Abstract—**Algorithmic trading strategies are most often evaluated by running against historical data and observing the results. This limits the evaluation scenarios to situations similar to those for which historical data is available. In order to evaluate high frequency trading systems in a broader setting, a different approach is required. This paper presents an agent-based financial market simulator that allows the exploration of market behaviour under a wide range of conditions. Agents may simulate human and algorithmic traders operating with different objectives, strategies and reaction times and market behaviour can use combinations of simulated and historical data. The simulator models the market's structure, allowing behaviours to be specified for market makers and liquidity providers and other market participants. The primary use of the system has been in the evaluation of algorithmic trading strategies in a corporate setting, but other uses include education and training as well as policy evaluation.**

Keywords*–*Agent-Based Simulation, Financial Markets, High Frequency Trading

I. Introduction

Algorithmic trading has grown rapidly around the globe and has dramatically changed how securities are traded in financial markets. According to a few reports [1]–[3], more than 50% of the volume of U.S. equity markets in recent years has been generated by algorithmic trading. In order to manage risk exposure and optimize profit, algorithmic trading strategies are generally evaluated for correctness and performance before launching them in real markets. This is carried out in practice through simulators. Existing simulators significantly rely on real-time market data or historical data, typically recorded from actual market for the purpose of back-testing trading models during their development cycle. While this can provide traders with valuable information, there are a number of pitfalls. First, live market data is not always available, which restricts the use of simulators to certain market hours. In addition, the trading strategies tested do not have any impact to the market as they can only follow the trend and their orders are simply executed based on the current market conditions. Similar issues also exist in back-testing approaches where trading strategies are tested against existing data set with the problematic assumption that the orders would not have changed the historical prices if they were executed in the real market. Moreover, this approach is limited by potential over-fitting. By refining the parameters of a trading strategy on a particular period of historical data, the results can become skewed and produce returns that can never work again. Last, but not least, existing simulators typically do not provide a standard protocol for interaction with users. Instead, they require skills in specific programming languages and demand trading strategies to be implemented on top of proprietary Application Program Interfaces (APIs). This can restrict the evaluation of trading strategies to a single simulation environment. This issue can be seen in TT Sim [4].

We are motivated by the question of how to design and implement a simulator that can support market simulation research and is suitable for evaluation of algorithmic trading strategies. The simulator should be independent of any particular data feed and be able to provide a realistic testing environment by reproducing certain phenomena of a real market. This would be beneficial to strategy testing as some phenomena, such as the flash crash [5], are hard to predict and do not occur often in real markets. Such phenomena can, however, be imitated easily in the simulation environment. Multiple users should be able to connect to the simulation server at the same time, allowing them to not only assess the viability of their trading strategies using pre-defined market conditions, but also to create very specific ones that suit their needs.

We present such a simulator that is intended to be useful for evaluating trading strategies. It supports a variety of security types, including equities, futures, foreign exchange, and options. The simulator may run a market consisting exclusively of simulated (human or robotic) trading agents using different trading strategies (e.g., to study algorithms or market effects), or external participants (typically human) may log in and interact with the simulated market (e.g., for training). Participants (logged in users or simulated trading agents) can submit both limit and market orders with different time-in-force, allowing them to interact with the simulation environment as if they were trading in a real market. At the same time, the simulator adopts the Financial Information eXchange (FIX) protocol [6], an open standard that is used extensively by global financial markets. This allows multiple users to interact with the simulation environment simultaneously and independently. In contrast to other simulators that require testing trading strategies to be built on top of proprietary APIs, our simulator uses open protocols so can be integrated easily into their systems with little modification. The simulator is able to run in two different settings, each of which is useful in certain scenarios.

The first setting uses simulated trading agents, each of which is able to adapt and react to real-time market events by following selected pre-defined strategies. All of the agents are configurable. By adjusting their configurations, we can create very specific market conditions that would occur only rarely in a real market. In addition, the agent-based simulator is able to run at any time as the data are generated by the computerized agents. The agent-based simulator is useful in that it allows

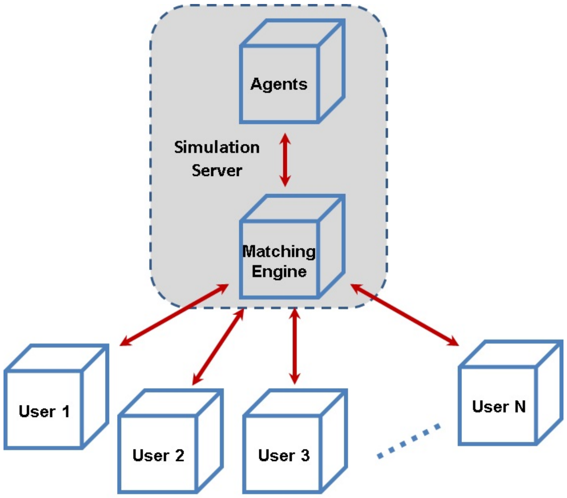
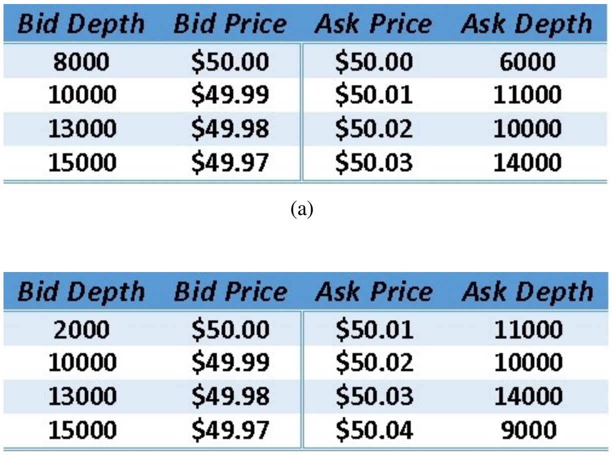


Figure 1: Agent-based simulator architecture.



Bid Depth	Bid Price	Ask Price	Ask Depth
8000	$50.00	$50.00	6000
10000	$49.99	$50.01	11000
13000	$49.98	$50.02	10000
15000	$49.97	$50.03	14000

(a)

Bid Depth	Bid Price	Ask Price	Ask Depth
2000	$50.00	$50.01	11000
10000	$49.99	$50.02	10000
13000	$49.98	$50.03	14000
15000	$49.97	$50.04	9000

(b)

Figure 2: Order book snapshots: (a) Before an order match (b) After an order match.

users to create desired market conditions where they can test their trading strategies whenever it is needed.

In the second setting, the simulator receives live market data from real exchanges and broadcasts this data to each user. Orders that are submitted by users are executed based on the current market conditions. This type of simulator is claimed by some to be more realistic. Meanwhile, the real market data simulator, in practice, has access to all the products available in the markets, while in agent mode this is not feasible unless there is a further configuration. We provide both these settings to users to evaluate their trading strategies, and give them the freedom to choose the one that is most suitable for their needs. A demo of our simulator is available for download at [7].

The remainder of this article is organized as follows: Section II presents the architecture of the simulator. Section III discusses the more important design and implementation details. Section IV illustrates a variety of scenarios where the simulator is found useful. Section V compares the current approach with related work. Finally, Section VI presents conclusions and discusses directions for future work.

II. SIMULATOR ARCHITECTURE

We now present the framework of the simulator which allows logged in traders to interact with artificial intelligent agents. The framework currently consists of a matching engine, a communication interface and a variety of simulated trading agents. Figure 1 shows a high-level overview of the architecture of the simulator. In the setting of using live market data, the agents will be replaced with data streams from real exchanges.

A. Matching Engine

The core of the simulator is a matching engine that accepts in-bound orders from both logged in users and computerized agents. It maintains a number of order books, each of which records the interest of buyers and sellers in a particular security and prioritizes their orders based on their price and arrival time (e.g., first come first serve). Buy orders with the highest price and sell orders with the lowest price are placed at the top. This centralized order system continuously attempts to pair buy and sell orders and trades are announced when certain matching rules are satisfied.

Matching rules are implemented based on continuous double auctions. An incoming buy order is first compared with the best sell order in the order book. If there is a price match, a trade will be generated at the best price. The matching engine will then send execution reports to the issuers of the two orders and announce a trade to all market participants. If the new order is not completely filled, the matching engine will try to fill the remainder of the order with the next highest ranking order available from the opposite side. This procedure is continued until either the new order is completely filled or there are no more matches. Figure 2 shows snapshots of an order book before and after an order match.

The matching engine also publishes quote updates to all subscribers. This is a major challenge as the matching engine must be able to process a high throughput of data with very low latency and broadcast updates to many subscribers at once. This requires us to design and implement a highly efficient financial information protocol for communications.

B. Communication Interface

The Financial Information eXchange (FIX) protocol is an electronic communication protocol, first introduced in 1992, whose primary objective is to exchange real-time stock trading data between entities. It has experienced a tremendous growth over the past years and has become the *de facto* communications standard in global financial markets. FIX messages are constructed with a number of fields, each given by a tag-value pair and separated by a delimiter. Figure 4 shows an example of a FIX message.

The simulator may run a market consisting exclusively of simulated trading agents running algorithmic strategies (e.g., to study algorithms or market effects), or participants (typically human) may log in and interact with the simulated market (e.g., for training). To allow multiple users to interact with our simulator simultaneously and independently, we use the FIX protocol. The simulator provides each user a designated port for login and maintains a dedicated channel for communication. Upon receiving a FIX message, the simulator retrieves the

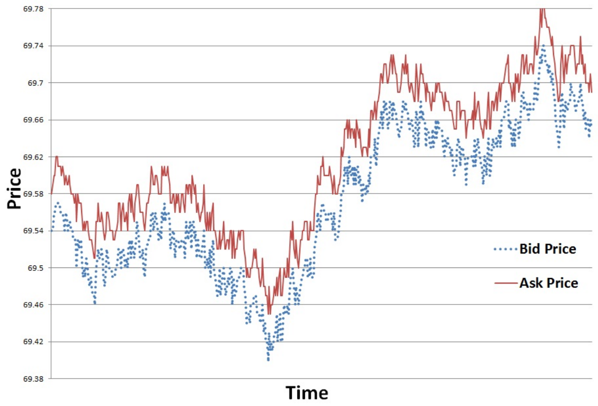


Figure 3: A simulated price movement in which spread was controlled by a Market Maker Agent. The pre-defined spread limit was set to 0.05.

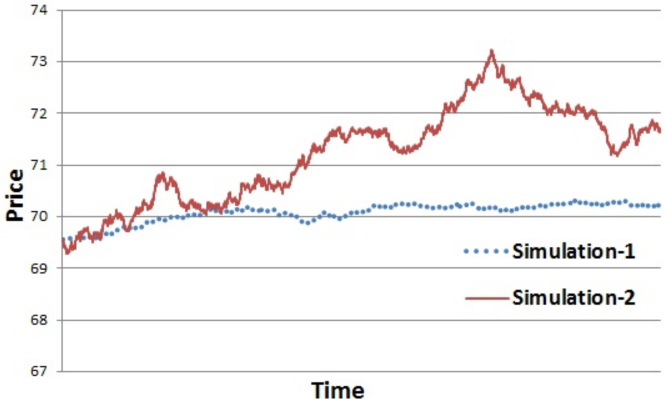


Figure 5: A comparison of two simulations with different liquidity taking.

```
8=FIX.4.2|9=201|35=D|49=Broker_16|
56=MatchingEngine|52=20140301-20:42:37.426|
34=357256|1=Sim_Account|11=8321660696624948305|
21=1|55=TD.CA|54=1|38=31600|40=2|44=93.77|
60=20140301-20:42:37.426|59=0|100=*|167=CS|10=128|
```

Figure 4: An example of FIX NewOrderSingle message. The original delimiters have been replaced by “|” for clarity. The message represents a buy limit order attempting to purchase 31,600 shares of the security *TD Canada Trust* at or below $93.77.

information by parsing the tag-value pairs and then processes the message accordingly. All returning messages, including execution reports and quotes, are also encoded as FIX format before sending back to each user. Using the FIX protocol also provides easy access to our simulator. Most users in both industrial and academic algorithmic trading settings are familiar with the FIX protocol or have it already implemented in order to connect to financial markets. This makes it possible to interact with our simulator with little modification to their systems.

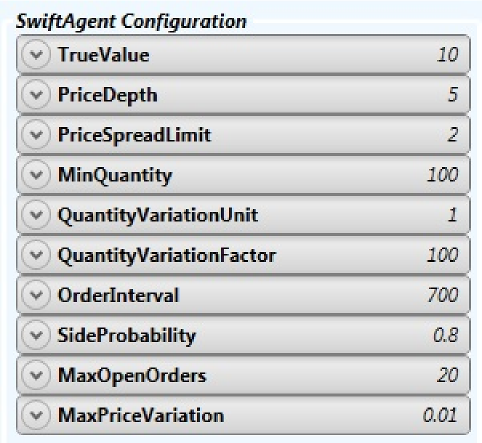
C. Simulated Trading Agents

A financial market is considered as a dynamic system constituting a heterogeneous group of investors, each following their own trading strategy in an attempt to gain superior return. A considerable amount of effort in the past years has gone into the pursuit of proper modelling of the financial market system. One well-known method, which we adopt, is to use agent-based modelling [8] where each computerized agent represents a human or robotic trader, and agents compete with each other. This determines the price of securities and consequently forms a market. Since price fluctuations can depend on the interactions of all the agents and on additional conditions that did not take place in historical data, trading strategies can be arguably more fully evaluated by this approach than in the back-testing model [9]. In order to create various market conditions that are suitable for testing, we have developed five pre-defined types of agents that represent an important subset of trading entities we observe in the real market. These agents are able to adapt to the market and interact with users’ algorithmic trading strategies. We outline each of these types of agent below. Other agents may be defined by programmed extensions, but the pre-defined agents are often enough to model desired scenarios.

Market Maker Agent The Market Maker Agent plays neutrally against the market and, as its name suggests, is an imitation of the market makers that commonly exist in many exchanges, such as TSX [10] and NASDAQ [11]. A market maker’s primary objective is to enhance the liquidity and the depth of the market, typically for a specific security. This provides an efficient way to get into and out of the market for small investments in the given security. Market makers also contribute to the stability of the market. When the security price is moving significantly up or down, the agent will post a reasonable volume of offers in an attempt to counter the trend. The Market Maker Agent is sensitive to price changes on both sides and keeps monitoring the difference between the bid and ask (the spread). If the difference exceeds a pre-defined threshold, it will adjust its orders accordingly to reduce the spread. Figure 3 shows a simulated price movement in which spread was monitored by a Maker Maker Agent.

Liquidity Taker Agent The Liquidity Taker Agent takes liquidity from the market by posting market orders that are often immediately executed at the best available price. This is a special type of agent as it can be used to introduce volatility to the market. By increasing the size or the frequency of the orders, it can potentially fill more orders from the opposite side, which often causes the quoted prices to change dramatically. Figure 5 shows a comparison of volatility between two simulations. The settings were the same except in the second simulation the Liquidity Taker Agent issues market orders at a higher frequency.

Liquidity Provider Agent In contrast, the Liquidity Provider Agent provides liquidity to the market by placing limit orders to the market. A buy limit order can only be executed at the specified price or lower and a sell limit order can only be executed at the specified price or higher. By posting limit orders on both sides without immediately triggering a trade, it adds liquidity to the order book and consequently increases the depth and the stability of the market.



SwiftAgent Configuration

TrueValue	10
PriceDepth	5
PriceSpreadLimit	2
MinQuantity	100
QuantityVariationUnit	1
QuantityVariationFactor	100
OrderInterval	700
SideProbability	0.8
MaxOpenOrders	20
MaxPriceVariation	0.01

Figure 6: The user interface of Swift Agent configuration.

Random Agent The Random Agent uses no information about the market and issues random orders at certain time intervals. This type of agent can be used to create chaos in the simulation environment as well as to investigate the cause of certain market phenomena. At each time a Random Agent decides whether to apply an action to the market, and if so, to which side. The order quantity varies within a specified range and its value is computed using a pseudo-random number generator. The order price follows a standard normal distribution with the mean at the current trade price and with a configurable standard deviation.

Swift Agent We have also developed a special type of agent that we call the Swift Agent. Compared to the other four agents, a Swift Agent is more sophisticated in that it is able to control the number of open orders it places. This prevents the agent from exposing itself to too much risk, just as human traders would also do in a real market. Meanwhile, by reducing the number of open orders in the matching engine, this type of agent lightens the burden on the simulation server and consequently improves overall performance. In addition, the agent is able to monitor the price fluctuations of the simulated market. If the price variation exceeds a certain threshold, the agent will attempt to place more orders on the opposite side to counter the trend.

III. IMPLEMENTATION

We have built two types of simulators based on the framework presented in Section II, and we explain their implementations below.

A. *Agent-Based Simulation*

A variety of existing agent-based simulators work in a synchronous way where all agents are placed in a waiting queue. At any given time only one agent is active and the rest are all in a “sleep” state. The active agent performs some actions against a matching engine, notifies the next agent to “wake up” and then transitions back to a “sleep” state. Thus, the next agent is not allowed to do anything until the active agent finishes its job. Clearly this typical mechanism differs significantly from real-world environments, which are continuous and asynchronous. In order to address this issue, our agent-based simulator, in contrast, supports asynchronous operation of agents so that a number of agents can perform a variety of actions at the same time. In addition, our simulator guarantees that all requests submitted by agents are processed at the precise time of their arrival. This avoids accumulating inaccuracies as the simulation evolves. Our simulator also maintains the correct ordering of the requests. This allows simulations to respect causality, and each agent can re-schedule its future actions.

Each agent provides a succinct graphical user interface for quick configuration and to adjust the parameters that define every aspect. Figure 6 shows a snapshot of the graphical user interface of the Swift Agent. By adjusting the configuration of each agent or the proportion of each agent type, one can develop a flexible, realistic, and efficient simulated market with large number of different agents. Figure 7 shows a snapshot of launching multiple agents in our simulation environment.

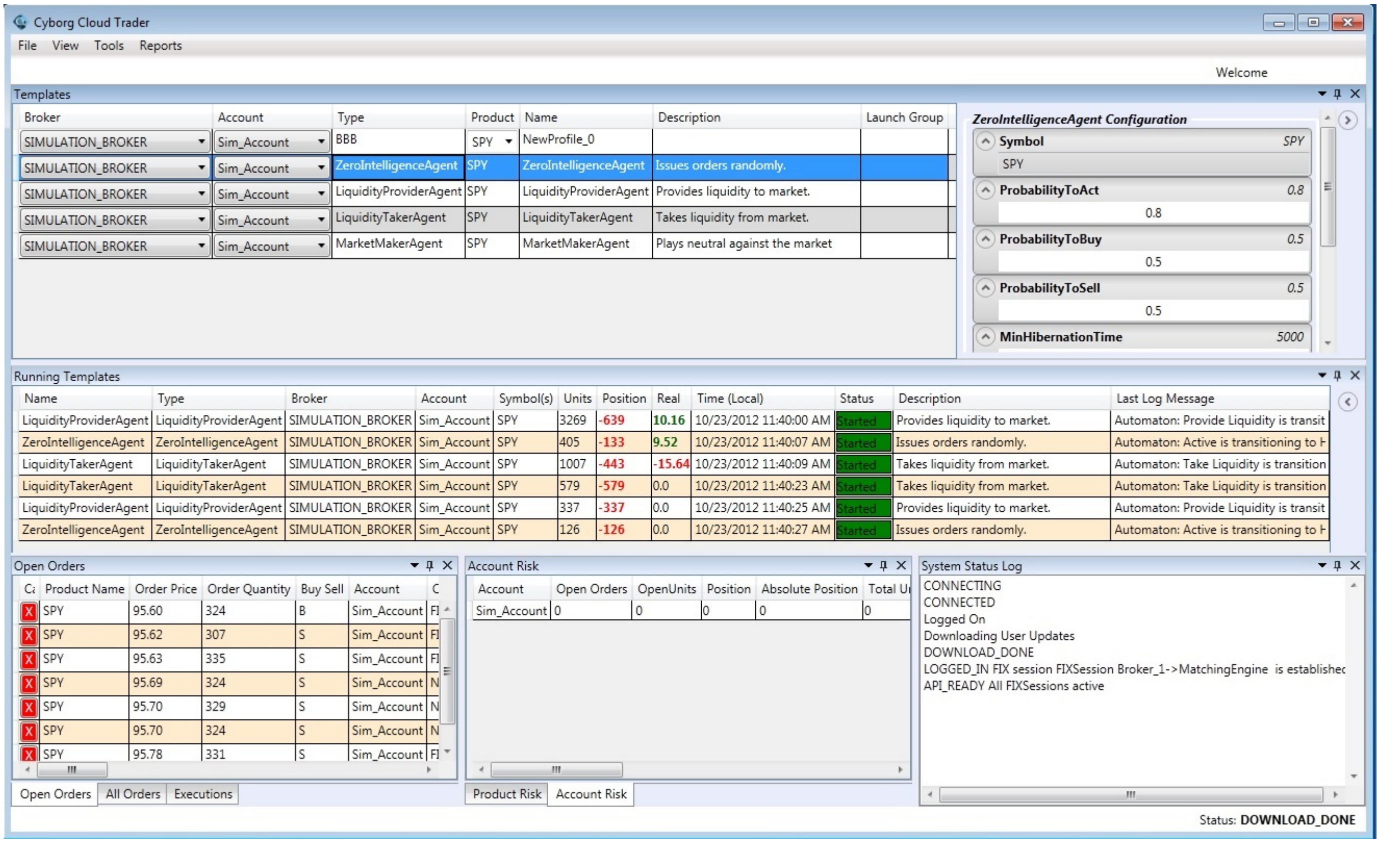
B. *Live-Data Simulation*

We now describe another type of simulator which uses live data from real exchanges. Orders received from users are matched against the current quoted prices in the market. This provides users a risk-free environment to test their trading strategies in a realistic setting (although, of course, orders in the simulation do not affect the live market). As mentioned earlier, prior real market data simulation technology already exists. However, our simulator outperforms others in several different aspects. First, it supports a variety of security types, including equities, futures, foreign exchange and options, while others typically allow only a single security type. Second, compared to existing simulators such as the Penn Exchange Simulator (see Section V), which pulls only snapshots of the current markets at certain time intervals, our simulator uses tick-by-tick data. This has a much finer granularity and therefore can reflect the current market conditions more precisely. Last, but not least, unlike existing simulators that typically require programming skills in specific languages, our simulator adopts the FIX protocol, which will ease the process of integrating the simulator into other systems.

The throughput of the matching system is extremely important. One of the biggest challenges we have encountered in the development of the simulator was the delivery of the massive volume of real-time market data to the trading agents and logged in participants. Following 100 products, we could easily hit the limit of our server, which is about 1500 quotes per second. As the simulator needs to publish the quotes to each of its subscribers, the number of quotes that actually pass through the system could be significantly larger. To improve performance, we use immutable objects in the matching engine, allowing multiple threads to use the data simultaneously without exclusive locking. We found this eliminated significant overhead. In our experiments, we found that the server was able to subscribe up to 350 products and latency was not observed after a long run.

C. *The Software*

The components of the simulator were written primarily in the C# programming language. We also adopted F# to implement the message objects that passed through the system as it has native support for immutability. In total, there were 137 classes with about 40,000 lines of code, and it took approximately 30 person-months to implement. The FIX protocol that we adopted was version 4.2, which is currently prevalent in industry. The simulation servers were deployed on Microsoft Windows 2012 servers.



Templates

Broker	Account	Type	Product	Name	Description	Launch Group
SIMULATION_BROKER	Sim_Account	BBB	SPY	NewProfile_0		
SIMULATION_BROKER	Sim_Account	ZeroIntelligenceAgent	SPY	ZeroIntelligenceAgent	Issues orders randomly.	
SIMULATION_BROKER	Sim_Account	LiquidityProviderAgent	SPY	LiquidityProviderAgent	Provides liquidity to market.	
SIMULATION_BROKER	Sim_Account	LiquidityTakerAgent	SPY	LiquidityTakerAgent	Takes liquidity from market.	
SIMULATION_BROKER	Sim_Account	MarketMakerAgent	SPY	MarketMakerAgent	Plays neutral against the market	

ZeroIntelligenceAgent Configuration

Symbol	*SPY*
SPY	
ProbabilityToAct	*0.8*
0.8	
ProbabilityToBuy	*0.5*
0.5	
ProbabilityToSell	*0.5*
0.5	
MinHibernationTime	*5000*

Running Templates

Name	Type	Broker	Account	Symbol(s)	Units	Position	Real	Time (Local)	Status	Description	Last Log Message
LiquidityProviderAgent	LiquidityProviderAgent	SIMULATION_BROKER	Sim_Account	SPY	3269	-639	10.16	10/23/2012 11:40:00 AM	Started	Provides liquidity to market.	Automaton: Provide Liquidity is transit
ZeroIntelligenceAgent	ZeroIntelligenceAgent	SIMULATION_BROKER	Sim_Account	SPY	405	-133	9.52	10/23/2012 11:40:07 AM	Started	Issues orders randomly.	Automaton: Active is transitioning to H
LiquidityTakerAgent	LiquidityTakerAgent	SIMULATION_BROKER	Sim_Account	SPY	1007	-443	-15.64	10/23/2012 11:40:09 AM	Started	Takes liquidity from market.	Automaton: Take Liquidity is transition
LiquidityTakerAgent	LiquidityTakerAgent	SIMULATION_BROKER	Sim_Account	SPY	579	-579	0.0	10/23/2012 11:40:23 AM	Started	Takes liquidity from market.	Automaton: Take Liquidity is transition
LiquidityProviderAgent	LiquidityProviderAgent	SIMULATION_BROKER	Sim_Account	SPY	337	-337	0.0	10/23/2012 11:40:25 AM	Started	Provides liquidity to market.	Automaton: Provide Liquidity is transit
ZeroIntelligenceAgent	ZeroIntelligenceAgent	SIMULATION_BROKER	Sim_Account	SPY	126	-126	0.0	10/23/2012 11:40:27 AM	Started	Issues orders randomly.	Automaton: Active is transitioning to H

Open Orders

Ca	Product Name	Order Price	Order Quantity	Buy Sell	Account	C
X	SPY	95.60	324	B	Sim_Account	FI
X	SPY	95.62	307	S	Sim_Account	FI
X	SPY	95.63	335	S	Sim_Account	FI
X	SPY	95.69	324	S	Sim_Account	N
X	SPY	95.70	329	S	Sim_Account	N
X	SPY	95.70	324	S	Sim_Account	N
X	SPY	95.78	331	S	Sim_Account	FI

Account Risk

Account	Open Orders	OpenUnits	Position	Absolute Position	Total Ui
Sim_Account	0	0	0	0	0

System Status Log

```
CONNECTING
CONNECTED
Logged On
Downloading User Updates
DOWNLOAD_DONE
LOGGED_IN FIX session FIXSession Broker_1->MatchingEngine  is establishec
API_READY All FIXSessions active
```

Status: **DOWNLOAD_DONE**

Figure 7: A snapshot of launching multiple agents.

IV. USEFUL SCENARIOS

Both the agent-based and the live-data simulator have been adopted by Quantica Trading, a company located in Kitchener, Canada, which develops algorithmic trading software. They have been deployed on production servers to provide immediate access to users both inside and external to the company. Both have been extensively used by researchers and engineers on a daily basis.

We have found, informally, the agent-based and live-data simulators to be useful in the following scenarios.

A. *Trading System and Strategy Testing*

Testing of algorithmic trading strategies can be challenging as there can be a variety of conditions that must be taken into account. Some conditions, such as a market crash, may not occur very often in a real market, but they are extraordinarily costly when they do occur and so must be examined. With the agent-based simulator, we can easily reproduce these conditions, and variants, to test strategies. Figure 8 shows a bubble created using our simulator.

We have also provided a list of special order books in our simulator, each of which applies a deterministic rule when executing orders, such as full fill, partial fill, slow fill and etc. We found they were particularly useful for testing trading systems which typically require deterministic input and output. By issuing orders on a special testing security, we can anticipate deterministic responses, which are very suitable for black-box testing.

B. *Education and Training*

Our simulator has also been found useful for education and training purposes. By executing trades in the risk-free simulation environment, it facilitates trading drills designed for new traders, allowing them to learn how to execute trades and manage risk faster.

C. *Software Demonstration*

Our simulator is also suitable for software demonstration. With round-the-clock access and risk-free testing, users can present demonstrations of their software applications at their convenience.

D. *Evaluating Regulatory Effects*

A simulator of the type we present also benefits users beyond the high frequency trading world. High quality simulation is essential to improve the regulatory environment for North American markets. At the moment, the true impact of regulation cannot be completely understood until it is in effect in the markets. This means that regulation can have unintended consequences or not achieve its desired results. High-quality simulation can help improve this, reducing risk of events such as the “Flash Crash” of 2010 [5].

V. RELATED WORK

Building a simulation environment that is suitable for evaluation of algorithmic trading strategies is a problem for which there has been considerable previous work. We highlight some notable relevant contributions here.

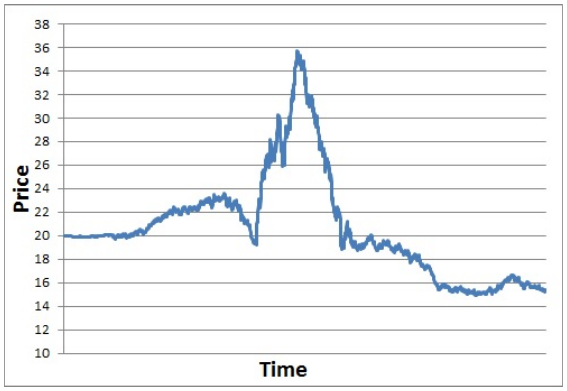


Figure 8: A bubble created by our simulator.

One of the earliest attempts was perhaps the Santa Fe Artificial Stock Market, which was initiated by Brian Arthur and John Holland. Their objective was to build a financial market with an ecological system where successful strategies would persist and replicate, and weak strategies would go away. The first version of the Santa Fe Artificial Stock Market became available in early 1990s and was published in [12]. It adopted agent-based modelling techniques but was operated naïvely. A price was announced by a market maker to all agents, then each agent submitted an order to buy or sell one share of stock. Most times, this market was out of equilibrium with either more buyers or sellers. The smaller of these two sets would get satisfied while the other would get rationed [13]. An advanced version was later developed, adding extensive modifications to the first version. It was able to generate several features similar to actual financial data. However, it did not support continuous execution mechanisms. Time was broken into discrete time periods $t = 1, 2, ...n$, in each of which agents were asked in a row to place an order in the market. This was not realistic compared to a real market where each participant can trade asynchronously. Meanwhile, human users were not allowed to interact with simulated environment.

The Penn Exchange Simulator was another software simulator for algorithmic stock trading. It received real-time order book data from Island Electronic Crossing Networks, which was provided in the form of snapshots of the top 15 limit orders (price and volume) [14]. The data was polled approximately every 3 seconds. Compared to other simulation platforms that modelled order book based on price information alone (i.e., quotes), the Penn Exchange Simulator provided a more realistic environment for simulation. For this reason, it had been chosen to be the testing platform of several Trading Agent Competitions [15]–[17]. The Penn Exchange Simulator accepted connection requests from users. In a multi-user simulation environment, the orders were matched both with other users' orders and with the orders from the real market. This allowed blending the internal and external markets. To evaluate the performance of each user, the Penn Exchange Simulator automatically computed various quantities of interest, including profit and loss. The Penn Exchange Simulator provided designated ports, to which a number of clients could connect. Each client was allowed to perform a variety of actions such as buy or sell orders, examine his or her profile, and monitor market data. On the other hand, there were several drawbacks. First, it supported limit orders only and did not allow securities other than stocks. This restricted its use to test certain trading strategies. Second, users with enough resources could potentially take advantage of the data latency by gaining access to a faster real-time data source. Last, but not least, once trading strategies were fixed, users were not allowed to intervene during the day.

The Investopedia Simulator [18] was a web-based simulator whose primary purpose was to introduce people to the stock market. It used market data from real exchanges in order to imitate the experience of dealing with a real online brokerage account. Each registered user was allocated with certain amount of virtual cash and could issue trades based on real market data. Similar to the Penn Exchange Simulator, the Investopedia Simulator was only available during market hours. In addition, as only manual orders were allowed, it was not possible to test algorithmic trading strategies.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

Recent market events, such as the "Flash Crash" and the Knight Capital trading disruption [19], have once again stressed the need to have a platform that is suitable to develop and evaluate algorithmic trading strategies. We have presented a financial market simulator that supports a full range of security types and allows users to interact as if they were trading in a real market. In addition, it adopts FIX as the communication protocol, allowing multiple users to interact with the simulation environment simultaneously and independently. We have also presented several types of simulated trading agents which represent a subset of traders observed in real markets. All of these agents are configurable and, by adjusting their parameters, very specific market conditions can be created to explore certain market behaviours. We have found that in a corporate setting that our simulator is useful in a number of scenarios, including system testing, education, training, and policy evaluation.

There are a few interesting directions that we would like to pursue in the future. First, all the securities offered by the current simulator are independent from each other, but in a real market some are correlated. That is, if the price of one security changes, the other security will move in either the same or the opposite direction. In order to create a more realistic simulation environment, we may compute how securities move in relation to each other from historical market data and use such correlations to guide the agents when they interact with the simulator. This would allow prediction of the price movement of multiple securities and consequently better evaluate overall market conditions. Second, we may wish to compare the simulated market with the actual one. This would allow us to assess not only the fidelity of the trading strategies employed by our agent models but also the possible impact if they were deployed in a real market. Last, but not least, it would be interesting to develop advising agents to provide support to human traders when trading against real markets, helping them reduce risk and optimize profit.

ACKNOWLEDGMENTS

This work was supported, in part, by a CU-I2I grant from the Natural Sciences and Engineering Research Council of Canada. We thank James McInnes, Jonathan Leaver and Travis Felker for discussions on desired functionality, and Peter Metford and Vadim Mazalov for discussions relating to implementation.

References

[1] T. Hendershott, C. M. Jones, and A. J. Menkveld, “Does Algorithmic Trading Improve Liquidity?” J. Finance, vol. 66, no. 1, Feb 2011, pp. 1–33.

[2] R. Curran and G. Rogow, “Rise of the (Market) Machines,” http://blogs.wsj.com/marketbeat/2009/06/19/rise-of-the-market-machines/, [retrieved: Sep, 2014].

[3] R. Iati, “The Real Story of Trading Software Espionage,” http://www.wallstreetandtech.com/trading-technology/the-real-story-of-trading-software-espionage/a/d-id/1262125?, [retrieved: Sep, 2014].

[4] Trading Technologies Inc., “TT SIM,” https://www.tradingtechnologies.com/ttsim/, [retrieved: Sep, 2014].

[5] U.S. Commodity Futures Trading Commission and US Securities & Exchange Commission, “Findings Regarding the Market Events of May 6, 2010,” http://www.sec.gov/news/studies/2010/marketevents-report.pdf, [retrieved: Sep, 2014].

[6] FIX Trading Community, “Financial Information eXchange (FIX) Protocol,” http://www.fixtradingcommunity.org/, [retrieved: Sep, 2014].

[7] Quantica Inc., “Simulator Demo,” http://works.bepress.com/stephen_watt/3, [retrieved: Sep, 2014].

[8] B. LeBaron, “Agent-based Computational Finance: Suggested Readings and Early Research,” J. Economics Dynamics and Control, vol. 24, no. 5-7, June 2000, pp. 679–702.

[9] N. Ponomareva and A. Calinescu, “Extending and Evaluating Agent-Based Models of Algorithmic Trading Strategies,” IEEE Int’l Conf. on Engineering of Complex Computer Systems, 2012, pp. 351–360.

[10] Toronto Stock Exchange, http://www.tmx.com/, [retrieved: Sep, 2014].

[11] NASDAQ Exchange, http://www.nasdaq.com/, [retrieved: Sep, 2014].

[12] R. G. Palmer, W. B. Arthur, J. H. Holland, B. LeBaron, and P. Tayler, “Artificial Economic Life: A Simple Model of a Stockmarket,” Physics D, vol. 75, no. 1-3, Aug. 1994, pp. 264–274.

[13] B. LeBaron, “Building the Santa Fe Artificial Stock Market,” Brandeis University, Tech. Rep., 2002.

[14] M. Kearns and L. Ortiz, “The Penn-Lehman Automated Trading Project,” IEEE Intelligent Systems, vol. 18, 2003, pp. 22–31.

[15] M. P. Wellman, A. Greenwald, P. Stone, and P. R. Wurman, “The 2001 Trading Agent Competition,” IEEE Internet Computing, vol. 13, 2000, pp. 935–941.

[16] M. P. Wellman et al., “Designing the Market Game for a Trading Agent Competition,” IEEE Internet Computing, vol. 5, no. 2, 2001, pp. 43–51.

[17] P. Stone et al., “ATTac-2000: An Adaptive Autonomous Bidding Agent,” J. Artificial Intelligence Research, vol. 15, 2001, pp. 189–206.

[18] Investopedia, “Investopedia Simulator,” http://www.investopedia.com/simulator/, [retrieved: Sep, 2014].

[19] Reuters, “Error by Knight Capital Rips Through Stock Market,” http://www.reuters.com/article/2012/08/01/us-usa-nyse-tradinghalts-idUSBRE8701BN20120801, [retrieved: Sep, 2014].

Simulation-based Completeness Analysis and Adaption of Fault Trees

Volker Gollücke
and Axel Hahn
System Analysis and Optimization,
Carl-von-Ossietzky-Universität Oldenburg,
Oldenburg, Germany
e-mail: {golluecke,hahn}@wi-ol.de

Jan Pinkowski, Christoph Läsche,
and Sebastian Gerwinn
OFFIS
Institute for Information Technology
Oldenburg, Germany
e-mail: {pinkowski,laesche,gerwinn}@offis.de

Abstract—**Safety analysis is a common and important task for any operational planning of missions such as, in a maritime context, the construction and maintenance of offshore wind farms. Identifying potential risks that might occur during the planned operation or a sequence of operations is the main task of an associated hazard and risk analysis. This paper introduces an approach to automatically adapt fault trees based on a simulation of a model of the operation in question enhanced by a formal description of hazards. Formalizing the corresponding hazard specifications allows us to generate observers which in turn identify failures and hazards during a simulation of the system. Such detection of critical situation can interact with the simulation via a specialized controller application for the simulation thereby triggering the completion of an existing fault tree. As potentially stochastic models of the environment are considered as well, this simulation approach naturally provides guarantees of the result in terms of statistical confidence statements. The feasibility of this approach is exemplified on an offshore lifting operation.**

Keywords–Simulation; Statistical Model Checking; Fault Trees; Risk Assessment; Offshore Operations; Observer.

I. Introduction

Identifying and mitigating potential hazards is an important yet demanding task in nearly every system design process. The likelihood of missing a potential risk depends critically on the degree of complexity of the operation to be planned. Therefore, having means of checking the completeness of an initial risk picture is highly desirable. Current safety regulations, e.g., within the domain of offshore operations, require a description of all involved risks [1][2]. Specifically, these regulations require an in-depth characterization of risks, including listing causal factors as well as options for mitigation of high-level undesired events [3]. A common way to visualize and construct such characterizations of the involved risks are fault trees [4]. Fault trees split the risk of violating an overall safety goal, e.g., physical inviolability of all personnel, into different failures which in its combination might lead to the undesired event. Although fault trees are easy to interpret and can be used to calculate the probabilistic rate of the undesired event, they are usually created manually and therefore prone to oversight. On the other hand, simulations are widely used to optimize complex operations, as they often can give a more accurate estimate of incurred costs of an operation [5]. Supplementary to the manual process of creating risk descriptions in terms of fault trees, we propose a simulation-based risk analysis which can be used to complete a given fault tree by iteratively adding failure-combinations which have been detected by the simulation to be able to trigger a violation of the overall safety goal. Importantly, this approach comes with a statistical statement which specifies the level of confidence that no further critical failure combinations exist. In particular, this includes a confidence statement about the probability of the occurrence of each top-level hazard being below a critical value. On the implementation side, we develop an integrated framework of high-level process planning, a platform for handling different co-simulations for different components of operations, and a tool to formalize the manually identified hazards and failure combinations to be able to detect them during simulation. We demonstrate the potential benefits of such a framework on a hazard and risk analysis problem for a specific offshore crane operation of moving a cargo on a vessel observed by a lift supervisor.

The paper is organized as follows. In Section II, we explain each component of the developed simulation framework including the used models, the generation of observers as well as the configuration of the simulation to be suitable for a statistical evaluation of the obtained simulation runs. To illustrate the framework, we present the evaluation of the framework on the considered use case in Section III before concluding in Section IV.

II. Methodology

Using simulation to identify additional failure combinations that lead to a hazard requires an accurate description of the operation to be performed and analyzed. This also means that potential risks have to be encoded into such model used for simulation – at least implicitly. Planning operations on a detailed level consists of planning many sub-tasks. As a result, undesired high level situations, such as personnel injury, might depend on complex circumstances which are not easily identified. We illustrate the proposed process for modeling, definition of risks, and integration with the simulation in Figure 1 which we will first describe on a high level before explaining the individual steps in more detail.

As a first step, the model for the operation to be performed has to be synthesized, as indicated in Figure 1 by *"System Model"*. Each of the defined sub-tasks can be annotated with potential failure combinations leading to hazards from which a *"Fault Tree Model"* can be synthesized. In parallel, the planned operation (and sub-tasks) can be refined iteratively, resulting in a *"Scenario Description Model"* which can be simulated. By using a suitable formalization of individual failures within the fault tree, observers can be generated. They are able to observe a simulation, indicating if an individual failure, hazard, or a combination thereof has occurred in a single simulation run. To this end, the *"Observer"* has to be triggered by a central *"Simulation and Control"* part. If an observer indicates

ISBN: 978-1-61208-371-1

failures that are currently not considered within the fault tree, a combination of them can be used to adapt the current fault tree, resulting in a more complete risk characterization. Otherwise, we can generate more simulation runs, until sufficiently many traces of the operation to be planned have been simulated without observing any non-considered failure combination.

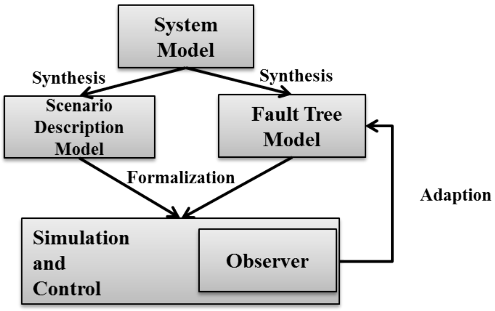


Figure 1. Schematic overview of the simulation-based analysis and adaption approach.

A. System Model

As a basis for our approach, we require a model of the system to be analyzed. We use a process model (cf. Droste et al. [6]) based on concepts from business process modeling languages such as BPMN [7] or the activity diagram of UML [8]. The respective process represent actors, tasks, and corresponding interactions. This model type has been developed for application in the maritime domain, more specifically for use for offshore operations. An exemplary model is described in Section III. The graphical elements of the model depict the task sequences and interactions and it is also possible to annotate corresponding hazards and failures to the model. A hazard describes a potential source of risks on a system-wide level, i.e., the injury or death of a person, the pollution of the environment, or the damage of systems. A failure describes a possible cause for a hazard or another, higher-level failure. In the proposed methodology, we use a formal specification for hazards and failures as described by Läsche et al. [9]. Additionally to the process model, the system model consists of an environment model which describes the considered resources like actors, physical objects, or environmental conditions. These resources are also accessible by the process model to map actors to specific avatars of the simulated environment. For example, a crane operator is mapped to a crane operator resource from the simulated environment with an orientation, a location, a mass, and further necessary attributes.

B. Scenario Description Model

In our case, a scenario description can be used to describe the course of a simulation. In our current approach it consists of

- Parameter bounds,
- Parameter exploration techniques,
- Exit conditions, and
- Different trigger types (i.e., simulation-time and simulation-step) to cause parameter changes or to define the end of a simulation (run).

Using the presented system model which describes the progress of an operation and the considered resources – like the ship, a crane, and a lift supervisor – and their properties, a scenario description template with default values can be automatically generated. The generation process reads the resources and their properties from the environment model as well as the modeled events and annotated failures and hazards from the process model. Based on this information the scenario description model is build. The build process categorizes the resources from the environment model and automatically sets a default trigger for each property. The default trigger is used to set the value of a property to the value defined in the environment model at the start of a simulation run. The scenario description is completed by event triggers which are extracted from the process model and are available to the modeler of a scenario.

After this automatic generation of the scenario description it can be adjusted. The adjustment allows to configure the parameter bounds of the resource properties like a minimum and maximum wave height, the parameter exploration techniques like a linear or random exploration and the configuration of the parameter change trigger and exit conditions which are used to define the end of a simulation or simulation run. The triggers can be based e.g., on the simulation time but also on the occurrence of situations such as the start of a task. These situations are defined in the process model. The completed scenario description can then be forwarded to the simulation control to configure the simulation runs (cf. Section II-E2).

C. Fault Tree Model

As part of risk assessment for offshore applications, Fault Tree Analysis (FTA) gains more and more attention in the maritime domain (cf. Vinnem [10], Lavasani et al. [11], and Sutton [12]). Thus, FTA has been recommended as methodology in several guidelines for risk assessment. However, fault trees are typically constructed manually. It requires experienced engineers to structure the system knowledge and necessary details. Manual construction of fault trees thus is time-consuming, cost-intensive, incomplete, and error-prone (cf. McKelvin et al. [13] and Tajarrod et al. [14]). Therefore, several approaches to automatically synthesize fault trees are published (cf. Chen [15] and Pai et al. [16]). These approaches, focusing on the construction of fault trees, typically use formal system models. If both, the underlying model and the individual fault tree elements, such as failures and hazards, are formalized, a formal completeness analysis can be performed via model checking (cf. Schellhorn et al. [17] and Ortmeier et al. [18] for more details on such a formal completeness analysis). By verifying these completeness conditions over the formalized fault tree via model checking, it is possible to guarantee that no additional failure as a cause for the specified system hazard has been missed within a given fault tree. However, this kind of completeness analysis typically does not scale well with the system's complexity. Furthermore, some parts of the system might only be given as a black box model for which no formal description is available but which can be simulated to generate traces describing its behavior. Thus, we focus on a simulation-based methodology to check fault trees completeness and provide a confidence statement about the statistical certainty of the existence of unconsidered failures. Thereby, the fault trees are automatically synthesized from the process model part of the system model. Process model elements such as tasks or actors have annotated hazards and failures. For each annotated hazard, we create a fault tree to logically structure the respective failures that could possibly cause the hazard. Since each hazard relates to an element of the process model, i.e., a task, the respective fault tree is also related to this task. Conversely, each top event of a fault tree corresponds to one hazard. For example, a task can have several fault trees that graphically represent its hazards and corresponding failures. In contrast to other fault tree construction approaches mentioned above all necessary information are included in the process model. The process model elements, such as a task or an actor have specific hazards and failures that are modeled by an experienced safety expert.

D. Observer

To be able to identify the occurrence of the hazards and failures depicted in the fault tree during run-time, we need a formalized description of them. Their identification is important as we want to check the fault trees for errors and completeness. Specifically, we can use the identified events to evaluate each simulation step regarding whether a top event has occurred. If this is the case, it is checked against the fault tree if there is a mismatch between failures identified by the observer and the failures in the fault tree. If a failure in the fault tree is not detected, the failure might not be required for the event to happen. Thus, the fault tree structure might be faulty and has to be reassessed. On the other hand it is checked if failures occur that are not covered in the current fault tree structure identified for the top event. If this is the case, we have to add the detected failure to the fault tree of the top event in order to complete the fault tree structure. As this process is repeated, a more and more accurate fault tree is created and failures required for the hazard to occur are highlighted whereas optional failures are identified and can be corrected. This allows us to have a most accurate estimation (to a certain confidence level) of the involved risks (to the limits of the simulation).

In order to identify hazards and failures during the simulation, we attach observers, i.e., programs to detect the occurrence of events, to the simulation environment. To be able to create such observers, a formal description of the events has to exist. This is achieved through the Hazard Description Language (HDL) which has been developed by us for this exact purpose. The language allows the analyzing person to formalize hazard and failures in near-natural language, allowing persons without knowledge about formalization languages to describe the events. It is used to describe the events that display a hazard or failure to be able to detect them. There is a set of potential formalization patterns, allowing the person to describe several aspects of an event to make it unambiguous. However, we do not directly use the HDL formalization and thus transform it into the Object Constraint Language (OCL) [19]. Thus, the HDL can be seen as an interface for describing hazards and failures using OCL without deeper knowledge of the constraint language by defining a set of HDL patterns that are automatically transferable to OCL expressions. OCL is a widely used language; thus, its usage minimizes our effort as we can rely on a well-evaluated language and multiple implementations.

Every OCL expression is evaluated by a separate observer client that is constantly updated with data from the simulation by the controller (cf. Section II-E1). At each simulation step, all observers are triggered by the controller to automatically evaluate if an event has occurred. If this is the case, the event then is logged for further analysis. The simulation result is analyzed and incorporated as described above. In contrast to the automatic observation of the simulation, this is a manual process.

E. Simulation-based Analysis

Our goal is to determine the completeness of a fault tree with possible failure combinations leading to an overall hazardous event. To this end, we need to ensure that the cutsets of the fault tree is minimal or that there are no other failure combinations which are able to trigger an overall hazardous event. As we are following a simulation-based approach, such a guarantee can only be of statistical nature. Therefore, before detailing the simulation setup, we present the theoretical foundation of determining the necessary number of simulation runs needed for gathering sufficient evidence of the completeness of the current risk description of the corresponding fault tree.

1) Statistical Evaluation of Simulations: Based on multiple runs of the simulation, it is of critical importance to quantify the confidence in the found failure and hazard combinations. In particular, when no more failure combinations leading to a hazardous situation are found using simulation runs, we are aiming at bounding the probability that there are indeed no more critical failure combinations possible. As we allow the underlying model to exhibit stochastic behavior, we cannot be absolutely certain that there are no more failure combinations based on finitely many simulations. However, we can bound the probability that a hazard is caused by a previously not considered failure or combination. To this end, we are using a well-known confidence statement. More precisely, if we assume that we will observe a failure combination with probability p, the probability of not observing a critical situation within N samples is given by

$$P(\text{no failures observed}) = (1-p)^N \tag{1}$$

This equation can be used to construct a confidence statement of the following form. Once we have observed N without any observable critical situation, the following holds true (see Annex D in part 7 of [20]):

Up to a confidence $1-\alpha$, the probability of the occurrence of a critical situation can be bound by $1 - \sqrt[N]{\alpha}$.

Vice versa, we can also bound the necessary number of simulations needed to be sure – up to a confidence level of $1-\alpha$ – that there is no critical situation having an occurrence probability larger than p^+ by :

$$N \geq -\frac{\log(\alpha)}{p^+} \tag{2}$$

By using this confidence statement, we are compliant with the IEC61508 [20]. Note that p^+ is used here to bound the compound probability of the failure combination. That is, using N simulations we can statistically show, that there are no failure combinations which have a probability of occurring together larger than p^+. Taken together we have the following procedure to perform a statistical completeness check:

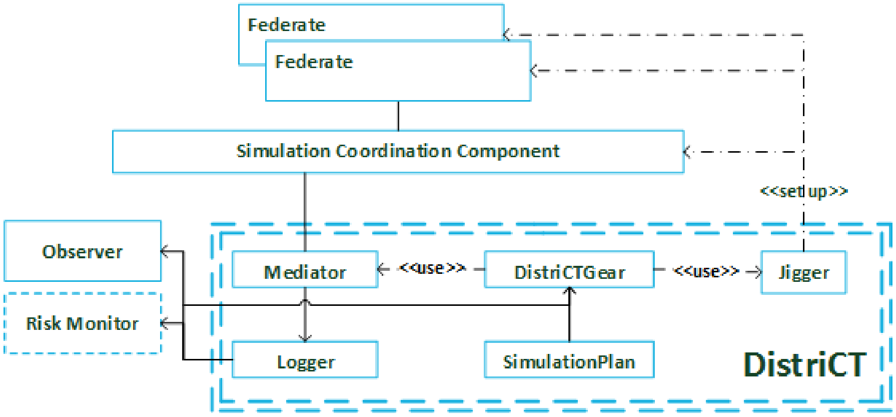


Figure 2. The structure of our distributed simulation setup. It consists of a simulation coordination component, federates (e.g., simulators), observer, and DistriCT to control and monitor the mentioned components.

1) Fix desired confidence level α and critical threshold for the probability p, for example $\delta = 0.05$ and $p = 10^{-8}$
2) Calculate number of simulation runs N
3) Generate N number of simulations
4) If all simulations do not reveal a previously not considered failure combination, we have reached the desired level of certainty of no additional failure combinations
5) If additional risks have been identified, return to 1.

Note, however, that the necessary number of simulations does not scale well with low thresholds for critical probabilities p (cf. equation (2)). In particular, the necessary number of simulation runs scales linearly with the critical threshold. For example, if we chose the critical probability threshold to be $p^{+} < 10^{-9}$, roughly 10^{9} simulation runs are needed. To be able to obtain feasible confidence statement in these kind of situations, specialized, guided simulation methods such as importance splitting or sampling [21][22][23] are needed. However, for the sake of simplicity, we use the procedure presented above. Having a formalization of the hazardous situations could also help to define a score function thereby defining a notion of proximity to a critical situation. This score function, for example could then in turn be used to guide the simulation toward the rare event (see [24]). These kind of simulations also require that the individual simulations can be controlled on an individual time step basis, which we will explain in the next section.

2) Simulation and Simulation Control: To be able to find overlooked failures based on simulation, we use a simulation setup and simulation control tool. Therefore the scenario description can be used together with a description of the required simulators to generate a simulation plan. The simulation plan itself describes a configured sequence of simulation runs and also accesses the simulation environment depicted in Figure 2. The proposed structure, also used in our use case (see Section III), consists of four distinguishable component types which are explained in this section, namely:

1) Simulation Coordination Component (SCC)
2) Federates
3) Observer
4) Distributed Controlling Toolkit (DistriCT)

The central component of the simulation is the *Simulation Coordination Component* (SCC) which manages the communication between the simulators and the time synchronization among them. All simulators or services, called *Federates*, register to the SCC. They specify which objects they want to be able to update and for which they want to receive updates. To support this, a common definition of all object classes and interactions exists called *Object Model Template* (OMT) [25].

Federates can be described as simulators or services connected to the SCC which have the possibility to publish data to and subscribe data from other Federates. In our considered use case two simulators exist. One of them is the *Lift Supervisor Simulator*, responsible for moving the lift supervisor on the ship deck. It uses observation paths which guarantee a free view to the cargo and the crane operator. The simulated lift supervisor chooses which path he takes on given probabilistic value. The other one is the *Physical World Simulator* (PWS), which in our example controls the environment or the crane movement. The PWS is used to provide a 3D model of the scenario and of the physical and environmental conditions. Physical effects are, for example, the collision of objects or soft body effects which are used to simulate the swinging of the crane rope. Environmental conditions are, in our case, particle effects like rain or snow. This simulator is based on the *GameKit* [26] game engine which contains *Ogre* [27] as visualization and *Bullet* [28] as physics engine. The PWS has an integrated visualization which is handy for testing purposes, manual observations, or to demonstrate the simulation. But the PWS can also be executed without graphical output to achieve a simulation speed up. More detailed descriptions of the PWS and its components can be found in Schweigert et al. [29] or Läsche et al. [25].

The *Distributed Controlling Toolkit* (DistriCT) consists of the following different program parts. *Jigger* is used to set up the simulation components on a software level. The Jigger component is divided into a client and a server part, the clients can be started on different computer systems and are available for the server. The server deploys the components, e.g., the simulators, to the clients and offers the ability to start, stop, and configure them. The *Mediator* is used to read all communicated changes via the SCC and sends data from DistriCT to the connected simulators. It allows us to listen and react to the

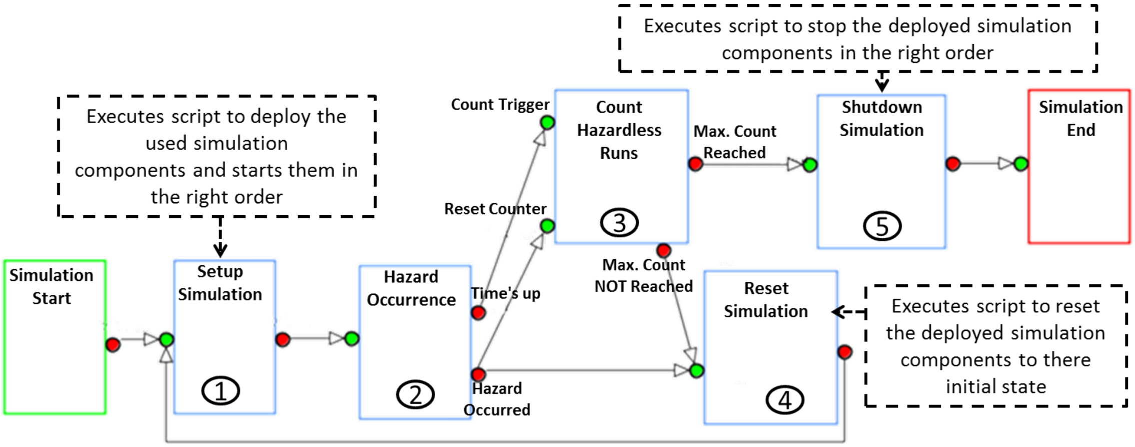

Executes script to stop the deployed simulation components in the right order
Executes script to deploy the used simulation components and starts them in the right order
Count Trigger
Reset Counter
Count Hazardless Runs
Max. Count Reached
Shutdown Simulation
Simulation End
Simulation Start
Setup Simulation
Hazard Occurrence
Time's up
Max. Count NOT Reached
Reset Simulation
Hazard Occurred
Executes script to reset the deployed simulation components to there initial state
1
2
3
4
5

Figure 3. Example of a simulation plan describing the course of simulation runs for a specific simulation scenario.

Observer when hazards or other events occur as well as instruct the other simulation participants, i.e., to reset them or to shut them down. The simulation run configuration can be described by a *Simulation Plan* which is a graph-based representation of the simulation course and can be created from a scenario description. In a simulation plan predefined nodes, like a setup node to initialize the SCC and the connected simulators, can be added. By connecting these nodes, the order of execution as well as sending necessary data from one node to another can be specified in the simulation plan. An example of a simulation plan can be found in Figure 3. In this figure, also representing our later in Section III presented use case, the SCC, the LiftSupervisor-Simulator and the PWS are created at first on a configured system; all necessary data like the mentioned OMTs is transmitted and the components are started in the right order. After that, the Mediator connects and sends the start signal to all Federates (cf. Figure 3:1). While the simulation is performed, DistriCT listens for the occurrence of hazards and the simulation time (cf. Figure 3:2). The *Logger* component writes the information about occurring failures and hazards to a file and also informs a counter (cf. Figure 3:3) if a hazard has occurred or a given maximum simulation time ran out. If a maximum number of non-hazard runs in a row was counted, the simulation is stopped (cf. Figure 3:5). The counter is set back to zero hazard-less runs if it is informed about another occurrence of a hazard. On the other hand, the simulation is reset and started again while the maximum count is not reached (cf. Figure 3:4).

The last component introduced is *DistriCTGear* which is the central element for the other DistriCT components and allows the execution of the simulation plan and its included commands and scripts.

The introduced Observers (cf. Section II-D) are connected to the Logger component from DistriCT. They evaluate the incoming logger data for the occurrence of predefined hazards and inform connected components like DistriCT about it. Note that the Logger can also be connected to monitor-like components (Risk Monitor) which allows us to estimate a distance to a hazardous situation by a semi-automatic created heuristic and use this to trigger the saving of simulation states and running new simulations beginning at the saved state.

III. Use Case – Loading Operation

In order to illustrate our simulation-based risk assessment, we have chosen a maritime loading operation on a jack up vessel – a special ship for offshore operations, cf. Figure 4 b). A crane operator (Figure 4 b:triangle) has to lift a cargo (Figure 4 b:rectangle) with new materials and transport it to a wind turbine platform (Figure 4 b:hexagon). While there are many factors, like a bad sight or communication problems, failures might occur, leading to fatal accidents. Because of these problems during the loading operation, a person is needed to supervise the loading (Figure 4 b:circle) and point out problems to the crane operator. Therefore, the lift supervisor follows the lifted cargo to always keep it, as well as the crane operator, in his field of view (cf. Figure 4 b).

a) System Model: The described system model consists of a process model depicted in Figure 4 a and an environment model. The process model structures the use case scenario and thus tasks of the involved actors in a BPMN-like graphical representation. The use case has been adapted from literature (cf. Droste et al. [6]). It has been reduced by some preparing measures such as load preparation or hook positioning. Thus, the scenario starts with the beginning of the actual lifting operation. The crane operator signals the intended lifting of the cargo to the lift supervisor (Figure 4 a:1). The lift supervisor leaves the safety critical zone around the cargo to be lifted. Meanwhile, the crane operator raises the hook. The lift supervisor signalizes the approval of the operation (Figure 4 a:2). When the crane operator received the OK (Figure 4 a:3), the cargo then is lifted, moved to the wind turbine platform, and positioned at the target. During the operation, the lift supervisor continuously informs the crane operator about the current status from his point of view (Figure 4 a:4), whereas the crane operator steers the crane accordingly. Hence, the lift supervisor requires clear sight on the cargo, the crane operator, and potential obstacles. The scenario ends with successfully placing the cargo on the wind turbine platform. The environment model from our use case describes the following actors and physical objects and their properties. The physical objects consist of the jack up vessel, the crane on the ship and the transported cargo. The actors comprise the crane operator and the lift supervisor. The physical objects as well as the actors have a start pose (position and orientation) and a mass.

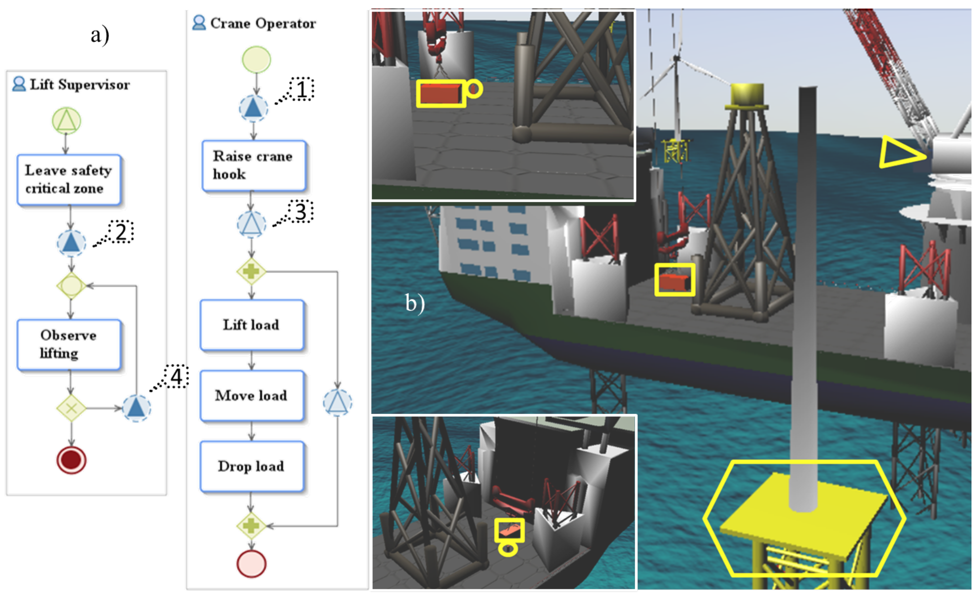


Figure 4. a) Process model of the use case including the involved actors crane operator and lift supervisor and their corresponding tasks. b) Overview of the simulation scenario from different vantage points. The lift supervisor is marked by a circle, the loading by a rectangle, the crane cockpit by a triangle, and the wind turbine platform by a hexagon.

b) Scenario Description Model: Using this system model a scenario description template is created. The jack up vessel, the crane, the transported cargo, as well as the lift supervisor and the crane operator are converted into scenario configuration elements and automatically completed by a start trigger which means that the standard property values from the environment model like the position are set at the start of a simulation run. The scenario description is then adjusted to define the simulation and simulation run ends. Therefore three exit conditions are added for the simulation end and for the simulation run ends. The simulation end is defined by a trigger listening to the current simulation run count and comparing it with a given maximum count of simulation runs. The simulation run end is defined by two triggers. The first one is a time trigger which sets the maximum simulation time to 30 seconds, the other one is an event trigger which stops the simulation run if the observer monitors a collision between the lift supervisor and the lifted cargo. From this scenario description, the simulation plan is generated (cf. Figure 3), then completed by the setup, shutdown, and reset scripts and executed by the DistriCT component.

c) Fault Tree Model: For the introduced operation, we annotated the potential hazard “lift supervisor collides with cargo” and corresponding failures, causing the hazard. These elements are depicted in the synthesized fault tree shown in Figure 6 a. The hazard occurs if a person is located below the cargo while it is dropping. Dropping of cargo can have several causes, for example it falls uncontrolled or it is intentionally dropped by the crane operator. Normally, in cases of imminent collisions the crane operator should initiate an emergency stop. If it is not initiated while the cargo is dropping, this could have several causes, e.g., technical defects which could be used to decompose one failure to more sub-failures. The fault tree depicted in Figure 6 a) represents the possible causes for a collision of the cargo with the lift supervisor. However, in the example of these failures the developer of the fault tree cannot know if there are other potential failures causing the hazard.

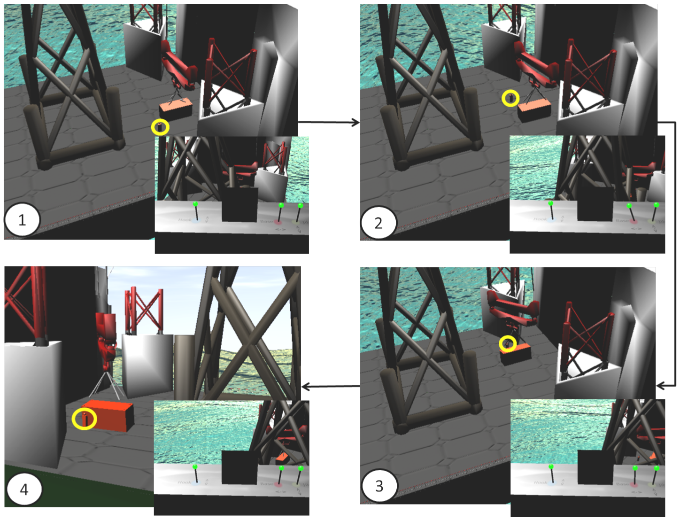


Figure 5. Collision case from load collision simulation run. **Big pictures:** Different views on the loading area short before the accident(1-3) and while the load collision hazard occurs(4). **Small pictures:** View from the crane cockpit taken at the same time as the corresponding big picture.

d) Observer: In beforehand to the first simulation run of the use case, a hazard was identified by the domain expert: A collision between cargo and lift supervisor (see top event in Figure 6 a). To avoid this situation, the domain expert analyzed the situation and identified as failure leading to that hazard that the lift supervisor is positioned below the cargo. Both situations have been formalized by the expert using HDL. The hazard “lift supervisor collides with cargo” has been described as “*distance between LiftSupervisor and Cargo equals*

0cm," i.e., the bounding boxes of the two objects overlaps. The failure "lift supervisor below cargo" has been written as "*position.x of LiftSupervisor equals position.x of Cargo and position.y of LiftSupervisor equals position.y of Cargo and position.z of Cargo is more than height of LiftSupervisor*," i.e., the position of the lift supervisor is a position below the cargo. These expressions are translated to OCL expressions during the generation of the observer. They are evaluated during execution of the simulation to indicate the occurrence of the described events.

All events that occur during each simulation run are logged. This allows the domain expert to check if the previously identified hazard occurred after the previously defined dependent failures have occurred. If this is not the case, a failure causing the hazard might have been missed or the fault tree might otherwise be faulty.

e) Simulation-based analysis: The 3D environment (see Figure 4 and Figure 5) was modeled with Blender [30]. It consists of a jack up vessel, the transported cargo, an uncompleted wind turbine, the lift supervisor and the crane operator. We completed it by a functional crane consisting of a rotatable crane base,a movable jib and a rope which can react to different physical forces (e.g., the wind). The PWS was used to load the 3D environment, handle the physical effects and control the crane. The LiftSupervisor-Simulator was used to move the lift supervisor in the 3d environment on a chosen path. Like explained in Section II-E we used the created scenario description to generate the used simulation plan shown in Figure 3.

If an error in the fault tree has been detected while observing the simulation, it has to be corrected. During the execution of the simulation plan, an observer indicated a collision between lift supervisor and cargo, although the lift supervisor was not standing below the cargo, which had been identified as a required failure.

At the beginning of the operation (cf. Figure 5:1), the lift supervisor stands next to the cargo waiting for the lifting operation to begin. While the lift supervisor reaches for his next observation point, the crane operator begins to lift the load. In the example case, the crane operator does not lift the cargo high enough and begins the rotation of the crane base too early (cf. Figure 5:2). Figure 5:3 shows the fatal development of the situation not marked as critical. The cargo is too low and moves towards the lift supervisor. The last picture, Figure 5:4, shows the actual accident. The lift supervisor is hit by the cargo. The difficulty for the crane operator to oversee the whole situation can be seen in the four smaller pictures at the corners of the big one, showing the bad sight out of the crane cockpit during the operation.

Thus, the analysis results are failures leading to the hazard that have not been considered in the synthesized fault tree, cf. Figure 6 a). The analysis discovered that the hazard occurs even if the lift supervisor is not located below the cargo (cf. Figure 6 b:1). Additionally, the results show that the rotation of the crane while cargo is raised also leads to the hazard (cf. Figure 6 b:2). These results are used to manually adapt the existing fault tree as depicted in (cf. Figure 6 b). Results are therefor used by the user to construct events such as intermediate or basic events in the fault tree as well as fault tree gates in order to logically structure simulation results in the fault tree logic. Thereby, the simulation results are already clustered as failure combinations which already propose a logical structure of found failures based on the simulation runs.

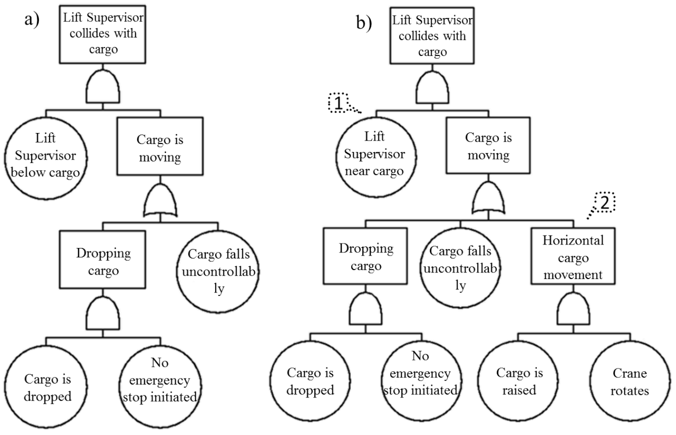

a)
Lift Supervisor collides with cargo
Lift Supervisor below cargo
Cargo is moving
Dropping cargo
Cargo falls uncontrollably
Cargo is dropped
No emergency stop initiated
b)
Lift Supervisor collides with cargo
1
Lift Supervisor near cargo
Cargo is moving
2
Dropping cargo
Cargo falls uncontrollably
Horizontal cargo movement
Cargo is dropped
No emergency stop initiated
Cargo is raised
Crane rotates

Figure 6. Fault trees of the load collision scenario. a) Exemplary fault tree before simulation analysis b) Adapted fault tree by simulation-analysis results. 1: Adapted existing fault tree element 2: New failure combination discovered by simulation

Using this setup, we tested our framework by recording the number of simulations needed to find the missing failure which could lead to the overall hazard that the cargo hits the lift supervisor. Empirically, averaging across 50 iterations, we found that 6.43 simulations were needed to identify the missing hazard. Such few number of simulations, however, will in general not suffice in other situations in which the probability of an overall hazardous event is much lower. In contrast, the high probability here was chosen for demonstration purposes and therefore does not require the use specialized simulation methods to detect rare events.

IV. Conclusion and Future Work

In this paper, we have presented a simulation-based approach to identify potentially missing failures of a given fault tree that represents a critical scenario in the maritime domain. Importantly, the presented method comes with a confidence statement about the statistical certainty about the existence of failures which have not been identified by the simulation. To provide sufficient evidence for a complete characterization of potential failures, a large number of simulation runs are required. Although this is a well-known problem in statistical model checking, setting up appropriate rare event simulations remain subject to future research. However, these confidence statements assume the correctness of the underlying model for simulation. The check for failures should be seen as a means for finding additional critical situations and therefore acts as a completeness check of a previous risk analysis. Nevertheless, having a simulation model at hand also allows for further analysis methods, which unlike the presented method can also access risks during an (offshore) operation by initializing the simulation with the current status and specifically search for potential risks in the near future conditional on the current situation. Again, to bound the number of necessary simulation runs and acquire sufficient information about the current level of risk, techniques from rare event simulation can be used. In particular, importance splitting (cf. Lagnoux [21]) can be used to guide the simulation to potentially more risky situations.

To this end, an artificial risk level has to be defined, which heuristically estimates the risk level and guides the simulation. Defining such heuristic and integrate a corresponding importance sampling technique into the presented approach will therefore be the focus of future research.

ACKNOWLEDGMENTS

This work was partially supported by the *European Regional Development Fund* (ERDF) within the project *Safe Offshore Operations* (SOOP) [31].

REFERENCES

[1] K. E. Thomsen, Offshore Wind: A Comprehensive Guide to Successful Offshore Wind Farm Installation. Amsterdam: Elsevier Ltd, Oxford, 2011.

[2] Guidelines for Onshore and Offshore Wind Farms – Health & Safety in the Wind Energy Industry Sector, RenewableUK, London, 2010.

[3] Guidelines for Onshore and Offshore Wind Farms – Health & Safety in the Wind Energy Industry Sector, RenewableUK, London, 2013.

[4] W. Vesely, Fault Tree Handbook with Aerospace Applications. Washington DC: Nasa, 2002.

[5] S. Bangsow, Manufacturing Simulation with Plant Simulation and SimTalk. Springer Science & Business Media, 2010.

[6] R. Droste, C. Läsche, C. Sobiech, E. Böde, and A. Hahn, "Model-based risk assessment supporting development of hse plans for safe offshore operations." in Formal Methods for Industrial Critical Systems, ser. Lecture Notes in Computer Science, M. Stoelinga and R. Pinger, Eds., vol. 7612. Springer Berlin / Heidelberg, 2012, pp. 146–161.

[7] OMG, Business Process Model and Notation (BPMN) Version 2.0, 01 2011.

[8] ——, Unified Modeling Language (UML) Version 2.4.1, 08 2011.

[9] C. Läsche, R. Droste, J. Pinkowski, S. Gerwinn, and A. Hahn, "Model-based risk assessment of offshore operations," in Proceedings 33rd International Conference on Ocean, Offshore and Arctic Engineering. ASME, 2014, proceedings available CD-ROM.

[10] J. E. Vinnem, Offshore Risk Assessment. London: Springer Series in Reliability Engineering, 2007.

[11] M. M. Lavasani, J. Wang, Z. Yang, and J. Finlay, "Application of fuzzy fault tree analysis on oil and gas offshore pipelines," Int. J. Mar. Sci. Eng, vol. 1, no. 1, 2011, pp. 29–42.

[12] I. S. Sutton, Offshore safety management: implementing a SEMS program. William Andrew Publishing, 2011.

[13] M. L. McKelvin Jr, G. Eirea, C. Pinello, S. Kanajan, and A. L. Sangiovanni-Vincentelli, "A formal approach to fault tree synthesis for the analysis of distributed fault tolerant systems," in Proceedings of the 5th ACM international conference on Embedded software. ACM, 2005, pp. 237–246.

[14] F. Tajarrod and G. Latif-Shabgahi, "A novel methodology for synthesis of fault trees from matlab-simulink model," World Academy of Science, Engineering and Technology, vol. 41, 2008, pp. 630–636.

[15] B. Chen, "Improving processes using static analysis techniques," Ph.D. dissertation, University of Massachusetts Amherst, 2011.

[16] G. J. Pai and J. B. Dugan, "Automatic synthesis of dynamic fault trees from uml system models," in Software Reliability Engineering, 2002. ISSRE 2003. Proceedings. 13th International Symposium on. IEEE, 2002, pp. 243–254.

[17] G. Schellhorn, A. Thums, W. Reif et al., "Formal fault tree semantics," in Proceedings of The Sixth World Conference on Integrated Design & Process Technology, Pasadena, CA, 2002, pp. 1–8.

[18] F. Ortmeier and G. Schellhorn, "Formal fault tree analysis-practical experiences," Electronic Notes in Theoretical Computer Science, vol. 185, 2007, pp. 139–151.

[19] OMG, Documents Associated With Object Constraint Language, Version 2.0, 05 2006.

[20] International Electrotechnical Commission , IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems, 2010.

[21] A. Lagnoux, "Rare event simulation," Probability in the Engineering and Informational Sciences, vol. 20, 1 2006, pp. 45–66.

[22] D. Reijsbergen, P.-T. de Boer, W. Scheinhardt, and B. Haverkort, "Rare event simulation for highly dependable systems with fast repairs," in Proceedings of the Seventh International Conference on Quantitative Evaluation of SysTems, QEST 2010. Los Alamitos: IEEE Press, July 2010, pp. 251–260.

[23] P. Zuliani, C. Baier, and E. Clarke, "Rare-event verification for stochastic hybrid systems," in Hybrid Systems: Computation and Control, 2012, pp. 217–225.

[24] C. Jegourel, A. Legay, and S. Sedwards, "Importance splitting for statistical model checking rare properties," in Computer Aided Verification, 2013, pp. 1–16.

[25] C. Läsche, V. Gollücke, and A. Hahn, "Using an hla simulation environment for safety concept verification of offshore operations." in ECMS, W. Rekdalsbakken, R. T. Bye, and H. Zhang, Eds. European Council for Modeling and Simulation, 2013, pp. 156–162.

[26] gamekit – A cross-platform 3D game engine using Ogre or Irrlicht and Bullet for Windows, Linux, Mac, Android and iPhone. [retrieved 07, 2014]. [Online]. Available: https://code.google.com/p/gamekit/

[27] OGRE – Open Source 3D Graphics Engine. [retrieved 07, 2014]. [Online]. Available: http://www.ogre3d.org/

[28] Bullet – Real-Time Physics Simulation. [retrieved 07, 2014]. [Online]. Available: http://bulletphysics.org/

[29] S. Schweigert, R. Droste, and A. Hahn, "Multi-Agenten basierte 3D Simulation für die Evaluierung von Offshore Operationen," in Go-3D 2012 – »Computergraphik für die Praxis«, 2012, pp. 105–126.

[30] blender.org – Home of the Blender project. [retrieved 07, 2014]. [Online]. Available: http://www.blender.org/

[31] soop.offis.de - Sichere Offshore-Operationen. [retrieved 07, 2014]. [Online]. Available: http://soop.offis.de/

Investigation of Solver Technologies for the Simulation of Brittle Materials

Using the Example of Bullet-Proof Glass

Arash Ramezani and Hendrik Rothe
Chair of Measurement and Information Technology
University of the Federal Armed Forces
Hamburg, Germany
Email: ramezani@hsu-hh.de, rothe@hsu-hh.de

***Abstract*—Since computers and software have spread into all fields of industry, extensive efforts are currently made in order to improve the safety by applying certain numerical solutions. For many engineering problems involving shock and impact, there is no single ideal numerical method that can reproduce the various regimes of a problem. An approach wherein different techniques may be applied within a single numerical analysis can provide the “best” solution in terms of accuracy and efficiency. This paper presents a set of numerical simulations of ballistic tests which analyze the effects of soda lime glass laminates. The goal is to find an appropriate solver technique for simulating brittle materials and thereby improve bullet-proof glass to meet current challenges. To have the correct material model available is not enough. In this work, the main solver technologies are compared to create a perfect simulation model for soda lime glass laminates. The calculation should match ballistic trials and be used as the basis for further studies. These numerical simulations are performed with the nonlinear dynamic analysis computer code ANSYS AUTODYN.**

Keywords-solver technologies; simulation models; brittle materials; optimization; armor systems.

I. Introduction

In the security sector, the partly insufficient safety of people and equipment due to failure of industrial components are ongoing problems that cause great concern. Since computers and software have spread into all fields of industry, extensive efforts are currently made in order to improve the safety by applying certain computer-based solutions. To deal with problems involving the release of a large amount of energy over a very short period of time, e.g., explosions and impacts, there are three approaches: As the problems are highly non-linear and require information regarding material behavior at ultra-high loading rates which is generally not available, most of the work is experimental and may cause tremendous expenses. Analytical approaches are possible if the geometries involved are relatively simple and if the loading can be described through boundary conditions, initial conditions, or a combination of the two. Numerical solutions are far more general in scope and remove any difficulties associated with geometry [1].

For structures under shock and impact loading, numerical simulations have proven to be extremely useful. They provide a rapid and less expensive way to evaluate new design ideas. Numerical simulations can supply quantitative and accurate details of stress, strain, and deformation fields that would be very costly or difficult to reproduce experimentally. In these numerical simulations, the partial differential equations governing the basic physics principles of conservation of mass, momentum, and energy are employed. The equations to be solved are time-dependent and nonlinear in nature. These equations, together with constitutive models describing material behavior and a set of initial and boundary conditions, define the complete system for shock and impact simulations.

The governing partial differential equations need to be solved in both time and space domains. The solution over the time domain can be achieved by an explicit method. In the explicit method, the solution at a given point in time is expressed as a function of the system variables and parameters, with no requirements for stiffness and mass matrices. Thus, the computing time at each time step is low but may require numerous time steps for a complete solution.

The solution for the space domain can be obtained utilizing different spatial discretizations, such as Lagrange [2], Euler [3], Arbitrary Lagrange Euler (ALE) [4], or mesh free methods [5]. Each of these techniques has its unique capabilities, but also limitations. Usually, there is not a single technique that can cope with all the regimes of a problem [6].

This work will focus on brittle materials and transparent armor (consisting of several layers of soda lime float glass bonded to a layer of polycarbonate to produce a glass laminate). Using a computer-aided design (CAD) neutral environment that supports direct, bidirectional and associative interfaces with CAD systems, the geometry can be optimized successively. Native CAD geometry can be used directly, without translation to IGES or other intermediate geometry formats [7].

The work will also provide a brief overview of ballistic tests to offer some basic knowledge of the subject, serving as a basis for the comparison and verification of the simulation results. Details of ballistic trials on transparent armor systems are presented. Here, even the crack formation must precisely match later simulations. It was possible to observe crack motion and to accurately measure crack velocities in glass laminates. The measured crack velocity is a complicated function of stress and of water vapor concentration in the environment [8].

The objective of this work is to compare current solver technologies to find the most suitable simulation model for brittle materials. Lagrange, Euler, ALE, and “mesh free” methods, as well as coupled combinations of these methods, are described and applied to a bullet-proof glass laminate structure impacted by a projectile.

The results shall be used to improve the safety of ballistic glasses. Instead of running expensive trials, numerical simulations should be applied to identify vulnerabilities of structures. Contrary to the experimental results, numerical methods allow easy and comprehensive studying of all mechanical parameters. Modeling will also help to understand how the transparent armor schemes behave during impact and how the failure processes can be controlled to our advantage. By progressively changing the composition of several layers and the material thickness, the transparent armor will be optimized.

After a brief introduction and description of the different methods of space discretization, there is a short section on ballistic trials, where the experimental set-up is depicted. The last section describes the numerical simulations. These paragraphs of analysis are followed by a conclusion.

II. STATE-OF-THE-ART

First approaches for optimization were already developed in 1999. Mike Richards, Richard Clegg, and Sarah Howlett investigated the behavior of glass laminates in various configurations at a constant total thickness [9]. Resulting from the experimental studies, numerical simulations were created and adjusted to the experimental results using 2D-Lagrange elements only.

Pyttel, Liebertz and Cai explore the behavior of glass upon impact with three-dimensional Lagrange elements [10]. In 2011, these studies were used to analyze crash behavior.

In the same year, Zang and Wang dealt with the impact behavior on glass panels in the automotive sector [11]. In doing so, self-developed methods of numerical simulation were supposed to be compared with commercial codes. For the first time, mesh-free methods were applied, although these were not coupled with other solver technologies.

In this study, different methods for the simulation of safety glass will be introduced. In so doing, the possibility of coupling various solver technologies will be discussed and illustrated by means of an example. For the first time, glass laminates will be modeled using coupled methods. Techniques previously applied, show considerable shortcomings in portraying the crack and error propagation in the glass. Mesh-free approaches, in turn, do not correctly present the behavior of synthetic materials. To overcome the shortcomings of these single-method approaches, this paper will present an optimal solution to the problem by combining two methods.

III. METHODS OF SPACE DISCRETIZATION

The spatial discretization is performed by representing the fields and structures of the problem using computational points in space, usually connected with each other through computational grids. Generally, the following applies: the finer the grid, the more accurate the solution. For problems of dynamic fluid-structure interaction and impact, there typically is no single best numerical method which is applicable to all parts of a problem. Techniques to couple types of numerical solvers in a single simulation can allow the use of the most appropriate solver for each domain of the problem [12]. The most commonly used spatial discretization methods are Lagrange, Euler, ALE (a mixture of Lagrange and Euler), and mesh-free methods, such as Smooth Particles Hydrodynamics (SPH) [13].

A. Lagrange

The Lagrange method of space discretization uses a mesh that moves and distorts with the material it models as a result of forces from neighboring elements (meshes are imbedded in material). There is no grid required for the external space, as the conservation of mass is automatically satisfied and material boundaries are clearly defined. This is the most efficient solution methodology with an accurate pressure history definition. The Lagrange method is most appropriate for representing solids, such as structures and projectiles. If however, there is too much deformation of any element, it results in a very slowly advancing solution and is usually terminated because the smallest dimension of an element results in a time step that is below the threshold level.

B. Euler

The Euler (multi-material) solver utilizes a fixed mesh, allowing materials to flow (advect) from one element to the next (meshes are fixed in space). Therefore, an external space needs to be modeled. Due to the fixed grid, the Euler method avoids problems of mesh distortion and tangling that are prevalent in Lagrange simulations with large flows. The Euler solver is very well-suited for problems involving extreme material movement, such as fluids and gases. To describe solid behavior, additional calculations are required to transport the solid stress tensor and the history of the material through the grid. Euler is generally more computationally intensive than Lagrange and requires a higher resolution (smaller elements) to accurately capture sharp pressure peaks that often occur with shock waves.

C. ALE

The ALE method of space discretization is a hybrid of the Lagrange and Euler methods. It allows redefining the grid continuously in arbitrary and predefined ways as the calculation proceeds, which effectively provides a continuous rezoning facility. Various predefined grid motions can be specified, such as free (Lagrange), fixed (Euler), equipotential, equal spacing, and others. The ALE method can model solids as well as liquids. The advantage of ALE is the ability to reduce and sometimes eliminate difficulties caused by severe mesh distortions encountered by the Lagrange method, thus allowing a calculation to continue efficiently. However, compared to Lagrange, an additional computational step of rezoning is employed to move the grid and remap the solution onto a new grid [6].

D. SPH

The mesh-free Lagrangian method of space discretization (or SPH method) is a particle-based solver and was initially used in astrophysics. The particles are imbedded in material and they are not only interacting mass points but also interpolation points used to calculate the value of physical variables based on the data from neighboring SPH particles, scaled by a weighting function. Because there is no grid defined, distortion and tangling problems are avoided as well. Compared to the Euler method, material boundaries and interfaces in the SPH are rather well defined and material separation is naturally handled. Therefore, the SPH solver is ideally suited for certain types of problems with extensive material damage and separation, such as cracking. This type of response often occurs with brittle materials and hypervelocity impacts. However, mesh-free methods, such as Smooth Particles Hydrodynamics, can be less efficient than mesh-based Lagrangian methods with comparable resolution.

Figure 1 gives a short overview of the solver technologies mentioned above. The crucial factor is the grid that causes different outcomes.

The behavior (deflection) of the simple elements is well-known and may be calculated and analyzed using simple equations called shape functions. By applying coupling conditions between the elements at their nodes, the overall stiffness of the structure may be built up and the deflection/distortion of any node – and subsequently of the whole structure – can be calculated approximately [15].

Due to the fact that all engineering simulations are based on geometry to represent the design, the target and all its components are simulated as CAD models [16]. Therefore, several runs are necessary: from modeling to calculation to the evaluation and subsequent improvement of the model (see Figure 2).

The most important steps during an FE analysis are the evaluation and interpretation of the outcomes followed by suitable modifications of the model. For that reason, ballistic trials are necessary to validate the simulation results. They can be used as the basis of an iterative optimization process.

IV. BALLISTIC TRIALS

Ballistics is an essential component for the evaluation of our results. Here, terminal ballistics is the most important sub-field. It describes the interaction of a projectile with its target. Terminal ballistics is relevant for both small and large caliber projectiles. The task is to analyze and evaluate the impact and its various modes of action. This will provide information on the effect of the projectile and the extinction risk.

In order to develop a numerical model, a ballistic test program is necessary. The ballistic trials are thoroughly documented and analyzed – even fragments must be collected. They provide information about the used armor and the projectile behavior after fire which must be consistent with the simulation results.

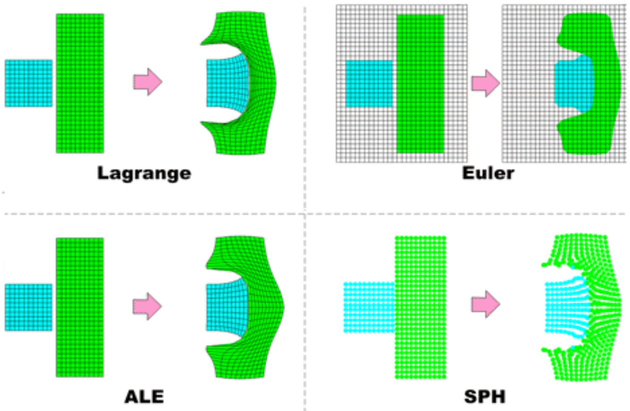


Figure 1. Examples of Lagrange, Euler, ALE, and SPH simulations on an impact problem [14].

In order to create a data set for the numerical simulations, several experiments have to be performed. Ballistic tests are recorded with high-speed videos and analyzed afterwards. The experimental set-up is shown in Figure 3. Testing was undertaken at an indoor ballistic testing facility. The target stand provides support behind the target on all four sides. Every ballistic test program includes several trials with different glass laminates. The set-up has to remain unchanged.

The camera system is a pco.dimax that enables fast image rates of 1279 frames per second (fps) at full resolution of 2016 x 2016 pixels. The use of a polarizer and a neutral density filter is advisable, so that waves of some polarizations can be blocked while the light of a specific polarization can be passed.

Several targets of different laminate configurations were tested to assess the ballistic limit and the crack propagation for each design. The ballistic limit is considered the velocity required for a particular projectile to reliably (at least 50% of the time) penetrate a particular piece of material [17]. After the impact, the projectile is examined regarding any kind of change it might have undergone.

Figure 4 shows a 23 mm soda lime glass target after testing. The penetrator used in this test was a .44 Remington Magnum, a large-bore cartridge with a lead base and copper jacket. The glass layers showed heavy cracking as a result of the impact.

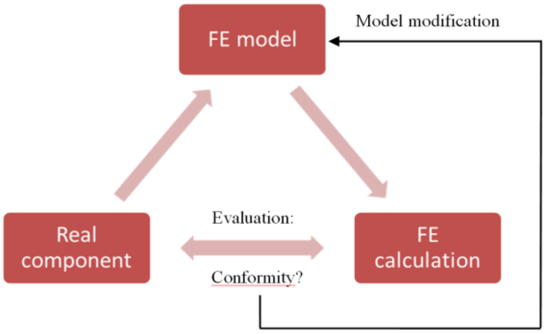


Figure 2. Iterative procedure of a typical FE analysis [15].

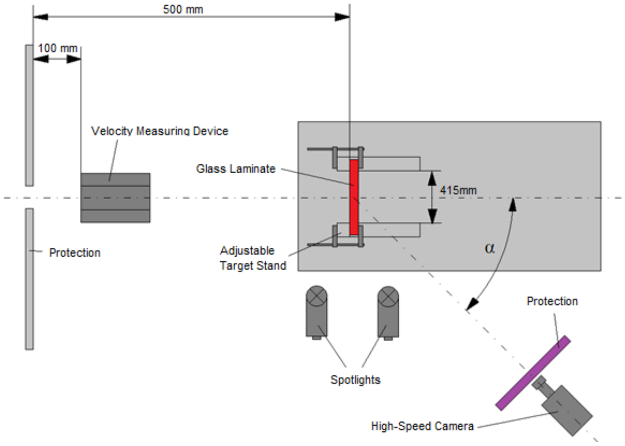


Figure 3. Experimental set-up.

Close to the impact point is the region of comminution. The comminuted glass is even ejected during the impact. Radial cracks have propagated away from the impact point. The polycarbonate backing layer is deformed up to the maximum bulge height when the velocity of the projectile is close to the ballistic limit. A large amount of the comminuted glass is ejected during the impact. Several targets of different laminate configurations were tested to assess the ballistic limit and the crack propagation for each design.

Cracks propagate with a velocity up to 2500 m/s, which is similar to the values in the literature. The damage of a single glass layer starts with the impact of the projectile corresponding to the depth of the penetration. The polycarbonate layers interrupt the crack propagation and avoid piercing and spalling. The first impact of a .44 Remington cartridge does not cause a total failure of our 23 mm soda lime glass target. Fragments of the projectile can be found in the impact hole. The last polycarbonate layer remains significantly deformed.

The results of the ballistic tests were provided prior to the simulation work to aid calibration. In this paper, a single trial will illustrate the general approach of the numerical simulations.

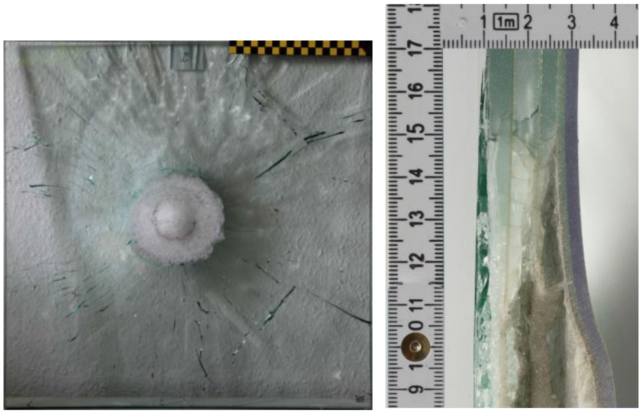


Figure 4. Trial observation with a 23mm glass laminate.

V. Numerical Simulation

The ballistic tests are followed by computational modeling of the experimental set-up. Then, the experiment is reproduced using numerical simulations. Figure 5 shows a cross-section of the ballistic glass and the projectile in a CAD model. The geometry and observed response of the laminate to ballistic impact is approximately symmetric to the axis through the bullet impact point. Therefore, a 2D axisymmetric approach was chosen.

Numerical simulation of transparent armor requires the selection of appropriate material models for the constituent materials and the derivation of suitable material model input data. The laminate systems studied here consist of soda lime float glass, polyurethane interlayer, polyvinyl butyral and polycarbonate. Lead and copper are also required for the .44 Remington Magnum cartridge.

The projectile was divided into two parts - the jacket and the base - which have different properties and even different meshes. These elements have quadratic shape functions and nodes between the element edges. In this way, the computational accuracy as well as the quality of curved model shapes increases. Using the same mesh density, the application of parabolic elements leads to a higher accuracy compared to linear elements (1st order elements).

Different solver technologies have been applied to the soda lime glass laminate. The comparison is presented in the following chapter.

A. *Solver Evaluation*

Before the evaluation starts, it has to be noticed that the Euler method is not suitable for numerical simulations dealing with brittle materials. It is generally used for representing fluids and gases, for example, the gas product of high explosives after detonation. To describe solid behavior, additional calculations are required. Cracking cannot be simulated adequately and the computation time is relatively high. For this reason, the Euler (and as a result the ALE) method will not be taken into consideration.

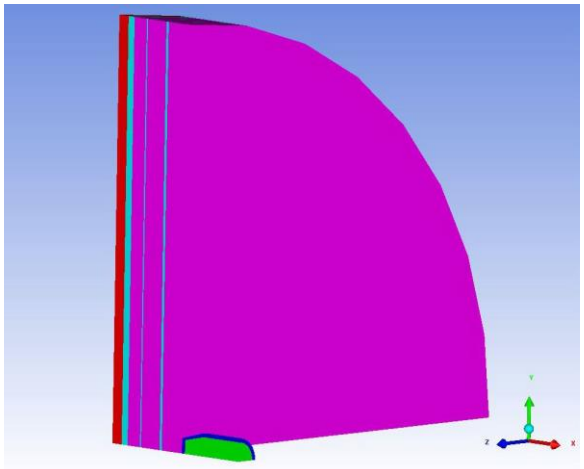


Figure 5. CAD model.

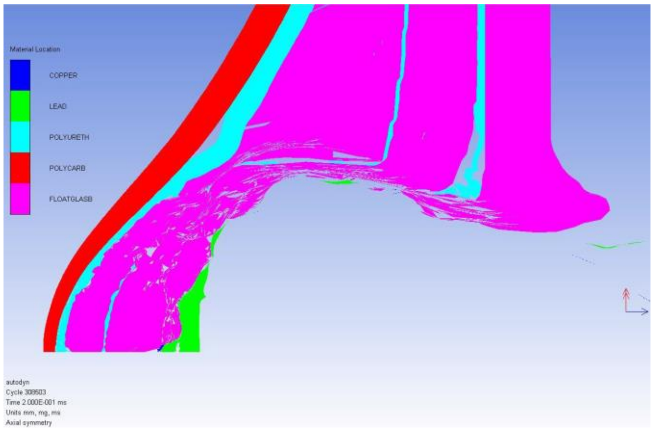


Figure 6. Lagrange method.

1) Lagrange method: Figure 6 shows the simulation with a single Lagrange solver in the first iteration procedure. This method, as mentioned before, is well-suited for representing solids like structures and projectiles. The advantages are computational efficiency and ease of incorporating complex material models. The polyurethane interlayer, polyvinyl butyral and polycarbonate are simulated adequatly. While the soda lime glass also deforms well, the crack propagation cannot be displayed suitably with this solver.

2) Mesh free Lagrangian method (SPH): The mesh free Lagrangian method is not appropiate for simulating bullet-proof glass. The crack propagation and failure mode of the soda lime glass are very precise. The problem here however is the simulation of the layers. The particles do not provide the necessary cohesion (see Figure 7). They break easily and then lose their function. However, the SPH method requires some of the particles to locate current neighboring particles, which makes the computational time per cycle more expensive than mesh based Lagrangian techniques. Also, the mesh free method is less efficient than mesh based Lagrangian methods with comparable resolution.

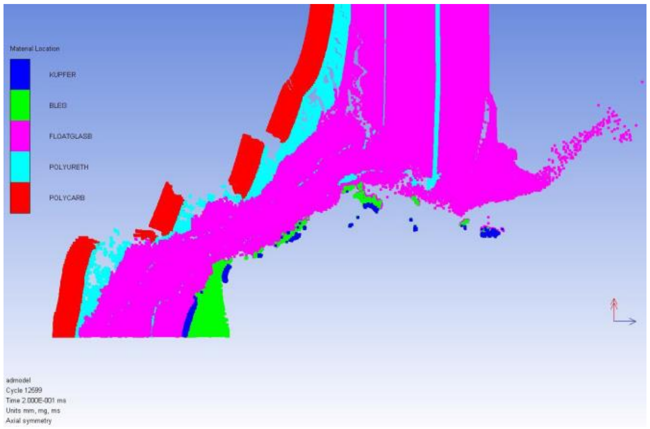


Figure 7. Mesh free Lagrangian method (SPH).

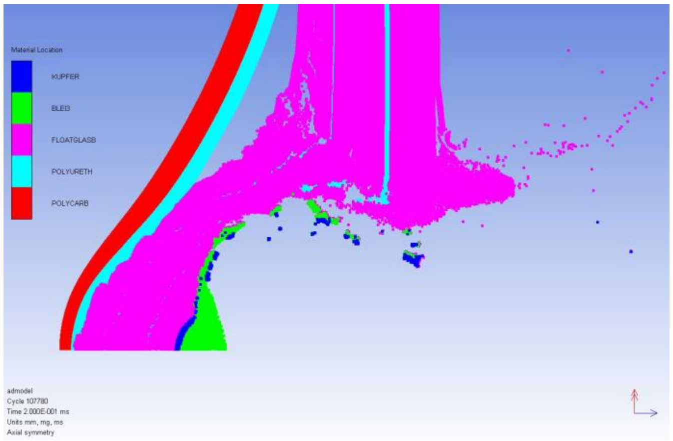


Figure 8. Coupled multi-solver approach (Lagrange and SPH).

3) Coupled multi-solver approach (Lagrange and SPH): The coupled multi-solver approach uses SPH for the soda lime glass and Langrange for the polyurethane interlayer, polyvinyl butyral and polycarbonate. The crack propagation can be simulated precisely. The deformation of the last layer is accurately displayed and the failure mode matches the ballistic trial. Figure 8 illustrates the simulation result for this case.

B. Simulation Results

With the coupled multi-solver and optimized material parameters, the simulation results adequately mirror the observations made in the ballistic experiments. Fragmentation and crack propagation are almost equal to the ballistic test shown in Figure 4.

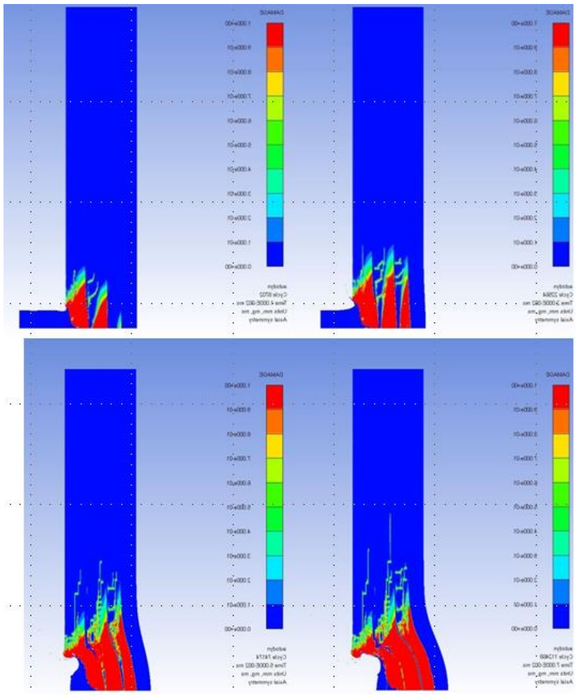


Figure 9. Crack propagation in a coupled multi-solver simulation model.

Figure 9 illustrates the development of fracture after 10, 20, 50, and 70 µs due to shear induced micro-cracking (damage) in the glass during the penetration process. Note that the failure of the glass in the second and third layers spreads from the glass / polyurethane interlayers back towards the oncoming projectile. This rapid material failure is owed to a reduction in material strength as rarefaction waves from the interface reduce the confining pressure [17].

Small fragments are automatically deleted from the program to reduce computing time. Regarding the protection level of our structures, these fragments are hardly important.

The projectile is subject to a significant deformation. It gets stuck in the target and loses kinetic energy. Figure 10 compares the numerical simulation of a .44 Remington impact with the experimental result.

A clear hole, 45-50 mm in diameter, is generated in the glass / polyurethane layers of the laminate. A comminuted region of glass, shows highly cracked and completely crushed material, of around 20 mm in diameter in the first layer which extends to around 120 mm in diameter in the last layer. Hence, the simulated diameter of comminution is almost identical to that observed experimentally. Even the delamination of the layers can be reproduced in the simulation. The predicted height of the bulge from the flat region of the polycarbonate is 28 mm compared to approximately 8 mm observed in the ballistic trials. In the simulation, comminuted glass is caught between the bullet and the polycarbonate layer. This leads to a larger deformation. In reality, comminuted glass is ejected during the impact. The polycarbonate dishes from the edge of the support clamp to form a prominent bulge in the central region. Therefore, reducing the instantaneous geometric erosion strain of the soda lime glass will significantly improve results. Owed to the adopted calibration process, these simulation results correlate well with the experimental observations.

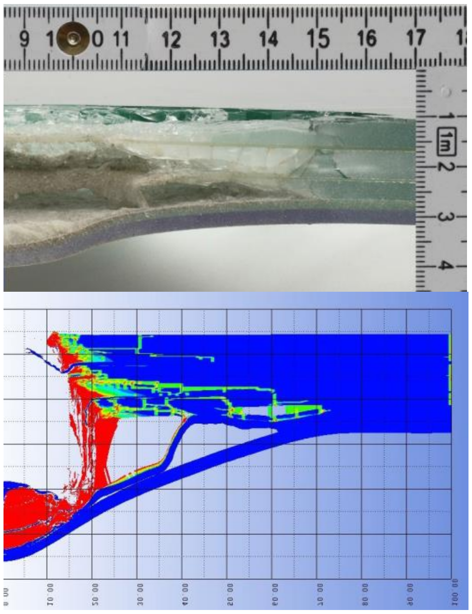


Figure 10. Comparison between simulation results and ballistic trial.

VI. Conclusion

This work demonstrated how a small number of well-defined experiments can be used to develop, calibrate and validate solver technologies used for simulating the impact of projectiles on complex armor systems and brittle materials.

Existing material models were optimized to reproduce ballistic tests. High-speed videos were used to analyze the characteristics of the projectile – before and after the impact. The simulation results demonstrate the successful use of the coupled multi-solver approach. The high level of correlation between the numerical results and the available experimental or observed data demonstrates that the coupled multi-solver approach is an accurate and effective analysis technique.

New concepts and models can be developed and easily tested with the help of modern hydrocodes. The initial design approach of the units and systems has to be as safe and optimal as possible. Therefore, most design concepts are analyzed on the computer. FEM-based simulations are well-suited for this purpose. Here, a numerical model has been developed which is capable of predicting the ballistic performance of soda lime glass / polycarbonate transparent armor systems. Thus, estimates based on experience are being more and more replaced by software.

The gained experience is of prime importance for the development of modern armor. By applying the numerical model a large number of potential armor schemes can be evaluated and the understanding of the interaction between laminate components under ballistic impact can be improved.

References

[1] J. Zukas, “Introduction to Hydrocodes,” Elsevier Science, February 2004.

[2] A. M. S. Hamouda and M. S. J. Hashmi, “Modelling the impact and penetration events of modern engineering materials: Characteristics of computer codes and material models,” Journal of Materials Processing Technology, vol. 56, pp. 847–862, Jan. 1996.

[3] D. J. Benson, “Computational methods in Lagrangian and Eulerian hydrocodes,” Computer Methods in Applied Mechanics and Engineering, vol. 99, pp. 235–394, Sep. 1992, doi: 10.1016/0045-7825(92)90042-I.

[4] M. Oevermann, S. Gerber and F. Behrendt, “Euler-Lagrange/DEM simulation of wood gasification in a bubbling fluidized bed reactor,” Particuology, vol. 7, pp. 307-316, Aug. 2009, doi: 10.1016/j.partic.2009.04.004.

[5] D. L. Hicks and L. M. Liebrock, “SPH hydrocodes can be stabilized with shape-shifting,” Computers & Mathematics with Applications, vol. 38, pp. 1-16, Sep. 1999, doi: 10.1016/S0898-1221(99)00210-2.

[6] X. Quan, N. K. Birnbaum, M. S. Cowler and B. I. Gerber, “Numerical Simulations of Structural Deformation under Shock and Impact Loads using a Coupled Multi-Solver Approach,” 5th Asia-Pacific Conference on Shock and Impact Loads on Structures, Hunan, China, Nov. 2003, pp. 152-161.

[7] N. V. Bermeo, M. G. Mendoza and A. G. Castro, “Semantic Representation of CAD Models Based on the IGES Standard,” Computer Science, vol. 8265, pp. 157-168, Dec. 2001, doi: 10.1007/ 978-3-642-45114-0_13.

[8] S. M. Wiederhorn, “Influence of Water Vapor on Crack Propagation in Soda-Lime Glass,” Journal of the American Ceramic Society, vol. 50, pp. 407-414, Aug. 1967, doi: 10.1111/j.1151-2916.1967.tb15145.x.

[9] M. Richards, R. Clegg, and S. Howlett, “Ballistic Performance Assessment of Glass Laminates Through Experimental and Numerical Investigation,” 18th International Symposium and Exhibition on Ballistics, San Antonio, Texas, Nov. 1999, pp. 1123-1131.

[10] T. Pyttel, H. Liebertz and J. Cai, “Failure criterion for laminated glass under impact loading and its application in finite element simulation,” International Journal of Impact Engineering, vol. 38, pp. 252-263, April 2011, doi: 10.1007/s00466-007-0170-1.

[11] M. Y. Zang, Z. Lei and S. F. Wang, “Investigation of impact fracture behavior of automobile laminated glass by 3D discrete element method,” Computational Mechanics, vol. 41, pp. 78-83, Dec. 2007, doi: 10.1007/s00466-007-0170-1.

[12] G. S. Collins, “An Introduction to Hydrocode Modeling,” Applied Modelling and Computation Group, Imperial College London, August 2002, unpublished.

[13] R. F. Stellingwerf and C. A. Wingate, “Impact Modeling with Smooth Particle Hydrodynamics,” International Journal of Impact Engineering, vol. 14, pp. 707–718, Sep. 1993.

[14] ANSYS Inc. Available Solution Methods. [Online]. Available from: http://www.ansys.com/Products/Simulation+Technology/Structural+Analysis/Explicit+Dynamics/Features/Available+Solution+Methods [retrieved: April, 2014]

[15] P. Fröhlich, “FEM Application Basics,” Vieweg Verlag, September 2005.

[16] H. B. Woyand, “FEM with CATIA V5,” J. Schlembach Fachverlag, April 2007.

[17] D. E. Carlucci and S. S. Jacobson, “Ballistics: Theory and Design of guns and ammunition,” CRC Press, Dec. 2008.

Improving Simulators Quality Using Model and Data Validation Techniques

Industrial Project Examples

Vesselin Gueorguiev
Technical University Sofia,
Sofia, Bulgaria
e-mail: veg@tu-sofia.bg

***Abstract* - This paper presents ideas and approaches to design, implementation and validation of simulators using the program generation approach. This approach is well presented in today's industrial control systems configurators. It has now extensive use also in embedded systems design and implementation. Unfortunately, the validation of the generated control systems usually is not part of the generating tool. Results of simulator implementation using the program generation approach and their validation are presented. Analyses of problems found and the steps for problem fixing are presented in the context of increasing the system model reliability. This paper also presents the validation of the program generation approach for design and implementation of object simulators and how the quality of the simulator can be increased using validation techniques.**

Keywords – simulation validation; program generation; re-design after validation; model reliability.

I. INTRODUCTION

The area of Validation and Verification (V&V) of system design and system modelling and simulation has been a hot topic for the last more than three decades [1][2]. A huge number of papers have been written on it. A large number of tools for V&V of different complexity has been designed, implemented and used. In this paper some validation aspects will be discussed. Ways to improve the model of the simulated system will also be discussed.

It is clear that there is no perfect solution for any validation problem. This is based on the generic fact that human activities of any kind are not only formal (scientific), but informal (art) too. The titles of two basic (very formal) too often cited books help to prove this – *The Art of Computer Programming* by Donald Knuth [22] and *The Art of Simulation* by K.D. Tocher [23]. Nobody can eliminate all problems and errors in system models and their implementation; here, an approach will be presented how it will be possible to increase models' reliability and to decrease the risk of design and implementation failures. The approach 'I was smart, I thought' is always valid; but, for the engineering practice, is needed something much more stable. Elements of these approaches will be presented hereafter.

Simulator-based studies of theories, algorithms and their implementation validity today are old and extensively exploited topics. These include vehicle, flight, nuclear power plant, robots and many other simulators [6][7][8][9]. Computer-based simulators of many different complex objects are a reality today. First, they appeared in military, aircraft and nuclear power plant applications; now they are available everywhere. Simulation of various systems has a lot of advantages compared to experimenting and using the actual systems. Such an advantage is the possibility to train the personnel to operate various types of machines, or to use such simulations in preparation and optimization of control algorithms, or to repeat and analyse specific situations.

In the area of control systems design, there are many implementation approaches – from ad-hoc design and programming to fully automated code generation [10]. One specific approach to control systems implementation is the program generation. Using the same approach for simulator implementation is very promising, too. This approach reduces the amount of investments and risks in the design and implementation phases of the control system design. Simulators of that type are not used only for control system tests and improvements. Personnel training and abnormal situations analyses are other areas of their use.

In this paper we discuss not only validation of generated control systems, but validation of the program generation approach as a validated basis for control systems and simulators generation.

The paper is structured as follows: Section II presents validation techniques for simulators. Section III presents the program generator used for simulator generation and its formal model. Section IV presents validation of the implemented simulators; Section V is the conclusion.

II. VALIDATION TECHNIQUES FOR SIMULATORS

To check that a programmed system implements just what is designed, an analyst needs to verify that system. Many methods have been developed over time to achieve this [2][11]. Additionally, the analyst has to check that the implementation is free of errors. But one question has not been answered yet, namely whether the design is correct and if the conceptual model is an accurate representation of the system it has to represent.

Let us start with the presumption that the validation is a compulsory element of the activities and procedures for improving the quality of the generated software as well as that of the whole system. Validation can be a "process of evaluating a system or its components in the process of its development or at its end to determine whether they satisfy specified requirements" [12] or a "process of collection of information showing that both software and associated products satisfy the system requirements at the end of each cycle of its life, as well as satisfy some user needs in a specific case".

Simulators, as software products, can be created using every one of the four basic models for development: (i) development of a new product; (ii) modification; (iii) configuration / reconfiguration, and (iv) use of ready-for-use software applications (through the "Pipes & Filters" architecture). In case of simulators produced by program generation, the very good results gives the method for validation of product type “configuration / reconfiguration”. On this basis, a modification of the classical methodology for validation of this class of applications was implemented. The main characteristics of this new approach are the following:

Validation when creating a new software product using the "Configure/Reconfigure" approach

It is performed in two main ways:

- Validation of the basic software;
- Validation of the configuration data.

The validation of the basic software is done by using the existing software (by applying one of the two options).

- For each subsystem and for each module responsible for any of the functions of the new SW;
- For each operation of the basic SW which results in a change of configuration/reconfiguration of the SW.

Validation of the configuration data is similar to a new program validation.

Data validation

Validation of data is divided into two activities:

- Validation of data values;
- Validation of techniques and methods for data storage, modification and use.

In both cases, the validation has to determine whether the data are used correctly and whether they allow to develop SW that will function correctly.

Validation of data values

The idea is to determine the correctness of every data at every moment of the software operation.

There exist different forms, some of which are entirely manually implemented, and other are fully automated.

- Check if the input signals are received in the right places;
- Check whether the input signals comply with the criteria for quality and correctness;
- Check for correctness of the reactions generated by the input signals.

It is a good practice first to perform validation using specially developed test environment, and only after that perform it in the actual environment.

Validation of the data management system

These are validation techniques and methods for data storage, modification and use. They are applicable in all cases, whether there are automated tools for database management or not.

The selection of validation techniques depends on the type of stored information and its usage. For the purpose of validation of program generated simulators, only items that affect the validated operation are checked.

Validation as part of a system

The final validation results from the successful validation of the following elements:

- Evaluation as a part of the system;
- Validation as part of a technological process;
- Product validation.

This is very important in case of a “partial” or “semi-natural” simulator.

The features of validation of a software application as part of an integral system are:

- In this case, in addition to the validation of software, a number of activities to validate the integral system are performed;
- Evaluation of the relation 'Requirements to the requirements − requirements to the software';
- In some cases of application development (using pre-programed software pieces, configuration/ reconfiguration) validation on system level can be performed directly.

III. The Program Generator PrGen

The PrGen program generator is designed to create distributed real-time control systems. It is implemented in many different versions [13][14][15]. It is based on an extended Moore machine implementing specific actions in each node of the state machine. Specific elements of its design are reflecting possibilities to generate both stand-alone and distributed systems. A graph representation of the control algorithms is chosen. The system can be described by its activities. Each activity is a separate thread. After some model and library extensions, the PrGEn started to be used also for simulator generation.

The PrGen program generator is designed to answer the following requirements [14][15]:

- hybrid object/process configuration specification
- an object model facilitating the implementation of open and reconfigurable systems;
- a process model capturing both the reactive and the transformational aspects of system behavior in the context of various types of control systems;
- predictable scheduling of process execution and communication in the context of local and remote subsystems interactions;
- support for modern software engineering techniques such as program generation, formal verification of process and process interaction, etc.

As has been pointed out by Angelov and Ivanov [14] the specification of system reactions and signal transformation is very important for simulator quality and model validation. They can be formally presented as follows:

Specification of system reactions

System reactions can be defined as functions, specifying the output signals, generated in response to certain activating events. $R = \{ r_i \}$, where :

$$r_i : \quad A \times E \times C \to Y$$

It can be shown that the function r_i is a composite function, i.e., $r_i = o_i \circ p_i$, *where* p_i is a state transition function and o_i is an output function. These are defined below as follows:

- p_i is a state transition function that can be generally defined as: $p_i : A \times E \times C \to A$

Consequently, $\forall a_m \in A$ we can specify a subset of successor states:

$$Fa_m = \{a_{n_1}, a_{n_2}, \ldots\ldots, a_{n_r}\},$$

and accordingly – a subset of alternative transitions :

$$a_m(t_-)\ \&\ e^m_{n_1}(t)\ \&\ c^m_{n_1}(x(t)) \to a_{n_1}(t);$$
$$a_m(t_-)\ \&\ e^m_{n_2}(t)\ \&\ c^m_{n_2}(x(t)) \to a_{n_2}(t);$$
$$\ldots$$
$$a_m(t_-)\ \&\ e^m_{n_r}(t)\ \&\ c^m_{n_r}(x(t)) \to a_{n_r}(t).$$

This will ultimately result in the construction of the state transition graph of process *P*.

- o_i is an output function, i.e: $o_i: A \to Y$

Consequently, the reaction function r_i can be specified as a composition of the above two functions:

$$r_i : A \times E \times C \to Y \Leftrightarrow A \times E \times C \xrightarrow{p_i} A \xrightarrow{o_i} Y$$

Specifically, the above transformation implies that $\forall\ r_i \in R : r_i \leftrightarrow y_k = f(a(t))$, which follows the well known definition of the Moore machine.

Specification of sygnal transformations

Signal transformation functions specify how the output signals are generated within the corresponding system reactions and the associated process states. Specifically,

$$\forall\ y_k \in Y \leftrightarrow s_k \in S \text{ and;}$$

$$s_k : X_k(t) \mapsto y_k(t),\ X_k \subseteq X$$

In one specific case y_k = const. This case is common for a class of discrete controllers, i.e., the so-called state-logic controllers, whereby the controller generates predefined combinations of discrete *on/off* control signals, within various operators (reactions) associated with the corresponding controller states.

In the general case, y_k may be a complex function that can be represented as a composition of simple functions, i.e.:

$$y_k = w^k_l \circ w^k_{l-1} \circ w^k_{l-2} \ldots\ldots \circ w^k_1 ,$$

where w^i_j are basic application functions that are usually implemented as a library of standard function modules (FM's). The above composition corresponds to a sequence of computations, which can be described by means of two types of models:

(1) Conventional control flow model, such as flowcharts, computation graphs, etc.

(2) Data flow models, e.g., (quasi) analog signal flow diagrams.

The presented mathematical model meets all the formal requirements to it. Formally, an extended Moore machine implementing activities (actions) of analogue, discrete and communication type at any state node can represent most of the industrial systems and objects available today. It is not limited either by numerical transformations or by logical constructions.

IV. VALIDATION OF IMPLEMENTED SIMULATORS

Below, simulators of different objects and validation of their conceptual models and implementations will be presented.

Harbour crane

The simulated harbour crane is of the type shown in Figure 1. It has to be modelled for the following actions:

- A1: Crane with no container, turning to container and lowering hook.
- A2: Crane with container, lifting the container and turning to the ship.
- A3: Lowering the load and positioning the container into the ship.
- A4: Empty crane, from initial position (turned to the port). Turning to the ship, positioning above container, lifting load and unloading it into port.

Each action consists of several different movements in vertical and horizontal directions. Additional limits (dimensions) for horizontal and vertical movements are included.

The motivation to design and implement this simulator is that experiments with the real crane are dangerous and expensive and that the exploitation personnel need a full-time and functional training environment.

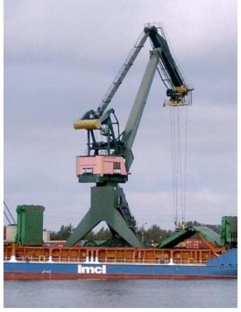


Figure 1. Harbor Crane

The crane has four different mechanical movements: turn around vertical axe, grapple lifting/lowering, grapple control and grapple horizontal shifting. Additionally the crane can move along its railway but it is out of scope of the presented simulator. The mechanical construction is independent for each of these movements. They can be started simultaneously but interviewing operators it was found that they run every movement separately. The explanation was that in the other case the crane can lose its stability. Because of this, limit for simultaneous movements, limits for horizontal and vertical speed, etc. have been incorporated in the simulator. A presentation of the MATLAB® models of one of the crane's movements is shown in Figure 2a and Figure 2b.

The simulator includes models for all movements, as well as simulation for operator's interface devices, predictor of the absolute (3D) grapple position and a load simulator.

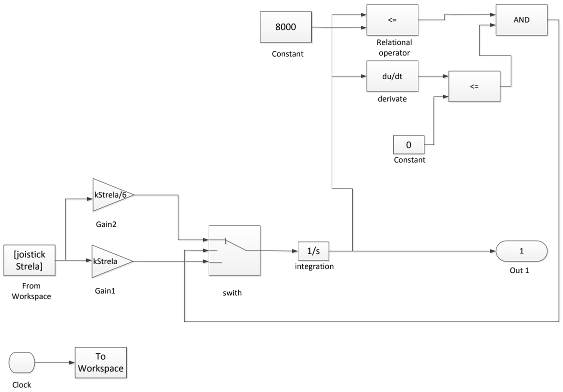


Figure 2a. Harbor Crane grapple non-linear model

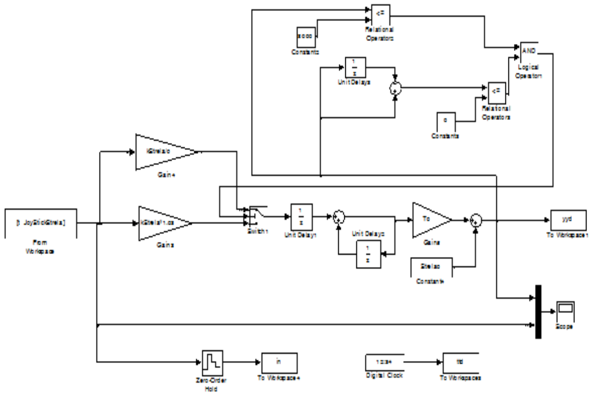


Figure 2b. Harbor Crane grapple discrete model

The motor control is implemented by drive controllers connected to the main controller via Profibus. To make simulation easier Profibus was substituted with MODBUS. The main controller supported both protocols. The simulation of Profibus connections by MODBUS is shown in Figure 3.

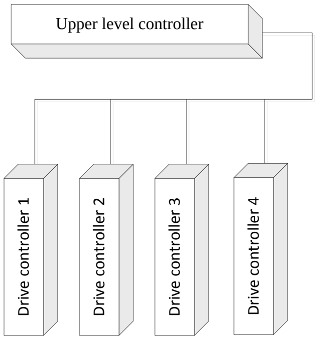


Figure 3. Harbor Crane Profibus/MODBUS model

Validation of the designed simulator envisaged that in the MATLAB® environment after structural and parametric identification the mathematical model and real data are acceptably close. After that a real simulator was implemented. It was developed using the PrGen program generator and its modules library. Real-time execution is done on single-board ARM computer equipped with peripheral devices and communication modules corresponding to the real object. Mathematical transformations are identical to these implemented in MATLAB. Surprisingly, the results from the simulator were very different from the MATLAB results. Detailed data validation was implemented. Finally it was found that the main difference came from the different accuracy in floating-point calculations. After passing incorrect data through non-linearities we received results different by 20% from the original ones. This problem was solved using calculations with extended accuracy. Sensitivity of some non-linear elements was changed as well. Results after simulator tuning for actions A1 to A4 are shown in Table 1.

TABLE 1. COMPARISON BETWEEN CRANE REAL DATA AND SIMULATOR OUTPUTS

N.	Measure-ments Crane (samples)	measure-ments Simulator (samples)	Min. Movement Acc. %	Max. Movement Acc. %	Total Acc. %
A 1	680,000	890,000	95	98	97
A 2	1, 280,000	1,460.000	94	97	95
A 3	1,800,000	1,960,000	95	98	97
A 4	3,640,000	4,000,000	93	97	96

Data validation using plotting is presented in Figure 4 and Figure 5. They show the similarity between real data and simulator outputs. They demonstrate the very high quality of the simulation. This made possible development and fine tuning of the crane control system and it also reduced development costs.

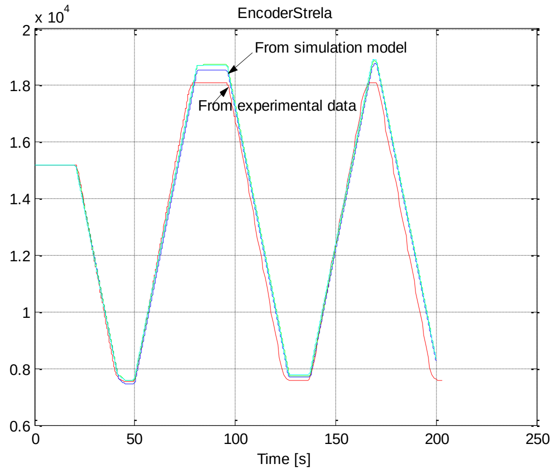


Figure 4. Harbor Crane simulator and experimental data comparison: simulation of grapple horizontal movement;

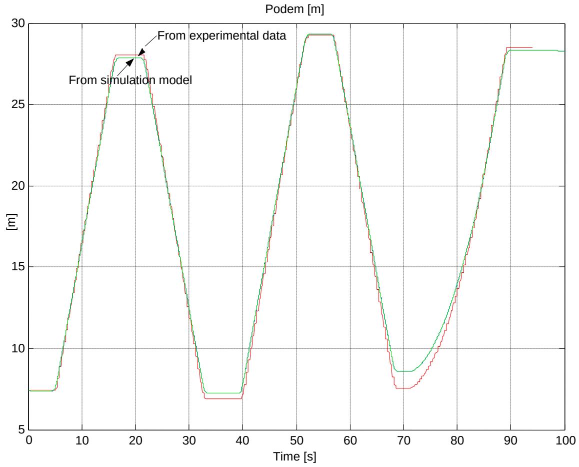


Figure 5. Harbor Crane simulator and experimental data comparison: simulation of grapple vertical movement

Large-scale textile plotter

Similar approach for validation of the quality of simulation was used for a large-scale textile plotter. The motivation to invest time and money in this was the requirements to combine maximal performance and geometrical accuracy.

Printing drawings with a length of 5 or more (up to 20) meters is one of the available solutions to prepare the fabric cutting sketch for real cutting. Several different solutions for this process are known; but one of the best is to draw this sketch on an unbreakable paper sheet with the required length. This is done by a special kind of plotters. Normally, they are of flat-bed type. The simulator of a plotter of that type and its validation will be presented here.

The scheme of the plotter is presented in Figure 6 and Figure 7.

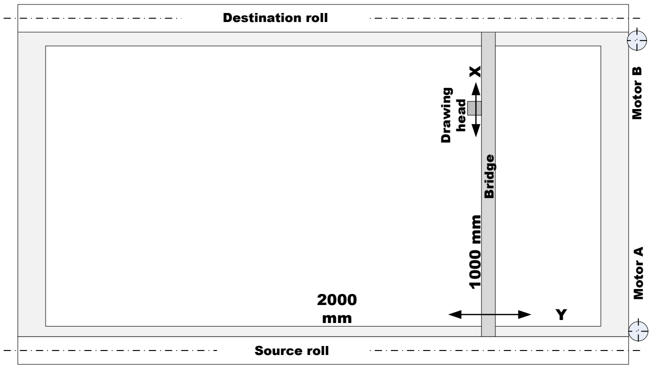


Figure 6. Large-scale textile plotter - top view

Kinematics and wiring schemes of the plotter are shown in Figure 8 and Figure 9. The closed loop kinematics, shown in Figure 8 and Figure 9, generates dynamics problems based on the fact that movements on axis X or Y need both motors to work but movement in direction with $tg\ \alpha = \pm 1$ engages only one of the motors which has to move both the bridge and the head.

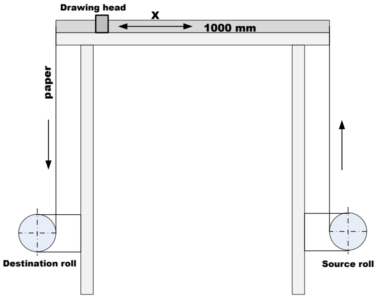


Figure 7. Large-scale textile plotter - side view

Simulation of such a dynamics is rather complicated. The weight of the object to be moved varies 20 times – it is only the head or both head and bridge. According to the angle of movement, every motor has a different load. The load varies all the time when the head has to draw some curve. Identification and modelling of that plotter are out of the scope of this paper. Here, we will discuss the simulator and its validation.

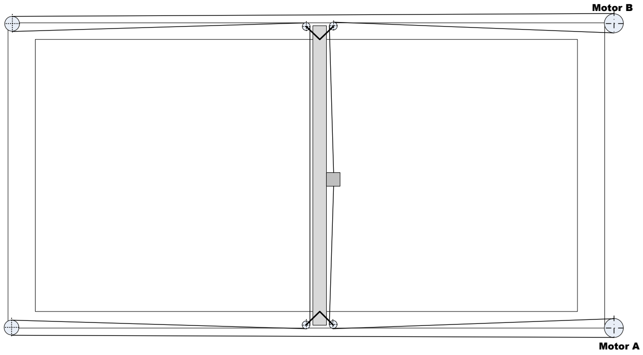


Figure 8. Kinematics scheme

ISBN: 978-1-61208-371-1

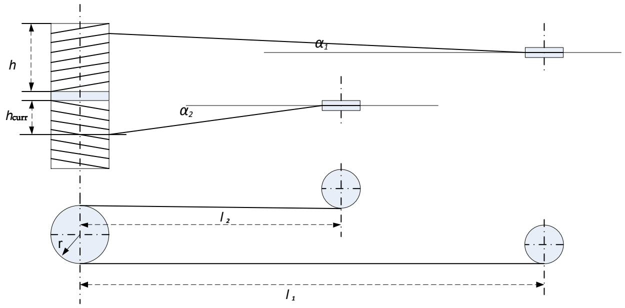


Figure 9. Wiring scheme

The simulation procedure is implemented in two different types of complexity. The first one is a simple computer simulation. The second is a parallel run of the simulator and the real machine. The possibility to run both types of simulation is the reason to avoid usage of MATLAB and SIMULINK for a real-time test-bed. The simulator is implemented on a 32-bit computer with a math co-processor running a real-time operating system and the real-time interpreter of the generated by the PrGen system. For testing purposes, it is equipped with a display having appropriate dimensions and resolution. The communication between the plotter controller and the simulator meets the requirements for Real-Time protocols. It is capable of transferring all the information for simulation and for the system log in a single control time interval. Visualisation and log modules of the simulator draw and save required and actual trajectories, internal parameters, etc.

Validation of the implemented simulator is done using model and data validation. Model validation is done off-line comparing real data and mathematical model outputs. After tuning, the mathematical model is implemented in the simulator. The simulator computational model is tuned again using the same approach. Real data are compared to the simulator outputs. Main differences are found in calculations accuracy. It follows from both calculation orders and numerical accuracy. After re-ordering calculation sequences in the simulator in the same way as they are done in MATLAB and increasing floating point accuracy both MATLAB and real simulator calculations become practically identical.

The implemented simulator was used for two different purposes: 1) to develop and implement new control algorithms to increase plotting accuracy and speed; 2) to analyse problems in drawn sketches, frame stitching, etc.

Using this simulator in simulated time (fast time) made possible to run long-lasting experiments (taking tens of minutes in normal speed) in minutes and thus enabled extensive experiments including full drawing of large sketches.

The simulator validity was proven using data validation on several levels. First, there was model and object output comparison implementing one and the same input sequences. Second, there was geometry accuracy validation. Drawn on paper and calculated by the simulator sizes were compared. All differences had the acceptable tolerance of ±0.25 mm, which is the initial requirement for mechanical accuracy.

V. Conclusion

This paper presented the validation of the program generation approach for design and implementation of object simulators and how the quality of the simulator can be increased using validation techniques. The presented results confirm that program generation is a promising way to implement object simulators with hardware and software structures representing the modelled object very closely. Once the program generator internal model is validated it becomes a useful tool for implementation of both simulators and control systems. The validation of the tool and generated by it systems are separated.

Validation of the implemented simulators follows both approaches – model validation and data validation. By implementing sequentially these validation techniques, the designed simulator can achieve required accuracy and functionality.

Using one and the same environment and tool for both controller and simulator building has the advantage to use only one tool for everything. This makes possible to distribute all changes in every building element immediately both in the controller and the simulator.

Acknowledgment

This work is funded partially by the Bulgarian NSF under DRNF02/3 project.

References

[1] S. Robinson, Simulation model verification and validation: increasing the users' reliability, Proceedings of the 1997 Winter Simulation Conference ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson.

[2] J.P.C. Kleijenen, Verification and Validation of simulation models, European Journal of Operational Research, vol. 82, 1995, pp. 145-162.

[3] G.W. Silvaggio, Emulated Digital Control System Validation in Nuclear Power Plant Training Simulators, http://scs.org/upload/documents/conferences/powerplantsim/2011/presentations/Session%207/Westinghouse/silvaggioFullPaper.pdf, [last accessed: 08.08.2014].

[4] Weilin Li, X. Zhang, and H. Li, Co-simulation platforms for co-design of networked control systems: An overview, Control Engineering Practice, vol. 23, 2014, pp. 44–56.

[5] O. Balci, 1997. "Verification, Validation and Accreditation of Simulation Models." In Proceedings of the 1997 Winter Simulation Conference, pp. 135-141.

[6] The Rapid Automotive Performance Simulator (RAPTOR), http://www.swri.org/4org/d03/vehsys/advveh/raptor/default.htm [last accessed: 08.08.2014].

[7] M. Pasquier, M. Duoba, and A. Rousseau, Validating Simulation Tools for Vehicle System Studies Using Advanced Control and Testing Procedure, http: //www.autonomie.net/docs/6 - papers/validation/validating_simulation_tools.pdf [last accessed: 08.08.2014].

[8] IAEA, Use of control room simulators for training of nuclear power plant personnel, Vienna, 2004, IAEA-TECDOC-1411, ISBN 92–0–110604–1.

[9] Johnstone M., D. Creighton, and S. Nahavandi, Enabling Industrial Scale Simulation / Emulation Models, In Proceedings of the 2007 Winter Simulation Conference, 2007, pp. 1028-1034.

[10] MathWorks, Generate and verify embedded code for prototyping or production, http://www.mathworks.com/embedded-code-generation/, [last accessed: 08.08.2014].

[11] F. G. Gonzalez, Real-Time Simulation and Control of Large Scale Distributed Discrete Event Systems, Conference on Systems Engineering Research, 2013, Procedia Computer Science 16 (2013), pp. 177 – 186.

[12] IEEE Standard Glossary of Software Engineering Terminology, New York, USA, 1990, ISBN 1-55937-067-X.

[13] N. Baldzhiev, V. Bodurski, and V. Gueorguiev, I. E. Ivanov, Implementation of Objects Simulators and Validators using Program Generation Approach, DESE 2011, Dubai, UAE, December 2011.

[14] C. K. Angelov and I. E. Ivanov, “Formal Specification of Distributed Computer Control Systems (DCCS). Specification of DCCS Subsystems and Subsystem Interactions”. Proc. of the International Conference “Automation & Informatics’2001”, May 30 - June 2, 2001, Sofia, Bulgaria, vol. 1, pp. 41-48.

[15] C. K. Angelov, I. E. Ivanov, K. L. Perev, and V. E. Georgiev. Design for Open and Predictable Automation Systems. Proc. of the 45th International Scientific Colloquium of the Technical University of Ilmenau, Oct. 2000, Ilmenau, Germany.

[16] I. E. Ivanov and V. Georgiev, “Formal models for system design”, Proc. of IEEE spring seminar 27th ISSE, Annual School Lectures, Bulgaria, 2004,vol. 24, pp. 564-568.

[17] D. Harel, “Statecharts: A visual formalism for complex systems” Science of Computer Programming 8 (1987) 231-274.

[18] I. E. Ivanov , “Control Programs Generation Based on Component Specifications”, PhD thesis, 2005, Sofia, (in Bulgarian).

[19] C. K. Angelov, I. E. Ivanov, and A. A. Bozhilov. Transparent Real-Time Communication in Distributed Computer Control Systems. Proc. of the International Conference “Automation & Informatics’2000”, Oct. 2000, Sofia, Bulgaria, vol. 1, pp. 1-4.

[20] A. Dimov, I. E. Ivanov, and K. Milenkov, "Component-based Approach for Distributed hard Real-time Systems", Information Technologies and Control, 2, 2005.

[21] A. Dimov and I. E. Ivanov, Towards development of adaptive embedded software systems, Proceedings of TU Sofia, vol. 62, book.1, 2012, pp.. 133-140.

[22] D. Knuth, The Art of Computer Programming I, Addison-Wesley,1968

[23] K. D. Tocher, The Art of Simulation, The English Universities Press Ltd., 1963

Fuzzy Discrete-Event Systems Modeling and Simulation with Fuzzy Control Language and DEVS Formalism

Jean-François Santucci and Laurent Capocchi

SPE UMR CNRS 6134 Laboratory
University of Corsica
Email: {santucci, capocchi}@univ-corse.fr

***Abstract*—There is an increasing use of fuzzy data in the field of discrete-event system modeling. This paper deals with an approach based on the use of Fuzzy Control Language (FCL) allowing to facilitate the modeling and simulation of Discrete EVent Systems (DEVS) involving uncertainty. The main contribution of this paper is to integrate the Fuzzy Control Logic three basic steps (fuzzification, defuzzification, fuzzy rules) into DEVS models in a generic way using a FCL-based implementation. The implementation has been performed with the DEVSimPy environment through a classical example. The DEVSimPy modeling of the example highlights how the DEVS modeler can realize the three basic steps inside the DEVS transition functions.**

Keywords–DEVS; Fuzzy systems; Modeling; Simulation; Discrete-event system.

I. Introduction

In the recent years, a set of work has concerned the introduction of fuzzy notions into DEVS concepts in order to propose incorporation of uncertainty into models [1]. Approaches [2]–[4] developed around the DEVS formalism have been proposed in order to apply fuzzy set theory [5] to the sets and functions defined on crips sets in DEVS formalism. Fuzzy modeling and simulation can be useful in the study of different kinds of systems for example in the case of the modeling of systems for which we have few observations of how and where the human being acts as a sensor or expert. The already developed approaches integrating fuzzy sets into DEVS bring nice formalisms allowing to deal with fuzzy events, fuzzy transition functions and even fuzzy time advance function. However, the modeling and simulation of concrete applications involving fuzzy notions based on human expertise are not easily implemented using the previous approaches. In order to set up this, the fuzzy logic provides methods closed to expert human labor. Fuzzy logic has already proved the ability to reason about imprecise or subjective data in wide range of fields, from finance to industrial control, through consumer electronics and weapons systems. Fuzzy logic is closed to reasoning and human language, as it is known to use vague concepts (such as “hot ”, “weak”, “strong enough”, etc.).

The fuzzy approach allows the computer modeling of vague notions (hotter, colder, etc.), commonly used by consumers or by plant operators. Fuzzy technology has the advantage of using explicit knowledge (the system works by applying rules) by highlighting three distinct steps: (i) initially the fuzzification process is studied in order to consider the inclusion of linguistic information. (ii) the second part is devoted to the identification of fuzzy rules mainly when observations on the behavior of the system are imprecise and uncertain. Fuzzy rules (such as “if X … then Y”) allow to express the knowledge concerning the problem to address; (iii) the last part concerns the defuzzification which is to convert the fuzzy domain to the digital domain, with a conversion preferences. The method of defuzzification is used to exploit the information encoded in the output fuzzy sets corresponding to expert knowledge on the output variable of the model.

The integration of fuzzy logic into the DEVS formalism involves three steps: (i) the definition of concepts allowing to deal with fuzzy logic in the framework of DEVS; (ii) the selection of a fuzzy logic programming language in order to accomplish the integration of DEVS and fuzzy logic; (iii) the implementation of the concepts into the DEVSimPy framework [6][7]. The first step allows to define the concepts allowing to perform with the DEVS formalism the three required steps involved in fuzzy logic control: fuzzication, the fuzzy rules definition and firing and finally defuzzification. Among a set of available fuzzy theory programming languages (FCL - Fuzzy Control Language [8], FPL-Fuzzy Programming Language [9], FTL - Fuzzy Technology Language [10], FSTDS-fuzzy STDS [11], LPL - Linguistic oriented Programming Language [10], FSML-Fuzzy System Modeling Language [12], etc.), we choose the FCL one for different reasons: it is a standardized Domain Specific Language created for fuzzy control applications; it allows to define blocks of fuzzy functions with fuzzy inputs and outputs; it allows also to simplify the writing of variables and linguistic values, as well as the writing of logical rules using syntax such as “IF, THEN”.

The main contribution of this paper is not concept-oriented as in Kwon et al. [4], but concerns the implementation of a Fuzzy Control Logic library (including fuzzification, defuzzification, fuzzy rules) into the DEVSimPy framework in a generic way using a FCL-based implementation. The DEVSimPy framework is a DEVS modeling and simulation tool written in Python language. Python is one of the programming languages that provides a framework for defining fuzzy inference systems through PyFuzzy package [13]. In addition, Python also allows the definition of these fuzzy inference systems from the FCL language. In order to facilitate for a DEVS modeler the use of fuzzy logic when performing simulation of a given application we choose to integrate FCL systems into the DEVSimPy framework. FCL facilitates the DEVS modeler to transform the imprecision in users request into defuzziyfied values which can be used easily in DEVS simulations. The rationale behind the coupling of FCL system with DEVS is that an “expert” human operator can control a process without understanding the details of its underlying dynamics. The effective and real control strategies that the expert learns through experience can often be expressed as a set of condition-action, “IF, THEN” rules, that describe the process state and recommend actions using linguistic, fuzzy terms instead of classical, crisp rules.

The rest of the paper is organized as follows: Section 2 presents the background of the work including the DEVS formalism and the FCL language. Section 3 gives the concepts that have been defined in order to integration fuzzy logic into the DEVS formalism. The implementation of the previously defined concepts is given is Section 4. A pedagogical example allows to illustrate the feasibility of the approach inside the DEVSimPy environment. Finally, in the last part, we conclude the work and present the future work around fuzzy inductive modeling.

II. Background

A. The DEVS Formalism

Since the seventies, some formal works have been directed in order to develop the theoretical basements for the modeling and simulation of dynamical discrete event systems [14]. DEVS [15] has been introduced as an abstract formalism for the modeling of discrete event systems, and allows a complete independence from the simulator using the notion of abstract simulator.

DEVS defines two kinds of models: *atomic models* and *coupled models*. An atomic model is a basic model with specifications for the dynamics of the model. It describes the behavior of a component, which is indivisible, in a timed state transition level. Coupled models tell how to couple several component models together to form a new model. This kind of model can be employed as a component in a larger coupled model, thus giving rise to the construction of complex models in a hierarchical fashion. As in general systems theory, a DEVS model contains a set of states and transition functions that are triggered by the simulator.

A DEVS atomic model AM with the behavior is represented by the following structure:

$$AM =< X, Y, S, \delta_{int}, \delta_{ext}, \lambda, t_a > \quad (1)$$

where:

- $X : \{(p, v)|(p \in inputports,\, v \in X_p^h)\}$ is the set of input ports and values,
- $Y : \{(p, v)|(p \in outputports,\, v \in Y_p^h)\}$ is the set of output ports and values,
- S: is the set of states,
- $\delta_{int} : S \rightarrow S$ is the internal transition function that will move the system to the next state after the time returned by the time advance function,
- $\delta_{ext} : Q \times X \rightarrow S$ is the external transition function that will schedule the states changes in reaction to an external input event,
- $\lambda : S \rightarrow Y$ is the output function that will generate external events just before the internal transition takes places,
- $t_a : S \rightarrow R^+_\infty$ is the time advance function, that will give the life time of the current state.

The dynamic interpretation is the following:

- $Q = \{(s, e)|s \in S^h, 0 < e < t_a(s)\}$ is the total state set,
- e is the elapsed time since last transition, and s the partial set of states for the duration of $t_a(s)$ if no external event occur,
- δ_{int} : the model being in a state s at t_i , it will go into s' , $s' = \delta_{int}(s)$, if no external events occurs before $t_i + t_a(s)$,
- δ_{ext} : when an external event occurs, the model being in the state s since the elapsed time e goes in s', The next state depends on the elapsed time in the present state. At every state change, e is reset to 0.
- λ : the output function is executed before an internal transition, before emitting an output event the model remains in a transient state.
- A state with an infinite life time is a passive state *(steady state),* else, it is an active state *(transient state)*. If the state s is passive, the model can evolve only with an input event occurrence.

The DEVS abstract simulator is derived directly from the model. A simulator is associated with each atomic model and a coordinator is associated with each coupled model. In this approach, simulators allows to control the behavior of each model, and coordinators allows the global synchronization between each of them.

B. Fuzzy Control Language

A Fuzzy Control Language (FCL) system allows to define the process for the mapping from a given input to an output using fuzzy logic. It is based on Mamdani [16] method for fuzzy inference which is the most suitable for capturing expert knowledge, as its rules allow us to describe the expertise in more intuitive and human-like manner. The fuzzy inference process comprises the following steps: fuzzify the input, evaluate the fuzzy rules, aggregate the outputs to reach the final decision, and defuzzify the output to obtain a crisp value.

A FCL system can be formalized as follows:

- V1: a set of input variables.
- V2: a set of output variables.
- Fuzzification part:
  <term (or set) name> := <points that make up the term> for each input variable.
- Rules description part:
  <operator>:<algorithm>;
  ACCUM:<accumulation method>;
  RULE <rule number>: IF <condition> THEN <conclusion>;
- Defuzzifiation part:
  METHOD:<defuzzification method>;

Each one of these parts has to be defined when considering a FCL system.

Fuzzyfication part: The values of V1 have to be converted into degrees of membership for the membership functions defined on linguistic variables. These linguistic variables shall be described by one or more linguistic terms. The linguistic terms are introduced and described by membership functions in order to fuzzify the variable.

$$membership\_function ::= (pointi), (pointj), \ldots \tag{2}$$

A membership function is defined by a table of points under the format presented in 2.

Rules description part: Two definitions are required in order to defined the rules: (i) The Fuzzy operators are used inside the rules; (ii) The accumulation method.

Fuzzy Operators: To fulfill de Morgans Law, the algorithms for operators AND and OR shall be used pair-wise e.g.:

- MAX shall be used for OR if MIN is used for AND.
- ASUM for Or if PROD for AND.
- BSUM for OR if BDIF for AND.

Accumulation method: Three methods are available:

- Maximum: MAX.
- Bounded Sum: BSUM.
- Normalized sum: NSUM.

The inputs of a set of rules are linguistic variables with a set of linguistic terms. Each term has a degree of membership assigned to it. It is possible to combine several sub-conditions and input variables in one rule. The general format is the following:

$$subcondition := linguistic\_variable\ IS\ [NOT] \\ linguistic\_term \tag{3}$$

The conclusion can be split into several subconclusions and output variables:

$$subconclusion := linguistic\_variable\ IS \\ linguistic\_term \tag{4}$$

Optionally it is possible to give each subconclusion a weighting factor which is a real with a value between 0.0 and 1.0. This shall be done by the keyword WITH followed by the weighting factor. The weighting factor shall reduce the membership degree (membership function) of the subconclusion by multiplication of the result in the subconclusion with the weighting factor. The generic format of a rule is:

$$IF\ condition\ THEN\ subconclusion \\ [WITH\ weighting\_factor]\ subconclusion \tag{5}$$

Defuzzification part: A linguistic variable for an output variable has to be converted into a value. The variable which is used for the defuzzification corresponds to an output variable.

A defuzzification method has to be chosen; The following defuzzification methods are possible:

- COG Centre of Gravity.
- COGS Centre of Gravity for Singletons.
- COA Centre of Area.
- LM Left Most Maximum.
- RM Right Most Maximum.

The following Section introduces the proposed approach based on the previous concepts.

III. Proposed Approach

The section introduces the concepts that have been defined in order to integrate FCL into DEVS models. In order to succeed in this integration, the following requirements have to be fulfill:

- Ability to define of membership functions (see Equ. 2) for creating fuzzy inference systems (V1, V2).
- Ability to support the AND, OR, and NOT logic operators in user-defined rules.
- Ability to embed a fuzzy inference system in a DEVS model.
- Ability to generate executable fuzzy inference engines.

The proposed solution to integrate FLC into DEVS models is to offer the possibility to define aggregate fuzzy logic to each one of the functions involved in the DEVS formalism (δ_{ext}, δ_{int}, λ and t_a). For that X, Y and S will contain variables which are going to be fuzzyfied. Then inferences rules (see Equ. 3,4,5) will be written inside of DEVS function and fired as soon as the corresponding function is activated. Finally, a defuzzyfication mis performed in order to obtain the result of the respective function (new state for δ_{ext} and δ_{int}, an output for λ and a real representing the σ for t_a). During the simulation, the execution of one of the traditional DEVS functions consists in firing the fuzzy engine associated with each DEVS function.

A DEVS atomic model AM integrating a fuzzy behavior using the FCL language and inference engine is represented by specifying the following functions:

- $\delta_{int} : S \to S$ is defined as a FCL system involving the three previously presented steps (fuzzyfication, rules definition, defuzzyfication). It is defined as a FCL system $\delta_{int}^{FCL}(V1 : S, V2 : S)$.
- $\delta_{ext} : Q \times X \to S$ is defined as a FCL system $\delta_{ext}^{FCL}(V1 : Q \times X, V2 : S)$
- $\lambda : S \to Y$ is defined as a FCL system $\lambda^{FCL}(V1 : S, V2 : Y)$
- $t_a : S \to R_{\infty}^{+}$ is defined as a FCL system $t_a^{FCL} : (V1 : S, V2 : real)$.

The proposed approach allows a modeler using the DEVS formalism to integrate fuzzy logic into the modeling scheme of the system under study just by specifying for one or more of the 4 atomic model functions (δ_{ext}, δ_{int}, λ or t_a): The fuzzification method of the input variables; Write the inference rules linked with the behavior of the system under study; Select the defuzzification method for the output variables. The classical DEVS modeling and simulation has been unchanged and it allows to perform the associated fuzzy inference engine. The next Section will point out how this approach has been implemented in the framework of the DEVSimPy environment.

IV. DEVSimPy Implementation

A. The DEVSimPy Framework

DEVSimPy [6] is an open Source project (under GPL V.3 license) supported by the SPE team of the University of Corsica "Pasquale Paoli". This aim is to provide a GUI for the modeling and simulation of PyDEVS [17] models. PyDEVS is an Application Programming Interface (API) allowing the implementation of the DEVS formalism in Python language. Python is known as an interpreted, very high-level, object-oriented programming language widely used to quickly implement algorithms without focusing on the code debugging [18]–[20]. The DEVSimPy environment has been developed in Python with the wxPython [21] graphical library without strong dependences other than the Scipy [22] and the Numpy [23] scientific python libraries. The basic idea behind DEVSimPy is to wrap the PyDEVS API with a GUI allowing significant simplification of handling PyDEVS models (like the coupling between models or their storage).

Figure 1 depicts the general interface of the DEVSimPy environment. A left panel (see bag 1 in Figure 1) shows the libraries of DEVSimPy models. The user can instantiate models by using a drag-and-drop functionality. The bag 2 in Figure 1 shows the modeling part based on a canvas with interconnection of instantiated models. This canvas is a diagram of atomic or coupled DEVS models waiting to be simulate.

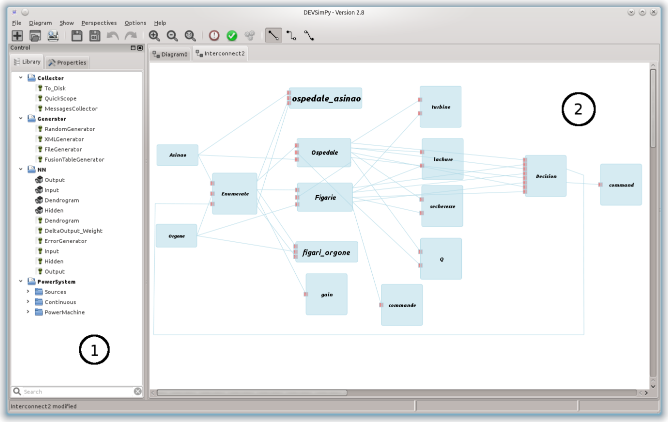


Figure 1. DEVSimPy general interface.

A DEVSimPy model can be stored locally in the hard disk or in cloud through the web in the form of a compressed file including the behavior and the graphical view of the model separately. The behavior of the model can be extended using specific plug-ins embedded in the DEVSimPy compressed file. This functionality is powerful since it makes it possible to implement new algorithms above the DEVS code of models in order to extend their handling in DEVSimPy (exploit behavioral attributes, overriding of DEVS methods, ...). A plug-in can also be global in order to manage several models through an generic interface embedded in DEVSimPy. In this case, the general plug-in can be enabled/disabled for a family of selected models.

B. Case of Study: A Burner tank

1) Description: We have validated the implementation using a classical pedagogical example issued from [15]. This example concerns a fuzzy modeling of a boiler system. The system consists of two subsystems, a burner and a boiler tank. The behavior of the boiler tank is specified as a deterministic one while the behavior of the burner system relies on fuzzy rules. Of course we are mainly interested in this paper into the burner fuzzy modeling and simulation. The burner model has two inputs ports: "on" port and "off" port. Events on "on" (resp. "off") port allow to turn on (resp. off) the burner. A set of specifications involving possibilities that a faulty behavior of the burner emerges has been defined in order to study some unusual or faulty operations. The specifications are expressed as follows (see Figure 2):

- When an input event "on" is applied to the model whose initial state is "OFF" , the possibility to remain in "OFF" is 0.3 and the possibility of state transition to state "IGN" is 0.8.
- When an input event "off" is applied to the model whose initial state is "OFF" , the possibility to remain in "OFF" is 0.9 and the possibility of state transition to state "EXT" is 0.3.
- When an input event "off" is applied to the model whose initial state is "ON" , the possibility to remain in "ON" is 0.2 and the possibility of state transition to state "EXT" is 0.9.
- When an input event "on" is applied to the model whose initial state is "ON" , the possibility to remain in "ON" is 0.7 and the possibility of state transition to state "IGN" is 0.5.
- When an input event "on" is applied to the model whose initial state is "IGN" , the possibility to remain in "IGN" is 0.3 and the possibility of state transition to state "ON" is 0.7.
- When an input event "off" is applied to the model whose initial state is "IGN" , the possibility of state transition to state "ON" is 0.2 and the possibility of state transition to state "OFF" is 0.8.
- When an input event "on" is applied to the model whose initial state is "EXT" , the possibility of state transition to state "ON" is 0.8 and the possibility of state transition to state "OFF" is 0.3.
- When an input event "off" is applied to the model whose initial state is "EXT" , the possibility to remain in "EXT" is 0.4 and the possibility of state transition to state "OFF" is 0.7.

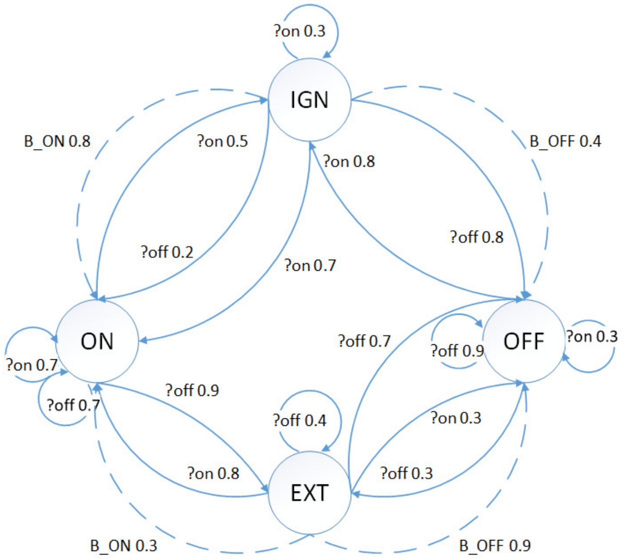


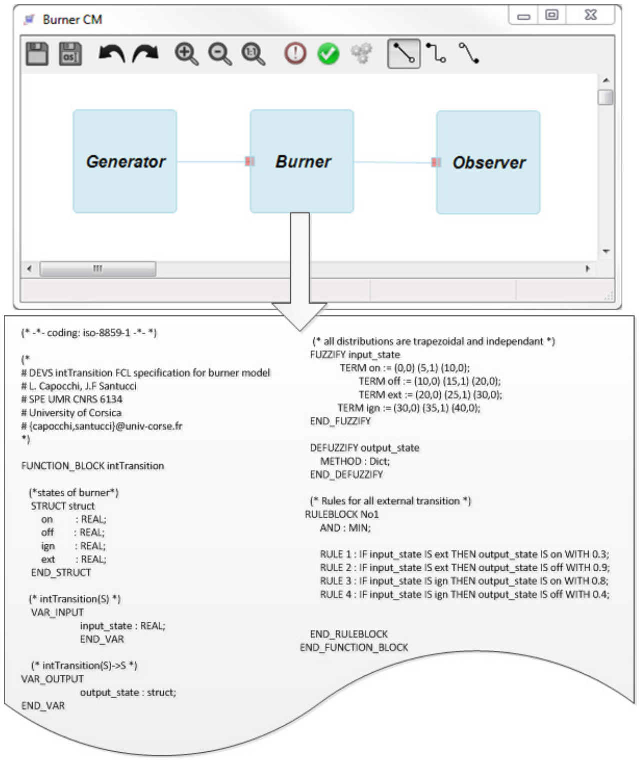
Figure 2. State automaton of the model *Burner*.

- In the state “EXT” ,there are two possible internal transitions: one is to state “ON” with possibility 0.3 and the other to the state “OFF” with possibility 0.9. The model generates an output event (“B_ON” or “B_OFF”) according to the transition.
- In the state “IGN” there are two possible internal transitions: one is to state “ON” with possibility 0.8 and the other to the state “OFF” with possibility 0.4. The model generates an output event (“B_ON” or “B_OFF”) according to the transition.

2) DEVSimPy Modeling: The DEVSimPy model is given in Figure 3. It is a coupled model involving three atomic models:

- An atomic model *Generator* allowing to generate the inputs events sent to the burner atomic model;
- The atomic model *Burner* which implements the fuzzy behavior of the system using three FCL systems; it has two input ports corresponding to the “on” and “off” ports and two outputs corresponding to the “B_ON” and “B_OFF” ports. The automaton of this model (see Fig. 2) is expressed through three set of rules included in three different files embedded in the model. Figure 3 depicts the file corresponding to the δ_{int} function.
- An atomic model *Observer* which allows to print the obtained solutions issued from the burner atomic model which implements a fuzzy behavior. Some of the obtained printed results are shown in Figure 4.

We have to highlight that, as it can be seen in Figure 3, the defuzzification method is called Dict. In fact it is not a true defuzzification method since the fuzzification does not deal with numerical value but instead a complex structure (a dictionary) involving the potential states. For this reason, no classical defuzzification methods (as COG, COGS, LM, RM, etc.) can be used. The user has to define manually after a function that according to the possibility values obtained after applying the “defuzzification” method Dict, what is the selected state. In regular cases, the user can give priority to the state value possibility being the greatest.



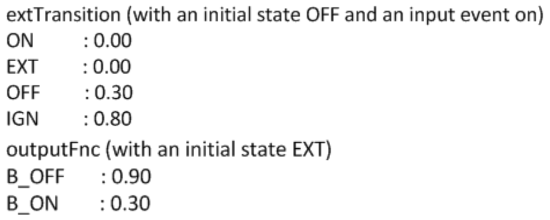
```
(* -*- coding: iso-8859-1 -*- *)

(*
# DEVS intTransition FCL specification for burner model
# L. Capocchi, J.F Santucci
# SPE UMR CNRS 6134
# University of Corsica
# {capocchi,santucci}@univ-corse.fr
*)

FUNCTION_BLOCK intTransition

  (*states of burner*)
   STRUCT struct
     on      : REAL;
     off     : REAL;
     ign     : REAL;
     ext     : REAL;
   END_STRUCT

  (* intTransition(S) *)
   VAR_INPUT
              input_state : REAL;
              END_VAR

   (* intTransition(S)->S *)
VAR_OUTPUT
              output_state : struct;
END_VAR
```

```
 (* all distributions are trapezoidal and independant *)
 FUZZIFY input_state
       TERM on := (0,0) (5,1) (10,0);
           TERM off := (10,0) (15,1) (20,0);
           TERM ext := (20,0) (25,1) (30,0);
       TERM ign := (30,0) (35,1) (40,0);
 END_FUZZIFY

 DEFUZZIFY output_state
   METHOD : Dict;
 END_DEFUZZIFY

 (* Rules for all external transition *)
RULEBLOCK No1
   AND : MIN;

   RULE 1 : IF input_state IS ext THEN output_state IS on WITH 0.3;
   RULE 2 : IF input_state IS ext THEN output_state IS off WITH 0.9;
   RULE 3 : IF input_state IS ign THEN output_state IS on WITH 0.8;
   RULE 4 : IF input_state IS ign THEN output_state IS off WITH 0.4;


 END_RULEBLOCK
END_FUNCTION_BLOCK
```

Figure 3. The DEVSimPy implementation of the *Burner* coupled model.

We can point out the three following main advantages of the proposed implementation. First, the use of the FCL language allows a DEVS modeler to access and manipulate both numerical and linguistic information under one common framework. The rules generated by the fuzzy inference are easy to extract and they are already in an understandable, human readable format. Second, the design of the fuzzy controller system is flexible because membership functions can be defined in a large variety of shapes. Third, a FCL system can be developed with the use of a straightforward one-pass build-up procedure. This will permit to avoid a tedious and time consuming activity, especially for the novice fuzzy modeler (as it is often the case for a DEVS modeler).

3) DEVSimPy Simulation: Some simulation results collected by the *Observer* model are presented in Figure 4. They highlight the interest in using fuzzy logic in the study of such a system. For example, the possibility that the burner subsystem can stay in the state “OFF” despite the fact that an input event should have changed the state of the burner sub-system is expressed in Figure 4. The results obtained by simulation of the model of Figure 3 with an input event “on” and an initial state of the burner system being “OFF” show that the possibility to be in the state “OFF” after the external transition execution due to the input event “on” is 0.3 while the possibility to change to state “IGN” is 0.8. In the same way when the state of the burner sub-system is “EXT” after an internal transition, an output event is generated by the burner sub-system: the possibility of a “B_OFF” event is 0.9 while the possibility of a “B_ON” event is 0.3.



```
extTransition (with an initial state OFF and an input event on)
ON      : 0.00
EXT     : 0.00
OFF     : 0.30
IGN     : 0.80
outputFnc (with an initial state EXT)
B_OFF   : 0.90
B_ON    : 0.30
```

Figure 4. Simulation results for the external transition function and the output function of the *Burner* model.

V. Conclusion and future work

This paper proposes an approach to integrate fuzzy logic control into the DEVSimPy framework using FCL language. This approach leans on the coupling of FCL systems with the classical DEVS formalism. The concepts of fuzzy logic control have been introduced in the field of DEVS through an uncomplicated integration scenario. Fuzzy logic controller systems are relatively simple to design in the DEVSimPy framework using FCL system specifications. Involved mathematical insight was not required and the process relied more on intuition and experience for a DEVS modeler. A pedagogical example has been introduced and implemented into DEVSimPy in order to validate the presented approach. This work can further be developed in the following areas: (1) The fuzzification of the time advance function has not yet been validated. We are currently working on this validation using a pedagogical example requiring the modeling of uncertainty of the time advance function. (2) We plan to combine fuzzy logic and neural networks in the framework of the DEVSimPy environment. By cooperating, the two technologies could combine their advantages in order to explain the results of neural networks and to simplify the development of fuzzy rules. (3) A last future orientation concerns fuzzy Inductive modeling [24]. The idea is to design models from observations of Input/Output Behavior. Fuzzy inductive modeling leans on the following four steps: (i) Discretization of quantitative information; (ii) Reasoning about discrete categories; (iii) Inferring consequences about categories; (iv) Interpolation between neighboring categories using fuzzy logic. The future work will also concern the application of the fuzzy modeling and simulation on real cases such as the prediction of the water consumption on the Corsica Island in order to plan the production of water using a water distribution network all over the country or the forecasting of rainfall on the island in order to predict flood or land drought.

References

[1] K. Saleem, “Fuzzy time control modeling of discrete event systems,” in Proc. of the World Congress on Engineering and Computer Science, International Association of Engineers (IAENG), 2008, pp. 683–688.

[2] M.-J. Son and T.-W. Kim, “Torpedo evasion simulation of underwater vehicle using fuzzy-logic-based tactical decision making in script tactics manager,” Expert Syst. Appl., vol. 39, no. 9, Jul. 2012, pp. 7995–8012.

[3] P. A. Bisgambiglia, L. Capocchi, P. Bisgambiglia, and S. Garredu, “Fuzzy inference models for discrete event systems,” in Fuzzy Systems (FUZZ), 2010 IEEE International Conference on, July 2010, pp. 1–8.

[4] Y. Kwon, H. Park, S. Jung, and T. Kim., “Fuzzy-devs formalism : Concepts, realization and application,” in Proc. of AIS, 1996, pp. 227–234.

[5] L. A. Zadeh, “Fuzzy sets,” Information and Control, vol. 8, 1965, pp. 338–353. [Online]. Available: http://www-bisc.cs.berkeley.edu/Zadeh-1965.pdf [accessed: August, 2014]

[6] L. Capocchi, J. F. Santucci, B. Poggi, and C. Nicolai, “DEVSimPy: A collaborative python software for modeling and simulation of devs systems,” in Proc. of 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, June, pp. 170–175.

[7] L. Capocchi. DEVSimPy software. [Online]. Available: http:// code.google.com/p/devsimpy/ [accessed: August, 2014]

[8] I. E. C. technical committee, “Industrial process measurement and control,” IEC 61131 - Programmable Controllers, Tech. Rep., 2000, part 7: Fuzzy control programming. IEC.

[9] J. Kelber, S. Triebel, K. Pahnke, and G. Scarbata, “Automatic generation of analogous fuzzy controller hardware using a module generator concept,” in Proc. of 2nd European congress on intelligent techniques and soft computing, 1994, p. 8 pages.

[10] G. Nhivekar, S. Nirmale, and R. Mudholkar, “A survey of fuzzy logic tools for fuzzy-based system design,” vol. ICRTITCS, no. 9, February 2013, pp. 25–28, published by Foundation of Computer Science, New York, USA.

[11] M. Umano, M. Mizumoto, and K. Tanaka, “FSTDS system: A fuzzy-set manipulation system.” Inf. Sci., vol. 14, no. 2, 1978, pp. 115–159.

[12] W. L. Fellinger, “Specification for a fuzzy systems modelling language.” Ph.D. dissertation, 1978, ph.D. Thesis, Oregon State Univ., Corvallis.

[13] R. Liebscher, “PyFuzzy python package,” http://pyfuzzy.sourceforge.net/ [accessed: August, 2014], 2013.

[14] B. P. Zeigler, “An introduction to set theory,” ACIMS Laboratory, University of Arizona, Tech. Rep., 2003. [Online]. Available: http://www.acims.arizona.edu/EDUCATION/ [accessed: August, 2014]

[15] B. P. Zeigler, H. Praehofer, and T. G. Kim, Theory of Modeling and Simulation, Second Edition. Academic Press, 2000.

[16] O. Cordón, “A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems,” Int. J. Approx. Reasoning, vol. 52, no. 6, Sep. 2011, pp. 894–913. [Online]. Available: http://dx.doi.org/10.1016/j.ijar.2011.03.004 [accessed: August, 2014]

[17] J. S. Bolduc and H. Vangheluwe, “The modelling and simulation package pythondevs for classical hierarchical devs,” McGill University, Tech. Rep., 2001, mSDL technical report MSDL-TR-2001-01.

[18] F. Pérez, B. E. Granger, and J. D. Hunter, “Python: An ecosystem for scientific computing,” Computing in Science and Engineering, vol. 13, no. 2, 2011, pp. 13–21.

[19] K. J. Millman and M. Aivazis, “Python for scientists and engineers,” Computing in Science and Engineering, vol. 13, no. 2, 2011, pp. 9–12.

[20] H. Langtangen, A primer on scientific programming with Python. Springer-Verlag New York Inc, 2011, vol. 6.

[21] N. Rappin and R. Dunn, WxPython in action. Greenwich, Conn: Manning, 2006.

[22] F. J. Blanco-Silva, Learning SciPy for Numerical and Scientific Computing. Packt Publishing, 2013.

[23] T. Oliphant, A Guide to NumPy. Trelgol Publishing, Spanish Fork, UT, 2006, vol. 1.

[24] J. Moreno-Garcia, J. Castro-Schez, and L. Jimenez, “A fuzzy inductive algorithm for modeling dynamical systems in a comprehensible way,” Fuzzy Systems, IEEE Transactions on, vol. 15, no. 4, Aug 2007, pp. 652–672.

Integrating Simulation Modelling and Value Stream Mapping for Leaner Capacity Planning of an Emergency Department

Esmat Swallmeh(*), Ayman Tobail(*), Waleed Abo-Hamad(*), James Gray(^), Amr Arisha(*)

(*)3S Group, College of Business, Dublin Institute of Technology (DIT), Dublin 2, Ireland
(^) Emergency Department, Tallaght Hospital, Tallaght, Dublin 24
Email: esmat.swallmeh@mydit.ie, ayman.tobail@dit.ie, waleed.abohamad@dit.ie, James.Gray@amnch.ie, amr.arisha@dit.ie

Abstract — **Recently, the application of lean thinking in healthcare has grown significantly in response to rising demand, caused by population growth, ageing and high expectations of service quality. However, insufficient justifications and lack of quantifiable evidence are the main obstacles to convince healthcare executives to adopt lean philosophies. This paper presents a real application of the successful implementation of a methodology that integrates Value Stream Mapping (VSM) and simulation modelling to improve an emergency department (ED) of a University hospital in Dublin. Appling lean approach in operations will minimize the patient waiting time and improve service time. VSM points out to the value-added and non-value-added activities in a clear schematic way. Simulation Model is also developed for the department in order to account for the variability and complexity resulted in healthcare processes due to dynamism and sharing resources. A comparative analysis of current and future state of the ED is provided and presented to managers to illustrate the potential benefits of adopting lean practices.**

Keywords: Lean; Modeling and Simulation; Healthcare; Capacity Planning.

I. Introduction

Health systems are complex [1]. Although a significant fraction of many governments' budgets is allocated to health, results have hardly matched expectations as many health system performance indicators have shown limited improvement. Long waiting lists, overcrowding, and patient's dissatisfaction are the main symptoms of the healthcare system in Ireland with over 46,400 adults and children waiting for hospital treatment, according to the latest figures from the Health Service Executive (HSE) in Ireland [2]. Healthcare managers are challenged by intrinsic uncertainty of the demands and outcomes of healthcare systems [3]; high public demand for increased quality [4]; high level of human involvement at both patients level and resource level (doctor, nurses, etc.); limited budget and resources; and large number of variables (e.g., staff scheduling, number of beds, etc.). As a result, service providers and healthcare managers are continuously studying the efficiency of existing healthcare systems and exploring improvement opportunities.

Traditionally, critical decisions are made based on the vast arrays of data contained in clinical and administrative records. This approach cannot succeed in representing the dynamic interaction between the interconnected components of the healthcare system. Consequently, this approach has limited use when it comes to predicting the outcomes of changes or proposed actions to the system. Analytical tools are needed to support managers' decisions at different levels within the healthcare system. If analytical models are impractical in the healthcare settings, it is usually due to the imposed simplifications on the model. Accordingly, important details and features of the underlined systems cannot be captured. Lean thinking and simulation modeling offer two distinct frameworks which healthcare organizations can adapt to address their challenges.

In the rest of the paper, Section 2 reviews related work. Section 3 introduces the case study and the application of lean and simulation, followed by experimentation and analysis in Section 4, while Section 5 concludes the paper.

II. Literature Review

With the healthcare organizations aim to improve the patient experience and system's effectiveness and efficiency, they were confronted with increasing costs, demand on services and expectations in service quality. Corresponding to these challenges, decision makers were forced to think of new flexible ways to reduce waste, improve process control and improve utilization of resources. Lean is one of the most valuable techniques that can be used to achieve these ambitions [5]. Lean is a concept related to philosophies derived from Toyota production system to create more value with less. The lean process evaluates operations step by step to identify waste and inefficiency and then creates solutions to improve operations and reduce cost [6]. The lean process represent an unending cycle toward perfection, in which the services in continually refined and improved [7]. The existing number of publications proves the importance of lean in healthcare [8]. Lean has been utilized effectively in healthcare to reduce waiting times and improve system performance by eliminating the wasteful activities from the existing processes [9], [10] in many departments across the hospital setting including the emergency [11], [12] and surgery departments [13]. Despite the efficiency of implementing lean philosophy, reports reflected on few drawbacks especially related to the interpretations of changes.

Reviewing the portfolio of techniques and tools that assist in implementing lean, three groups can be distinguished. While some of these tools such as process mapping are used to review the performance of the organization in terms of waste and process, improvement tools are focusing on redesigning and developing the existing processes. Finally, monitoring tools, such as visual management, are used to examine the existing and the developing processes [14]. Furthermore, while some of these tools are focusing on the entire organization as the unit of analysis, others such as Value Stream Mapping (VSM) are more focused on a product value stream. A value stream is a mapping tool that is used to identify Value-Added activities (VA) from Non-Value added (NVA) activities, the NVA activities increase throughput time and develop an undue service time [15]. In recent years, VSM has emerged as the preferred assist to implement lean [16] and in particularly with complex dynamic systems such as emergency department [17]. VSM is considered a valuable tool in helping health care managers to recognize waste and its sources. The content of VSM does not only include information about the materials flow but also about the flow time based performance [18]. VSM can be formulated in three steps; the first step is to formulate the current state value map to give a snapshot of the current processes. To achieve this, managers illustrate every step in service and document facts such as cycle time, buffer sizes, and personal requirements [19]. The second step consists in the identification and analysis of the waste encountered along the value stream. Finally, a parallel map is developed to describe the ideal future state without the removed waste [20] Critique to VSM approach is due to the fact that it is a static tool and sometimes unable to illustrate the dynamic behavior between system components. For managers, this might affect their judgments to recognize whether the best future state as regards to the desired level of system performance is achieved or not [21]. At the moment, there is a need to wait for few months to monitor the impact of the changes on the system key performance indicators. Consequently, a complementary tool with VSM is required to handle uncertainty and model the dynamics between system components for different future state maps [22]. Using simulation modeling in the assessment process of lean implementation would explore the various opportunities of process improvement and the impact of the proposed changes before implementation [23]. The future state has to be validated before implementation in order to minimize risks and reduce trial and error adjustments. Simulation can also be used to systemically identify the best alternatives of the future state. By using simulation, operation process data can be analyzed fully to examine the parameter’s variability and uncertainty in operation [24] and [25]. Simulation has been widely proven to be a flexible tool to model the uncertainty and complexity [26]. By testing design alternatives, simulation analysis will provide an effective way of examining solutions prior to their actual implementation and reducing the risk associated with lean process.

III. Simulation-Based Value Stream Mapping

A. *Hospital and ED Background*

In Ireland, with more than 1.2 millions attending the ED every year, the ED overcrowding has been declared a national emergency. The shortage of beds coupled with the high patient demand and shortage of staff has affected the ED capability to provide efficient and safe care to patients. Prolonged waiting times has been reported with more than 40% of the patients waiting between 10-24 hours (HSE Performance Monitoring Report, 2010). Consequently, the hospitals are not compliant with the Health Service Executive (HSE) waiting time targets (6 hour patient experience time target). It was also reported by the task force in 2007 that the overall ED physical space and infrastructure is insufficient. Accordingly, analysing the patient flow in emergency departments to optimise the ED capacity in order to minimize the length of stay has become a crucial requirement. The hospital studied is a public, adult teaching hospital that holds 463 beds and handles almost 220,000 patients annually. Its ED logs over 45000 patients a year. The ED has 4 resuscitation beds, 4 minor injury beds, 9 major beds, 1 triage room, and 2 X-ray rooms. It has also a Clinical Decision Unit (CDU), which is designated for patients who are still under ED physicians and need further investigations. The uncertainty and complexity of the ED processes require a technique that can handle system variation and validate improvements steps before it can be implemented [27]. The proposed framework is used to identify VA and NVA activities and model the complexity of the ED to explore potential solutions to meet the national metrics.

B. *Data Collection and Analysis*

In order to acquire knowledge about the ED processes and build a VSM, extensive information was gathered. Direct observation to the service delivery operation was conducted to represent the interaction between the different service delivery processes and to deliver average processes times. Extensive interviews with ED doctors, nurses, administrators, nursing staff, and unit porters were conducted to gain precious insights into the ED operation processes in theory and practice. The vast amount of data collected was very valuable in determining the current ED service delivery processes and developing the VSM. ED service delivery process have the following sets of activities, some occur in sequential; some are omitted for different patient conditions. These activities are: patient arrival; triage process; initial ED doctor assessment; diagnostic tests; doctor follow up or treatment planning; admission or discharge.

There are two main ways for patient’s arrival to EDs, ambulance or walk-in. Patients arriving via ambulance are always giving priority as they are considered as urgent and need to be triaged immediately. By using the Manchester triage system, patients are categorized into 5 groups and each group is given a color: Blue, Green, Yellow, Orange, and Red. The red category signals a very sick patient that needs to be seen immediately and blue signals the non-urgent patient. For the purpose of this paper, the patient flow was given number from 1-15. The Patient arrives and is registered by a receptionist. In the next step, the patient will be triaged by a senior nurse to determine the severity of their condition and the patient will be labeled by a triage category. If the patient is given the red or orange category, the patient will be moved to resuscitation or to major immediately. The triaged patients (yellow, green, and blue category) will wait in the designated area (waiting room) for an available bed. The patients will be seen according to their triage category. If the ED cubicles are full or the doctors are busy with life threatening case, the patients have to wait longer to be seen, resulting in overcrowding. Hence, the arrival of critically ill patients will be given priority to be seen ahead of other patients. The patients will be called by a nurse and allocated a cubicle waiting to be assessed by ED doctor. Depending on the initial ED doctor assessment, one or more diagnostic tests are recommended. Following the diagnostic tests, the patient will be reviewed by ED doctor to recommend admitting them to the hospital or discharged from ED or further investigation required. If the patient requires a medical or surgical assessment, the bleep system will be used to contact the designated team and a verbal handover will be given by ED doctor.

If the surgical team is carrying out an operation in Theatre, the patient has to wait for few hours before an assessment will be carried out. If the patient needs further investigation under ED doctor care, the patient will be admitted to CDU. During the patient stay in CDU, the medical care is provided by ED health care staff. If the diagnostic test comeback as abnormal, the patient will be referred to the appropriate medical/surgical team and will be admitted to the hospital, otherwise the patient will be discharged. If a decision is taken to admit the patient, the hospital bed manager has to be notified to allocate a bed. During the wait time for ward bed, the ED nurses are responsible for the patient's care.

C. Development of the VSM

Data for the current stat map was collected as recommended by Rother and Shock [28]. The data collection for the ED process started at patient leaving ED (admission or discharge), and worked backward all the way to the patient arrivals, gathering snapshots of data such as process cycle time, waiting time before each process, number of employees for each process and their qualification. The VSM for ED is presented in Figure 1.

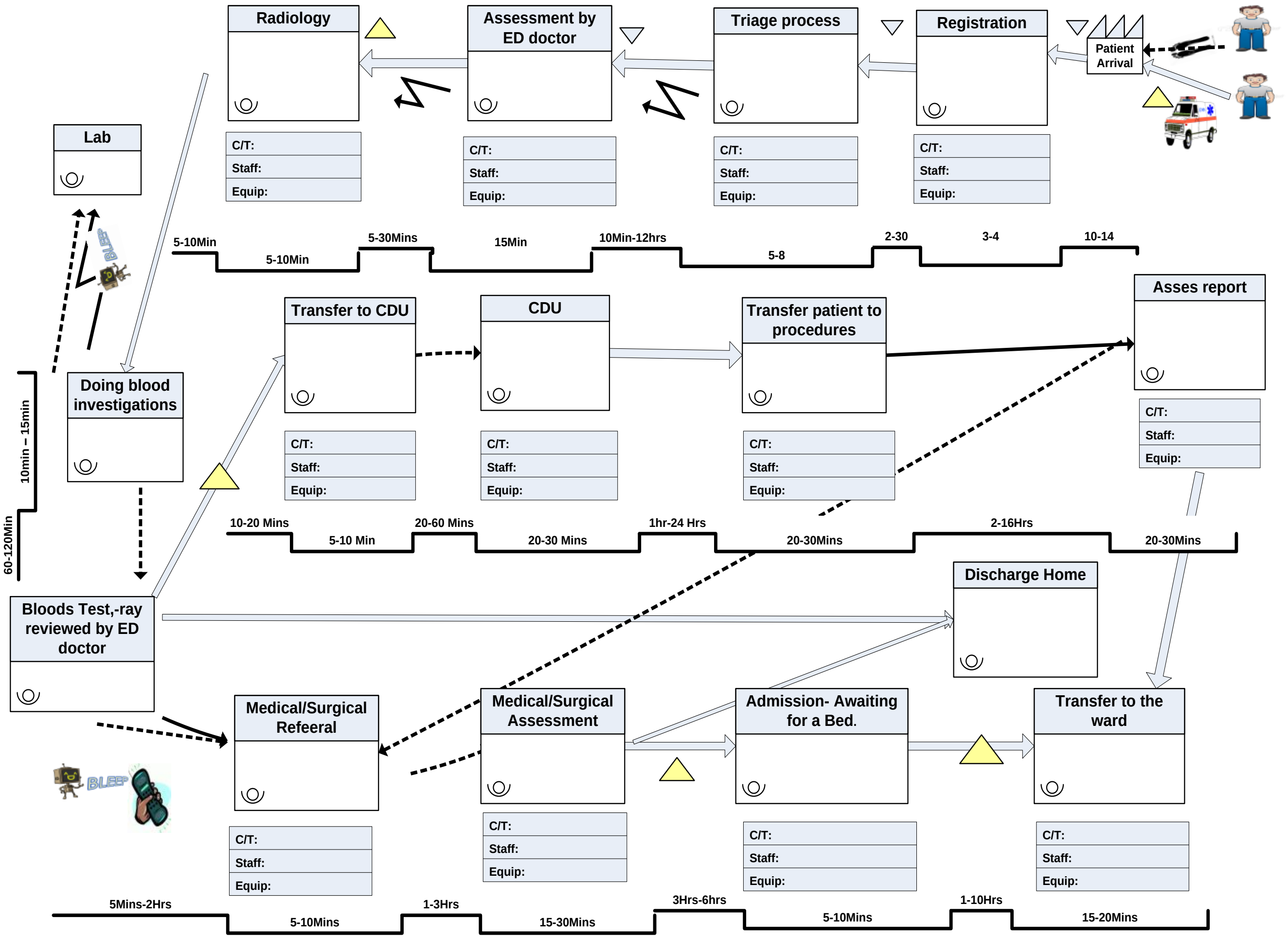


Figure 1. Current State Value Stream Map of the ED

The large boxes in the map represent the process and steps needed to achieve it. Also, each process has a data box below, which contains the health care staff in each process, and the facility required. The timeline at the bottom has two elements. The first element of the timeline is the processing time (VA time). VA time is obtained by summing all the processing times in the value stream. The second component of timeline is the process waiting time (NVA). This time is calculated by adding the waiting time before each process.

As represented in the VSM, the average waiting time to see a doctor is between 10 to 420 minutes depending on the patient's condition and the availability of a bed in ED. Despite the process of doing blood investigation average time is between 10-15 minutes, waiting for the results to be ready for review is between 60-120 minutes. There are three possible paths for patients in ED after the ED doctor review the diagnostic test results and commence follow up or begin treatment of the patient. These paths are:

1. The patient need further diagnostic tests but still under ED physician care. The average process time from patient arrival to ED to be ready to be discharged or admitted to the ward is between 103-157 minutes compared to 182-2154 minutes average waiting time.

2. Patient treatment is finished and is to be discharged. The average process time from arrival to ED to be discharged is 48-77 minutes while the waiting average time is between 102-654 minutes.

3. Finally, if the patient required a medical/surgical consultation, the average process time is between 58-97 minutes while the average waiting time is between157-934 minutes.

To find the causes behind the long NVA time in the ED process, conducting a root cause analysis is essential.

D. Root Cause Analysis

Root cause analysis is an applied tool developed by Professor Ishikaua in 1943 [29] to analyze possible causes or problems while organizing the causal relationships. A root cause analysis was conducted to explore the causes behind the long waiting time (NVA) in ED process. Figure 2 represents a root cause analysis diagram. According to the diagram, six categories were found to be the main contributors in extending waiting time for patients in the ED journey. These categories were:

1. Resources:
   a. The shortage of beds in ED and hospital resulting from the high patients demand has impacted negatively on ED performance. Admitted patients have to wait a long time in ED before transferring them to their wards is not only blocking ED beds, but is also over utilized ED resources.
   b. Because of the lack of ED doctor's access to outpatient clinic, chronic patients were found to occupy acute beds in ED.

2. Information Technology (IT):

   There was no IT system found in order to inform ED staff members that blood results or X-ray are ready for collection. The waiting time for diagnostic results when they are ready to be reviewed is NVA time.

3. Manpower:
   a. There was no dedicated nurse in minor injury. The nurse in minor injury was distracted to help in resuscitation if needed. This causes longer waiting time to patients attending minor injury area.
   b. When the nurses on duty were not skilled to take bloods, the ED doctors have to do it themselves.
   c. If the senior doctor was busy in resuscitation, there will be no decision maker available to discharge or admit patients. Junior doctors have to wait to discuss patient case with senior doctor to direct the patient treatment plan.

4. Investigations:

   The limited access to diagnostic tests, such as CT scan at night and weekends, increases the patient's length of stay in ED dramatically.

5. Communication
   a. If a patient triaged and labeled with red or orange, triage nurse have to search for the nurse manager to allocate a bed immediately. No communication system in place available between triage nurse and nurse manager.
   b. Medical/surgical teams were not available all the time to respond to consultation referral from ED.
   c. Laboratory technicians were not informed about sending blood samples to the lab.

6. Process:
   a. Over processing in triage process. Because of the long waiting times to see a doctor, the tri-age nurse has to carry out a comprehensive over processing triage.
   b. Delay in transporting blood samples to the lab.

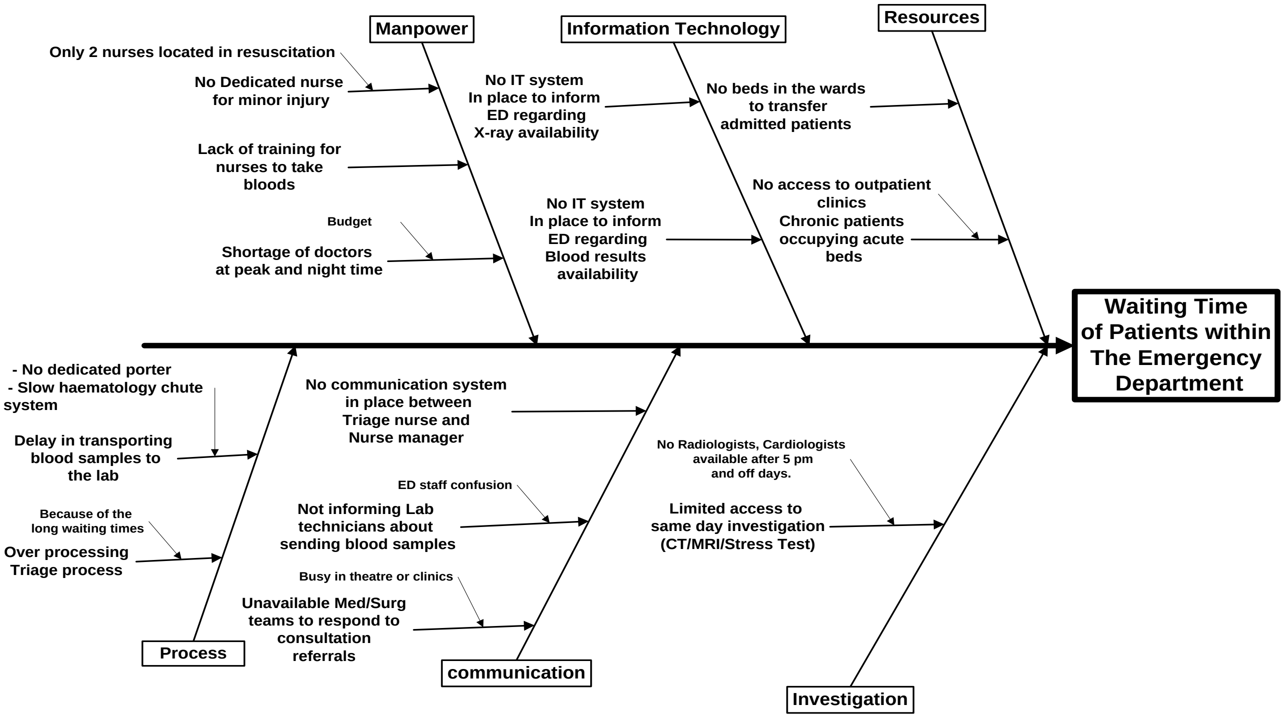

Manpower
Information Technology
Resources
Only 2 nurses located in resuscitation
No Dedicated nurse for minor injury
Lack of training for nurses to take bloods
Budget
Shortage of doctors at peak and night time
No IT system In place to inform ED regarding X-ray availability
No IT system In place to inform ED regarding Blood results availability
No beds in the wards to transfer admitted patients
No access to outpatient clinics Chronic patients occupying acute beds
Waiting Time of Patients within The Emergency Department
- No dedicated porter
- Slow haematology chute system
Delay in transporting blood samples to the lab
Because of the long waiting times
Over processing Triage process
Process
No communication system in place between Triage nurse and Nurse manager
ED staff confusion
Not informing Lab technicians about sending blood samples
Busy in theatre or clinics
Unavailable Med/Surg teams to respond to consultation referrals
communication
No Radiologists, Cardiologists available after 5 pm and off days.
Limited access to same day investigation (CT/MRI/Stress Test)
Investigation

Figure 2. Root cause analysis

E. *Simulation Model Development*

Based on the discussed patient flow, VSM, and root case analysis, a comprehensive simulation model was developed. As the work involves processes at the patient level, Discrete-Event Simulation (DES) is suitable. The simulation model has been designed to reflect the interaction between entities in the ED real world. Starting from the early stage of the development process, it took into consideration the whole model into the hierarchy scheme to enable the modularity and extendibility. Mainly, the simulation model is divided into three basic stages; patient generation and arrival, patient treatment and patient discharge. Generation of the patient arrival pattern in the first stage depended on the distributions extracted from the real data of the ED. Patients arrival has two sources; patients come with the ambulance and patients come as walk in. There are two entry points to the ED zones one for the ambulance patients who has higher priority and the other one for the walk in patients. The treatment stage of the model contains the main zones and labs of the ED. This stage starts with the triage block, which decides the priority level of treating patient on scale from 1 to 5 with 1 determining the highest priority. Patients move from triage to the zones or the waiting room according to the severity level. There are three zones for patient treatment, the first one zone1 is dedicated to the most severe cases such as unconscious or non-breathing patients, the second one is zone2 which receives patients with major injuries and the third one is zone3, which is dedicated to the minor injuries patients. Each zone provides a sequence of treatment and examining processes according to the severity level of the patient.

The routers blocks in the simulation model control the pathway of the patient treatment process according to the statistics extracted from the ED real data. Different resources batches to the patient by joining the process. When the patient enters the zone, a trolley and cubicle resource is assigned to the patient. In the treatment process, a doctor or nurse assigned to the patient. It depends on the capacity and availability these resources.

The journey in each zones ends with the discharging stage, which distribute the patient to one of the discharge outcomes; Home, GP, Die, Transfer and Others. During the whole journey of the patient and at certain points, the average waiting time and patient experience time (PET) - length of stay - are accumulated for each patient to calculate the final average PET and waiting time for all patients. These measurements or KPIs are used later to examine the impact of the change of the simulation variables on the performance of the ED.

For the simulation model to be considered as representative of the processes within ED and to be able to use it for testing scenarios, validation and verification are needed. Using visual tracking, the simulation model was verified by examining every group of blocks to ensure that patients follow the correct care path. Also, the conceptual model was validated by the ED senior managers and senior staff. A face validation approach was performed with the ED senior managers to validate the generic simulation model.

Finally, the output of the simulation model was compared with actual ED processes time and the deviation average was not more than 5% from the actual values.

IV. Experimentation And Analysis

Simulation model scenarios are in relation to the ED capacity in terms of number of major and minor cubicles required to better manage current and future demand for care. Besides ED capacity concerns, three other bottlenecks have been identified; time to admit patients from ED to hospital wards, ED staffing level, and time for specialty team to see a patient. A summary of the simulation variables and their agreed range of values is given in Table I.

Table I Simulation Variables

Simulation Variable		**Possible rang of values**
ED Size	Major Cubicles	9, 15, 20, 25
	Minor Cubicles	6, 8, 10, 12
	Resus Cubicles	4, 6
ED to Wards time (hrs)		1, 2, 3, 4
ED Staffing level		Current, infinity
Specialty Team Time (hrs)		2, 3

The main Key Performance Indicator used in the simulation model is the percentage of patients with patient experience time (i.e., length of stay) less than 6 hours. Due to the large number of combinations of variables, a total of 45 scenarios have been selected

Regarding the staffing level, the value ‘c’ refers to the current staffing level where the value ‘i’ represents an infinite supply of staff. Although not realistic, the objective of the assumption of infinite supply of staff is to switch on and off the staffing bottleneck and to focus on the determining the ED size regardless of the actual staff needed as requested by the ED stakeholders. Each scenario was tested by the simulation model for 12 months for 15 runs.

As shown in Table II, percentage of patients staying the ED less than 6 hours does not improve with only increasing the ED size.

TABLE II SIMULATION RESULTS OF INCREASING CAPACITY

Speciality Time (HRS)	**Staffing Levels**	**ED to Wards**	**Major**	**Minor**	**Resus**	**PET<6hrs (%)**
3	c	4	25	12	4	36
3	c	4	20	10	6	40
3	c	4	15	8	6	45

Along with current staffing level, decreasing the waiting time for patients who need to be admitted to hospital wards is crucial for improving PET. Additionally, decreasing the time for specialty teams to check patients in the ED is important for decreasing overall PET (Table III).

With resolving these two bottlenecks, namely ED toward time and specialty team time, the effect of increasing the ED size on PET is constrained by the current staffing level. Assuming an infinite supply of staff, it was found that the ED needs to increase its major cubicles by at least 122% of its current capacity (from 9 to 20) and minor cubicles by 33% with no need to increase Resus cubicles.

TABLE III SIMULATION RESULTS FOR ALL VARIABLES

Speciality Time	Staffing Levels	ED to Wards	Majors	Minors	Resus	PET < 6 hrs (%)
3	c	4	25	12	4	36
3	c	3	25	10	6	39
3	c	4	20	10	6	40
3	c	3	20	12	4	43
3	c	2	25	8	4	44
3	c	2	20	6	6	45
3	c	4	15	8	6	45
3	c	3	15	6	4	49
3	c	1	25	6	4	49
3	i	4	15	8	6	50
3	i	4	20	10	6	52
3	c	1	20	8	4	52
3	c	2	15	12	6	52
3	i	4	25	12	4	54
3	c	1	15	10	4	56
3	i	3	15	6	4	56
3	c	3	9	8	6	57
2	i	4	15	8	6	57
3	i	3	20	12	4	57
2	i	4	20	10	6	58
3	c	2	9	10	4	59
2	i	4	25	12	4	59
3	i	3	25	10	6	60
3	c	1	9	12	6	61
3	i	3	9	8	6	62
3	i	2	15	12	6	62
3	i	2	20	6	6	63
2	i	3	9	8	6	63
3	i	1	15	10	4	64
3	i	2	9	10	4	65
3	i	2	25	8	4	65
2	i	3	15	6	4	66
3	i	1	20	8	4	67
3	i	1	9	12	6	68
2	i	2	9	10	4	68
2	i	3	20	12	4	68
3	i	1	25	6	4	70
2	i	3	25	10	6	71
2	i	2	15	12	6	71
2	i	1	9	12	6	72
2	i	2	20	6	6	75
2	i	1	15	10	4	76
2	i	2	25	8	4	77
2	i	1	20	8	4	82
2	i	1	25	6	4	84

V. Conclusion

Integrating lean thinking into decisions is a well-thought management practice in complex business systems. It can help not only to improve the quality of service but it also eliminates non-value added activities. This approach can be applied to support healthcare managers to reduce costs and improve healthcare organization performance. To exploit the improvement opportunities, VSM was used to identify the non-value added activities in a structured diagram that presents system parameters (i.e. materials flow and the cycle time). This enables the management team to discover system bottlenecks and impact of changes on performance. Application of Simulation modelling was also utilised in the case study to demonstrate the dynamism of the key system resources. Results show that the integration between VSM and Simulation Modelling was effective in modelling capacity planning activities within the Emergency Department of one of the leading hospitals in Dublin. The framework has a potential to be used as a template of planning emergency departments in Ireland.

References

[1] D. Balabanova, A. Mills, L. Conteh, B. Akkazieva, H. Banteyerga, U. Dash, L. Gilson, A. Harmer, A. Ibraimova, and Z. Islam, "Good health at low cost 25 years on: lessons for the future of health systems strengthening," The Lancet, vol. 381, pp. 2118-2133.

[2] HSE, "Health service executive emergency departments - patient profiles, experiences and perceptions," 2010. (Online). http://www.hse.ie,. (retrieved: July, 2014).

[3] K. Katsaliaki and N. Mustafee, "Applications of simulation within the healthcare context," Journal of the Operational Research Society, vol. 62, 2011, pp. 1431-1451.

[4] S. Al-Balushi, A. Sohal, P. J. Singh, A. Al Hajri, Y. Al Farsi, and R. Al Abri, "Readiness Factors for Lean Implementation in Healthcare Settings–A Literature Review," Journal of Health Organization and Management, vol. 28, 2014, pp. 135-153.

[5] C. Koelling, D. Eitel, S. Mahapatra, K. Messner, and L. Grove, "Value stream mapping the emergency department," 2005. (online). htp:/w.ientorg/upladefis/SHReourc_Libay/Detils180.pdf. (retrieved: July, 2014).

[6] E. W. Dickson, S. Singh, D. S. Cheung, C. C. Wyatt, and A. S. Nugent, "Application of lean manufacturing techniques in the emergency department," The Journal of emergency medicine, vol. 37, 2009, pp. 177-182.

[7] E. M. Wojtys, L. Schley, K. A. Overgaard, and J. Agbabian, "Applying lean techniques to improve the patient scheduling process," Journal for Healthcare Quality, vol. 31, 2009, pp. 10-16.

[8] R. L. Wijewardana and T. Rupasinghe, "Applicability of Lean Healthcare in Sri Lankan Healthcare Supply Chains," International Journal of Supply Chain Management, vol. 2, 2013, pp. 42-49.

[9] T. Papadopoulos, "Continuous improvement and dynamic actor associations: A study of lean thinking implementation in the UK National Health Service," Leadership in Health Services, vol. 24, 2011, pp. 207-227.

[10] A. Virtue, T. Chaussalet, and J. Kelly, "Healthcare planning and its potential role increasing operational efficiency in the health sector: A viewpoint," Journal of Enterprise Information Management, vol. 26, 2013, pp. 8-20.

[11] M. J. Vermeulen, T. A. Stukel, A. Guttmann, B. H. Rowe, M. Zwarenstein, B. Golden, A. Nigam, G. Anderson, R. S. Bell, and M. J. Schull, "Evaluation of an Emergency Department Lean Process Improvement Program to reduce length of stay," Annals of emergency medicine, 2014, pp. 1-12.

[12] H. Chan, S. Lo, L. Lee, W. Lo, W. Yu, Y. Wu, S. Ho, R. Yeung, and J. Chan, "Lean techniques for the improvement of patients' flow in emergency department," World, vol. 5, 2014, pp. 24-28.

[13] R. M. Collar, A. G. Shuman, S. Feiner, A. K. McGonegal, N. Heidel, M. Duck, S. A. McLean, J. E. Billi, D. W. Healy, and C. R. Bradford, "Lean management in academic surgery," Journal of the American College of Surgeons, vol. 214, 2012, pp. 928-936.

[14] S. Robinson, Z. J. Radnor, N. Burgess, and C. Worthington, "SimLean: Utilising simulation in the implementation of lean in healthcare," European Journal of Operational Research, vol. 219, 2012, pp. 188-197.

[15] D. Hadfield, S. Holmes, S. Kozlowski, and T. Sperl, The New Lean Healthcare Pocket Guide: Tools for the Elimination of Waste in Hospitals, Clinics, and Other Healthcare Facilities: MCS Media, Incorporated, 2009.

[16] Y. H. Lian and H. Van Landeghem, "Analysing the effects of Lean manufacturing using a value stream mapping-based simulation generator," International Journal of Production Research, vol. 45, 2007, pp. 3037-3058.

[17] D. Cookson, C. Read, and M. Cooke, "Improving the quality of Emergency Department care by removing waste using Lean Value Stream mapping," The international journal of clinical leadership, vol. 17, 2011, pp. 25-30.

[18] A. Mahfouz, J. Crowe, and A. Arisha, "Integrating Current State and Future State Value Stream Mapping with Discrete Event Simulation: A Lean Distribution Case Study," in SIMUL 2011, The Third International Conference on Advances in System Simulation, 2011, pp. 161-168.

[19] P. Solding and P. Gullander, "Concepts for simulation based value stream mapping," in Procedding of the 2009 Winter Simulation Conference M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds., 2009, pp. 2231-2237.

[20] A. J. Donatelli and G. A. Harris, "Combining value stream mapping and discrete event simulation," in Proceedings of the Huntsville Simulation Conference,[online, 21.02. 2012]: https://rsic. redstone. army. mil/hsc/papers/hsc003. pdf, 2001. (retrieved: July, 2014).

[22] S. Sigari and R. Clark, "Applying lean thinking to improve the production process of a traditional medium–size British manufacturing company," International Journal of Information and Operations Management Education, vol. 5, 2013, pp. 154-169.

[23] C. Chen, J. Snowdon, and J. Charnes, "The current and future status of simulation software," in Proceedings of the 2002 Winter Simulation Conference E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds., 2002, pp. 1633-1640.

[24] A. Arisha and P. Young, "Intelligent simulation-based lot scheduling of photolithography toolsets in a wafer fabrication facility," In Proceedings of the 2004 Winter Simulation Conference R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds., 2004, pp. 1935-1942.

[25] A. Mahfouz, S. A. Hassan, and A. Arisha, "Practical simulation application: Evaluation of process control parameters in Twisted-Pair Cables manufacturing system," Simulation Modelling Practice and Theory, vol. 18, 2010, pp. 471-482.

[26] K. Ismail, W. Abo-Hamad, and A. Arisha, "Integrating balanced scorecard and simulation modeling to improve Emergency Department performance in Irish hospitals," In Proceedings of the 2010 Winter Simulation Conference B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds., 2010, pp. 2340-2351.

[27] A. Tobail, P. Egan, W. Abo-Hamad, and A. Arisha, "Application of Lean Thinking Using Simulation Modeling in A Private Hospital," in SIMUL 2013, The Fifth International Conference on Advances in System Simulation, 2013, pp. 22-28.M.

[28] Rother, Shock, J, Learning to See: Value Stream Mapping to Add Value and Eliminate Muda: The Lean Enterprise Institute, Inc., Brookline, May, 1999.

[29] B. Anderson, and T. Fagerhaug, Root cause analysis:Simplified tools and techniques: Milwaukee: ASQ Quality Press, 2000.

Model-based Method to achieve EMC for Distributed Safety-Relevant Automotive Systems

Andreas Baumgart*, Klaus Hörmaier†, Gerhard Deuter‡
*Carl von Ossietzky Universität Oldenburg
Email: andreas.baumgart@uni-oldenburg.de
†Infineon Technologies Austria AG
Email: klaus.hoermaier@infineon.com
‡TWT GmbH - Science & Innovation
Email: gerhard.deuter@twt-gmbh.de

***Abstract*—Automobiles contain an increasing number of distributed Electrical and Electronic systems. Among such systems, Electromagnetic Compatibility (EMC) is a major issue that impacts functional safety and requires a high verification effort. The automotive standard for functional safety ISO 26262 already provides requirements for EMC and refers to agreed standards. It addresses the topic during the development process in order to obtain a tolerable level of risks arising from possible causes for system malfunctions. Yet, a coherent process for EMC and functional safety on a concept level is missing. With this publication a model-based method is provided that connects specification, design, and analysis of functional as well as physical characteristics for distributed safety-relevant systems with EMC simulations. By applying our method, potential coupling paths and EMC issues can be systematically identified, analyzed and addressed by introducing counter measures at a concept level of the development process. The resulting technical design is compliant to a technical safety concept and traceable with respect to electromagnetic characteristics of affected hardware components and wires.**

***Keywords*–Electromagnetic compatibility (EMC); Model-based design; Automotive; ISO 26262; Safety-relevant systems**

I. Introduction

Today's cars contain an increasing amount and variety of Electrical and Electronic (E/E) systems. Thereby, many of them implement safety-relevant functions. The ISO 26262 is the agreed automotive standard for functional safety, which provides requirements and recommendations for the development of such systems.

One issue addressed by that standard is EMC for a safety-relevant system and its environment. Electromagnetic interference (EMI) between hardware components is a typical cause for failure dependencies (e.g., common cause failures) and for systematic failures in hardware components that realize functions of safety-relevant systems. Physical characteristics of hardware components and interconnecting wires can provide coupling conditions leading to critical malfunctions. Therefore, EMC must be addressed when defining a safety concept for a safety-relevant E/E system and when designing its hardware such that it is sufficiently robust against EMI. However, the parameter space for a specific EMC problem is typically very large. Thus, the evaluation of its electromagnetic (EM) properties is very complex. Trying to simulate all hardware components on a detailed level requires enormous effort. Abstraction of the EMC problem with conservative over-approximation of EMC can help. In order to reduce the complexity and to ensure functional safety w.r.t EMC this work presents a guideline on how to tackle the problem on the system level of the safety process.

The proposed method is a combination of analysis and simulation-based test methods. It allows a systematic identification and evaluation of safety-relevant malfunctions and faults related to EMC. We propose a model-based approach together with contracts. It allows specifying dependencies between functional and physical characteristics of distributed and interconnected components. With it the EMC problem can be addressed for distributed system development. Typically, contracts are used to model and analyze functional characteristics of E/E systems. The electromagnetic influence of hardware components and connector systems (wiring harness) and their relationship to functional system properties is neglected. Thus, we propose a meta-model together with a corresponding workflow to address this problem. In order to cope with the complexity of the EMC problem the scope of this paper is limited to electrical near field coupling. Since the paper refers mainly to the ISO 26262, we use the wording and terms from this standard.

This paper is structured as follows. We first describe the state of the art with regard to this publication. In the following section we introduce the basics of contracts, functional safety according to the ISO 26262, and multi-perspective modeling. Finally, we define the meta-model and the workflow of the proposed methodology in IV and discuss its benefits.

II. State of the Art

In this section, the state of the art is described and discussed regarding methods for safety-relevant automotive systems, EMC, model-based methods including existing standards and tools as well as related work.

A. Safety-relevant automotive systems

Today's automotive system engineering processes and quality management guidelines take into account the topic functional safety respecting the standard ISO 26262 [1] that has been effective since 2011. The standard addresses EMC in the context of robustness as well as systematic and dependent failures on different levels of abstraction, i.e., system and hardware level. Standardized testing methods are referenced for usage to ensure EMI robustness of hardware components as well as parts and to address EMC in system integration tests. Depending on the ASIL safety goals and derived safety requirements may only be violated with a maximum failure rate which is analyzed using quantitative methods like Fault Tree Analysis (FTA) or by applying fault metrics. FTA is defined in [2]. A typical topic regarding the analysis of fault-trees is the analysis of cut-sets and the determination of minimal cut-sets as shown in [3]. In order to analyze the quantitative influence of common cause failures like those related to EMI the parent standard IEC 61508 [4] suggests the usage of a β-factor analysis. The analysis is systematic using a table with questions to determine the β- factor. Implementation assumptions including EMC-related wire alignment such as cable alignment and further environmental characteristics are considered in this catalog of questions. The analysis is for instance supported by the tool FaultTree+ [5] or by Medini Analyze from IKV[6].

B. EMC

EMC is a major topic for automotive industry and subject to many publications, standards and physical simulation tools. A detailed description of cross talk in the automotive environment is given in [7]. The number of tool, suitable for modelling and simulation of EMI of car components seems quite limited. Yet their number increases. EMC Studio [8], for example, provides a useful framework to simulation EMI between cables and devices. The same is possible with FEKO [9]. Both tools allow defining a cable tree, attaching circuits and simulating the interference in terms of parasitic voltages, currents, and critical frequency ranges. Powerful hybrid tools such as COMSOL Multiphysics [10], also allow correct simulation of EMI providing a profound knowledge of electromagnetism. All those tools are equally suitable for the evaluation of EMI at a fixed geometry. The data obtained from such simulation is available for formulating or refining contracts, which can be used in the authors evaluation process.

The authors focus on the development of such a simulation tool which is also able to optimize cable and cable tree geometry with regard to EMC for given external boundary conditions, such as a case or housing. With regard to the topic of functional safety discussed above, an overview about the impact of functional safety on EMC is provided by Kado et al. [11]. The authors of that publication discuss environmental factors like temperature and ageing effects and note their importance for EMC. According to them, a safety manager must have an overview on all EMC activities. Failure modes caused by EMI have to be taken into account during the hazard analysis and risk assessment. Inductive and deductive safety analyses as well as fault injection tests need to be performed. Common cause failures with regard to environmental conditions need to be identified for all safety-critical functions to determine appropriate measures. A respective test plan should be created with regard to EMC relevant characteristics and traceability of EMC related documents needs to be ensured.

C. Model-based approaches

Capturing the complexity is one challenge for safety-relevant systems and also for EMC. This challenge is often addressed by model-based methods and tools supporting them. One related formalism is the use of contracts. Contracts have been subject to various research projects in the context of safety-relevant systems like SPEEDS [12], CESAR [13] or SPES2020 [14]. The contracts enable formal requirements engineering and systems engineering by explicitly distinguishing between assumptions and promises of system components. An overview about contracts is provided by Benveniste et al. [15]. Formal representations of assumptions and promise are often discussed, e.g., regarding the use of Linear Temporal Logic (LTL) formulas for functional conditions or other pattern-based expressions with defined syntax and semantics for various aspects. Among other things virtual integration testing is defined for contracts with functional and safety characteristics allowing formal analysis of consistency and compatibility between system components as well as checking the correct implementation of a contract by components with derived contracts. To analyze integration of components regarding physical conditions typically tools like Simulink [16] or Modelica [17] are referenced. Modelling languages can be used to describe design artefacts and their relationships such as system components and requirements to be satisfied on different abstraction levels and architectural perspectives as described by Damm et al. [18]. An architectural modelling language for automotive system design is EAST-ADL [19]. It addresses topics like hardware modelling, error modelling and traceability between design and requirements on different abstraction levels and architectures. Hardware connectors between hardware components are considered as wires. It is possible to allocate requirements (including requirements on EMC) to both, connectors and components. The alignment of wires can be described by using descriptive concepts from the Harness Description List (HDL) [20]. Functional design traceable with the topological layout of wires can be defined in the tool PREEvision from Vector. It provides different architectural views including a logical architecture view, a hardware architecture view, a wiring harness view, and a geometric view. Therefore, reasoning about the alignment of wires is possible with PREEvision.

D. Related work

A methodology similar to the approach of this document is proposed in previous work [21]. The methodology suggests the usage of contracts to cope with environmental factors related to physical conditions in a dependent failure analysis with regard to the topology of a safety-related automotive system. Simulation methods were recommended to determine potential dependent failures. This approach instruments the heterogeneous rich components (HRC) meta-model developed in the SPEEDS project [22] in order to model contracts. The method is evaluated for temperature. It will be shown how the approach can be extended to investigate wire installations and EMC together with physical simulations.

E. Discussion

From our point of view the presented state of the art lacks in the missing interaction between EMC and functional safety aspects. On the one hand, EMC is typically addressed on detailed levels (II-B), which incorporates several risks. Adhering to the recommendations of standards leads to over-engineered (too robust) systems, or is not necessarily complete regarding all safety-relevant EMI faults of a system under design. Furthermore, the consistency of a system's EMC properties with its electromagnetic environment is not ensured at design time. Unintended additional development loops can therefore be required. On the other hand, an ISO 26262 compliant safety process is systematic and starts at an early stage of system design. But, the standard does not define exact methods or formalisms to integrate and analyze relevant EMC properties with derived safety concepts. In order to perform a β-factor analysis for EMI-related dependent failures EMC characteristics of the system implementation must be known. A systematic design process and derivation of requirements is not considered. The discussed model-based methods, formalisms, and tools only partially address the needs to address EMC in an automotive safety process. For instance, existing contract based methods do not address EMC. Formalisms like EAST-ADL or KBL and tools like PREEvision do not allow to define and analyze all relevant EMC characteristics with functional and hardware architectures or wiring harnesses.

Thus, we propose a method to combine functional and safety views of a system under design with a physical view. By using concepts from the previously published approach [21], extending it with concepts from HDL and PREEvision we are able to link and analyze an automotive wiring topology and related EMI) with a functional safety concept compliant to the ISO 26262.

The main difference to the state of the art is the systematic approach for EMC in the context of safety relevant automotive systems. Starting with the requirements on the level of a functional safety concept, technical EMC requirements are consistently derived for concepts of a safety concept and its environment. EMC requirements are made explicit in terms of physical assumptions and promises for all relevant components.

III. Basics

Our methodology builds upon contracts, functional safety, EMC principles and multi–perspective modelling.

A. Contracts

Our approach includes the usage of contracts. They provide a component specification framework for distributed system development. The usage of contracts allows a formalized argumentation about the combination of the single component's specifications in a defined environment. Contract-based reasoning can be used for early validation in distributed development with different suppliers or company divisions and for the reuse of existing designs. Within the contract specifications, different viewpoints such as functional, safety or physical concerns can be addressed.

The basic idea of contracts is a structured specification of component characteristics in terms of assumption - promise pairs. Crucial for contracts is the explicit definition of assumptions made on the integration environment of a single component. The component itself is considered as a black box. Characteristics promised in the component's specification are only guaranteed if the corresponding assumption holds.

In particular, compatibility between contracts of distributed components and their combined integration in a context with a contract on a higher level of abstraction can be analyzed by applying contract based methods like a Virtual Integration Test (VIT) as shown in [18].

In order to illustrate the structure of contracts, the following contract specifies a function with a promised execution time for processing an output signal b, assuming that an input signal a is available at a certain frequency: Assumption: ”a occurs at most every 100 ms.” Promise: ”whenever a occurs b occurs at most after 80 ms.

B. EMC, Functional Safety and ISO 26262

The automotive functional safety standard ISO 26262 [1] provides a framework of requirements, activities, and recommendations to achieve functional safety for automotive E/E systems. According to this standard, functional safety is a property of E/E systems that guaranties the "*absence of unreasonable risk due to hazards caused by malfunctioning behavior*".

In an early stage of the safety process the E/E system is defined in its boundary on vehicle level as item. For this item hazardous malfunctions are identified in a hazard analysis and risk assessment. These hazards are considered in different operational scenarios as risks. An Automotive Safety Integrity Level (ASIL) is determined and assigned to the risks, rating a risk as safety relevant (ASIL A to D) or as a matter of an established quality management (QM). Safety goals are derived from the safety relevant risks. They go along with further requirements for the system design depending on the ASIL. For each safety goal functional safety requirements are derived, defining together the functional safety concept. They consist of technology independent definitions on how safe states are maintained by a system's elements in order to achieve a safety goal. Dependent failures caused by, e.g., EMI have to be taken into account. Reasoning about them on the level of the functional safety concept typically requires knowledge about the technical realization, which is not complete at that stage of the process. Yet, preliminary architectural assumptions and environmental constraints are typically available and can be refined. Therefore, dependent failures have to be taken into account when defining the technical safety concept and when arguing about its compliance to the functional safety concept. When deriving technical safety requirements from the functional safety requirements, analysis methods like a FTA are recommended to "*identify causes of systematic failures and the effects of systematic faults*". EMC violations are systematic causes related to the integration environment of a hardware element (victim) implementing parts of a safety relevant system. Therefore, EMC is related to integration testing. Also, when reusing an existing technical design, environmental conditions such as EMC shall be checked before implementing a system. Additionally, on hardware level the "*level of robustness shall be demonstrated using feasible test methods*" for which the ISO 26262 refers to typical standards such as ISO 7637 "*Road vehicles - Electrical disturbances from conduction and coupling*". To test whether an EMC-related environmental conditions is relevant for functions safety, the safety-relevant situations must first be identified in the safety process.

C. Electromagnetic Compatibility (EMC)

A component is EM compatible when the emitted interference is low enough to not affect other components or systems and is robust enough not to be influenced by other sources of disturbance. As shown in Figure 1, each atomic EMC problem consists of an aggressor, the victim and at least one coupling path. EMC can generally be improved from three different angles in a system: Reduction of the interference generated by the aggressor, improving EM decoupling or improving the victim's robustness. Simulation of the system can be used to determine the most effective counter measure or verified the compatibility.

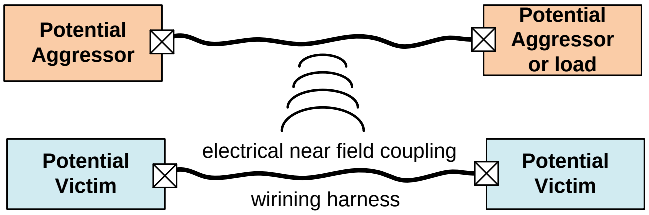


Figure 1. Overview of the aggressor, the coupling path and the victim

In electrical engineering, EM coupling is typically modeled as a feedback from the victim to the aggressor. To describe the aggressor, the victim and the coupling path as impedance networks, design models, and measured data can be used. The component's impedances are either determined by measuring S-parameters or extracted from the design model. Regarding the aggressor, a guideline for modelling an Integrated Circuits (IC) black box emission model is given in [23]. For the characterization of external aggressors the standard IEC 61000-2 [24] can be used. The coupling path can be modeled, as shown in [7], by using a lumped transmission line model.

Currently, simulations use a configurable aggressor model, able to emit standardized output. The victim can be specified either by its impedance or an equivalent circuit. Modelling and simulation of the coupling path is a more complex task. Coupling between wires depends on different variables including their geometry, their materials, and additional surrounding conductors such as a housing. Given all specific parameters for the coupling path, one can calculate the frequency and temperature-dependent coupling parameters, resistance, conductance, capacitance, and inductance per unit length (including their parasitic components) for the specific transmission line(s). An equivalent circuit can be defined, the aggressor and victim model added and the crosstalk simulated using SPICE or any other circuit solver (for a simple schematic model see Figure 2). Those solvers usually instrument different numerical methods (various matrix solvers, finite elements, finite differences) to obtain results for the system of coupled differential equations describing the time and frequency behavior of the circuit. Using suitable numerical methods, qualitative and quantitative results can be obtained in time and frequency domain. Using such EMI simulation in the context of functional safety analysis will be discussed in Section IV-B. The complexity of the simulated system is usually only limited by the simulation hardware.

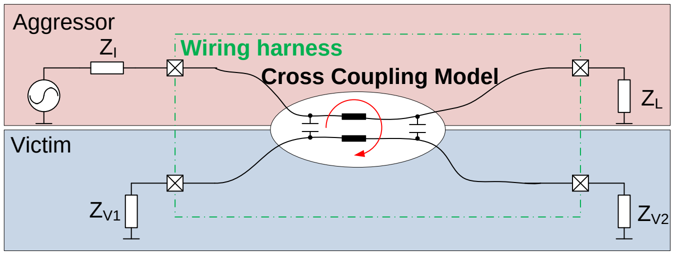


Figure 2. Schematic circuit for aggressor, wiring harness (coupling path) and victim with impedances.

Currently, any possible dependence of EMC to the victim or aggressor operation modes must be considered in the general analysis and cannot be handled by such a simulation. The operation modes of the victim and the aggressor have influence on their EMC properties.

D. Multi-Perspective Modelling

Since EMC is a physical property respective characteristics need to be related to functional design for a model-based argumentation in the context of functional safety. A recent approach described in research projects like SPES 2020 [18] is the description of models at different structural viewpoints called perspectives. Their elements are related to another by means of allocation. This allows for the separation of concerns while preserving a system's properties. For the perspectives the viewpoints operational, functional, logical, and geometrical are considered where this work's focuses on the last three. Functions are mapped to an architectural description in the logical perspective with decomposed and interconnected logical components as a "*description of the logical solution independent from technological constraints*" [25]. A model in a technical perspective is a description of a system's hardware and software, their interactions, as well as their technical and physical properties and constraints. In a geometrical perspective, a model describes the topological layout of a system i.e., a definition of physical positions claimed by elements of a hardware architecture within a reference coordinate system. The idea and the concepts of multi-perspective modelling provide an important base for the proposed EMC method, which will be discussed in IV.

IV. Design Methodology

In this section, the proposed methodology for contract based automotive EMC is presented. At first, the application of the meta-model is discussed. Then the EMC-workflow is presented. We conclude with a comparison of the disadvantages and advantages of our method.

A. Meta–Model

The model that we propose for EMC-related safety analysis extends hardware modelling concepts from EAST–ADL by contracts, wires, and topological information as depicted in Figure 3. The model can be used as an extension of EAST-ADL in order to support the workflow discussed in section IV-B. EAST–ADL is implemented in different ways such as profiles for several UML tools (e.g. Eclipse / Papyrus UML) or with the EATOP platform, all listed by the EAST-ADL association [26]. The concepts and relationships of the model can also be used in a meta-model of a Reference Technology Platform, discussed in research projects like CESAR, MBAT or CRYSTAL, to support the workflow with integrated and interoperating tools, as shown, for instance, in [27]. EAST–ADL allows modelling of a hardware architecture with hardware components, parts and logical as well as electrical connections. The proposed extension allows the definition of contracts with regard to the concepts of the HRC meta-model [21]. The contracts contain specifications of assertions that define the assumptions and promises of the contracts. EAST–ADL allows to allocate requirements to architectural elements like hardware components by using the concept of a satisfy-link. We extend the satisfy-link concept such that contracts can be allocated to the architectural elements. By introducing concepts for wires in our extension, contracts can be allocated to wires of a wiring harness, too. Prototypes (instances) of such wires are related to hardware connectors via a wire mapping allowing assignment of physical properties to the hardware connectors. A crucial part of our proposed model is the introduction of a geometric view with topological information as discussed in previous work [21]. Topological elements can be defined like a component hierarchy with topological nodes as parts and routed interconnection segments in between them. Looking at PREEvision and HDL one would distinguish between hierarchies of installation spaces and locations as topological nodes and installation routes described by topological segments with defined curves between such nodes. Geometrical information is assigned to the hardware elements like prototypes of hardware components and wires by installing them to the topological elements. By such an installation, physical space claimed by the elements of the hardware architecture is defined. Reasoning about physical properties like distance, length of wires, or alignment of wires is enabled this way. These properties are needed for analyses like a dependent failure analysis w.r.t. environmental factors such as EMC.

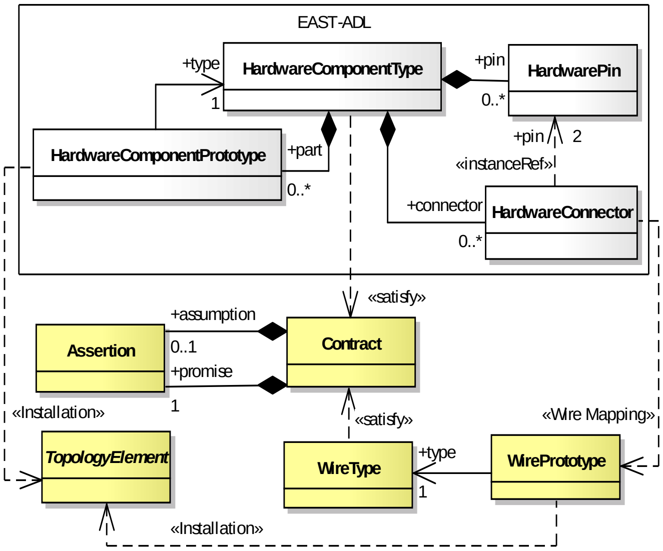


Figure 3. Overview on the meta-model concepts and relationships

A generic instance of the proposed meta–model is as follows. With existing EAST–ADL concepts an automotive hardware architecture with the hardware components of a vehicle (typically ECUs, actuators, sensors) is modelled as described in [19]. By using the concepts and relationships of the proposed EAST-ADL extension, depicted in Figure 3 and described above, a model can be defined in a way that it can be used for the proposed workflow. Safety requirements are formulated as contracts by using the `Contract` concept. Like in [22], assumption and promise of a contract are modelled using the `Assertion` concept with an attribute to carry the descriptions of the respective conditions. As considered by EAST–ADL for safety requirements the contracts are allocated to the `HardwareComponentTypes` of the hardware components by using the `≪satisfy≫` link concept indicating that a hardware component shall satisfy the contract. The wiring harness of a vehicle is modelled by using the concept `WirePrototype` for every wire of the wiring harness. Each wire is typed by a `WireType` defining the type of the wire with its properties. As intended by the proposed methodology the properties are defined in contracts using the `Contract` concept, as described above, and the `satisfy` link concept. A `HardwareConnector` becomes a wire by using the `≪Wire Mapping≫` link concept between a `HardwareConnector` and a `WirePrototype`. The geometric perspective for the hardware architecture and the wiring harness is defined by using the *`TopologyElement`* concept. With it the geometric topology of the vehicle is defined and its installation spaces. Furthermore, with this concept installation locations for the hardware components and routes for the wires are defined. Details on how to define a geometric architecture with concrete topology elements are given in [21]. The hardware components are assigned installation locations by using the `≪Installation≫` link concept between a `HardwareComponentPrototype` and the related *`TopologyElement`*. The hardware components of the hardware architecture are assigned installation locations by using the `≪Installation≫` link concept between a `HardwareComponentPrototype` and the related *`TopologyElement`*. The wires of the wiring harness are assigned routes by using the `≪Installation≫` link concept between a `WirePrototype` and the related *`TopologyElement`*.

For a model-based realization of our workflow, different architectural, mapping, and requirement views are required. These include technical hardware architecture, wiring harness, geometric topology, contract-based requirements, and mapping relationships between the respective views. The proposed meta-model provides concepts and relationships needed for the required views. The development of a technical safety concept (TSC) for an item w.r.t. ISO 26262 is usually done by the Original Equipment Manufacturer (OEM) before the actual implementation is done by the suppliers on hard- and software level. Our goal is to specify and validate EMC on system level in an environment of preliminary architectural assumptions with technical safety as well as functional and EMC requirements.

B. *Proposed workflow*

The prerequisites for our workflow, as illustrated in Figure 4, consist of a Functional Safety Concept (FSC) with functional safety requirements. The functional safety requirements are allocated on system elements that belong to preliminary architectural assumptions. Typically, dependent failure situations cannot be addressed in the functional safety concept because a refined architecture description must be known. Dependencies between functions mainly arise with lower abstraction levels (implementation details), like e.g., shared resources or physical implementation. For EMC, dependent failures regarding EMI and systematic failures in the TSC affecting the elements of the item are investigated. To address EMI faults, the requirements specifications of other electrical systems are also taken into account. The technical specifications and requirements for the wiring harness must be checked as well, because they describe the possible coupling paths. To rate the EM coupling paths, an existing geometrical model that defines the topology of the technical elements and the wiring harness is used. The entire topology model is created by mapping the technical elements and the elements of the wiring harness to the geometrical topology elements as described by the meta-model.

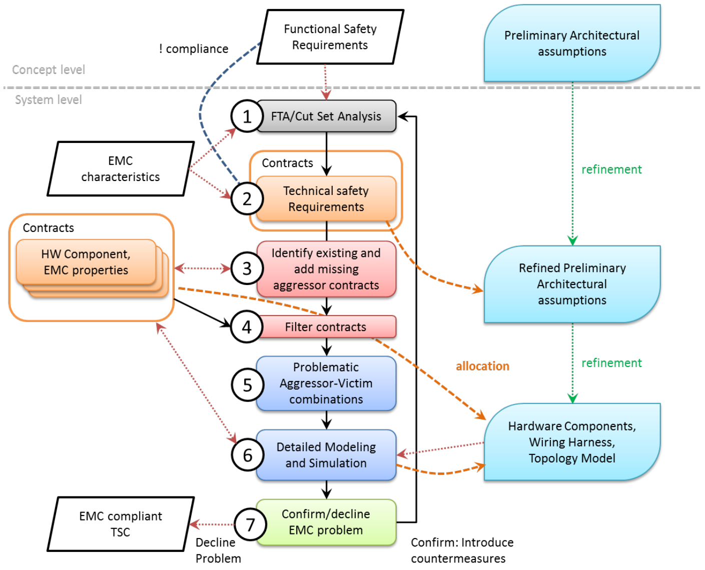

Functional Safety Requirements
! compliance
Preliminary Architectural assumptions
Concept level
System level
1
FTA/Cut Set Analysis
EMC characteristics
Contracts
2
Technical safety Requirements
refinement
Contracts
HW Component, EMC properties
3
Identify existing and add missing aggressor contracts
Refined Preliminary Architectural assumptions
4
Filter contracts
allocation
refinement
5
Problematic Aggressor-Victim combinations
Hardware Components, Wiring Harness, Topology Model
6
Detailed Modeling and Simulation
EMC compliant TSC
7
Confirm/decline EMC problem
Decline Problem
Confirm: Introduce countermeasures

Figure 4. Proposed workflow

The first step (1) of the proposed method is a FTA and a cut-set analysis. In the FTA combinations of possible cause (faults) are analyzed which lead to violations of the safety goals assigned to the item. How to perform a cut-set analysis can be found in [2] or [28]. Additionally, the preliminary architectural assumptions are refined towards an architectural description with technical elements of the item that become hardware components (e.g., sensors, ECUs, actuators). In the FTA also faults related to EMI are visible as base events. The refined preliminary architectural assumptions allow to address failures with regard to the hardware components of the item and to derive a TSC that respects EMC for the hardware components as possible victims. Therefore EMC-related faults are included.

An example for an EMI related fault is the radio frequency (RF) signal, which is inadvertently identified as valid signal. We define EMC faults as violations of electrical-constraints caused by coupled RF signals. These characteristics have to be considered in the FTA leading to base events in the fault tree, which can be potentially caused by disturbances or coupling with other systems. With the cut-set analysis the combinations of the faults including EMC faults, which lead to violations of the safety goals for the item are extracted from the fault-tree.

In the second step (2), the technical safety requirements are formulated. They are derived from the functional safety concept as well as from the FTA and cut-set analysis results including the EMC faults. For each functional specification that can be violated by an EMC fault according to the fault-tree an EMC assumption is formulated. The EMC assumption specifies an EMC constraint whose violation leads to the EMC fault. For instance for the fault ”RF-voltage at sensor port” an assumption is formulated ”RF-voltage at sensor port in [-5 V, 5 V]. Note: Other EMC assumptions can address electro-magnetic characteristics including frequency, time, power or current. The contract defining the technical safety requirement consists of the EMC assumption and the related functional specification as promise. The technical safety requirements are allocated to the technical elements of in the preliminary architectural assumptions. By neglecting the EMC characteristics of the assumptions in the technical safety requirements existing contract-based integration analysis tests, as shown in [29], can be used in order verify compliance between the TSC and the FSC. The compliance between the TSC and the physical environment w.r.t. EMC is analyzed in later steps.

The next step (3) aims to elicit the potential aggressor contracts and the EMC properties. Typically, all technical elements can act as EM aggressors w.r.t. the technical elements of the item including its own. At this point it is assumed that the OEM has already decided which components will be installed. We assume that the requirements specifying EM properties take the form of contracts. In a new environment all contracts are not necessarily available. Thus, data has to be requested by the suppliers or the components have to be characterized. The assumption is typically a functional condition e.g., given by a LTL formula or similar concept like the Requirements Specification Language (RSL) described in [18]. The promise is a physical condition that depends on e.g., input voltage range and operating mode. An example for contracts with physical conditions regarding EMC in the promise is given with the following contracts. They are written down using natural language pattern.

Contract 1:
Assumption: *(mode = off implies SwitchingTime > 20 ms)*;
Promise: *mode = off implies RF voltage(power_out) in [-0.1 V, 0.1 V] with a inner resistance higher 120* Ω,
Contract 2:
Assumption: *(mode = on implies SwitchingTime* ≥ *10 us)*;
Promise: *mode = on implies RF voltage(power_out) in [-20 V, 20 V] with a inner resistance higher 120* Ω,
Contract 3:
Assumption: *(mode = error implies 10 ns < SwitchingTime < 10 us)*;
Promise: *mode = error implies RF voltage(power_out) in [-120 V, 120 V] with a inner resistance higher 18* Ω.

The result of step 3 will typically be a large amount of EM aggressor contracts. Step 4 therefore targets a reduction of the amount of involved EM aggressor contracts by filtering. For the filtering an ideal coupling path is assumed, which is the worst case. An assumption on a voltage between -5 V and 5 V is not violated by hardware components with a guaranteed RF voltage between -5 V and 5 V. So the contract
Contract 1:
Assumption: *(mode = off implies SwitchingTime > 20 ms)*;
Promise: *mode = off implies RF voltage(power_out) in [-0.1 V, 0.1 V] with a inner resistance higher 120* Ω,
from step 3 is not taken. The result is a set of critical victim - aggressor combinations (including modes of operation), for which the TSC is violated. Due to the overestimation not necessarily all of the combinations will violate the TSC in the implementation. Therefore, refinement of the models and analysis as described in Step 5 is necessary.

In Step 5, the correct physical model is created reducing the overestimation. For each critical victim - aggressor combinations from step 4, the coupling model is generated using the physical and geometrical information from the specifications. An example for a contract specifying characteristics of a wire is

Contract 1:

Assumption: *(shielding type = braided & manufacturer_tolerance > -5% frequency range = [10 Hz to 200 kHz])*;

Promise: *shielding_thickness* $\geq$ *1 mm; attenuation > 10 dB*

These models can be simulated using the method described in Section III-C or commercial tools (e.g., FEKO, EMC Studio).

Each outcome of an executed simulation has two possible states regarding the EMC assumptions of the victim:

- The EMC assumptions are not violated within the simulated EM environment with the respective aggressors and coupling paths.

- They are violated by the simulated environment.

For the victim - aggressor combinations that remain after filtering and violate the EMC assumptions, adequate counter measures are specified. Possible measures w.r.t to EMC faults and their impact are described as follows:

- For a violation of an EMC assumption the robustness of the victim has to be increased by changing the EMC assumption (e.g., immunity against RF voltages up to 20 V instead of 5 V).
  Impact: Introduction of hardware measures to achieve a better EM immunity. This will eventually be covered by the supplier.
- If there are no degrees of freedom for the victim's specification, modification of the wiring harness can be required. This can lead to stronger requirements for the affected parts of the wiring harness (e.g., a cable within the harness must be shielded) or to a modification of the geometry of the wiring harness (e.g., change the install of a wire from an identified coupling path to another location).
  Impact: Coordination with the person responsible for the wiring harness.
- If there are no degrees of freedom for the specification of the victim and the wiring harness, it seems reasonable to strengthen the aggressors EMC promises (e.g., voltage output in case of error below 20 V instead of 50 V).
  Impact: Coordination with the person responsible for the aggressor.
- If there are violations of EMC assumptions in case of failures related to the aggressors (e.g., error states) and there are no degrees of freedom for changes on aggressor, victim or wiring harness, the respective EMC assumption can be removed and instead a safety mechanism is specified. The definition of the safety mechanism takes the form of a contract and is specified in the promise. The safety mechanism typically defines how a safe state is maintained (e.g., Promise: *Whenever the voltage at sensor port is larger than 20 V the sensor interface shall deliver the code "No valid data"*.
  Impact: New verification of compliance between functional and technical safety concept.

The order of the different measures is exchangeable and depends on the effort to be spent as well as the influence of the OEM on the affected suppliers. After a “safety measure” has been chosen, the fault tree is adjusted and the cut-set Analysis performed again. The modification(s) have to be applied to the requirement specifications and the TSC, also, the architecture models have to be adjusted. Depending on the modification, some simulation models might not change and therefore don't need to be investigated again. The determination of such models will not be discussed during this work.

C. Summary

The benefits of the presented method are the following: Coarse filtering of dependencies and identification of possibly safety-relevant problems have been simplified by using a systematic model-based approach with contracts. Early validation of the TSC for typical EMC problems is possible, reducing unnecessary iteration loops during testing and development. Changes and/or removal of hardware components do not necessarily force a reevaluation of the TSC. A less pessimistic EMC design is possible, the OEM can reduce overprotection of individual technical system elements. Including topological information to the safety analysis allows characterization of the EM coupling paths. The approach and the proposed meta-model allows traceability of functional and physical specifications in the safety process.

The drawbacks, on the other hand, are: Assumptions about the completeness of the system definitions must be made. If a technical system element is added, a major change is committed, which results in reevaluation of the TSC. The restriction to near field coupling is an approximation, as simulation of all possible EM disturbances is not possible. Also, the initial effort to specify the architecture is large.

To summarize, a structured approach for identification of EMI problems of safety-relevant automotive systems has been shown. It is important to point out, that conditions of a component's physical environment are now accessible for investigation of safety concerns on system level. The use of contracts allows to consider EMC victims and aggressors as black boxes. Their EMC characteristics are identified as part of the safety lifecycle w.r.t. the ISO 26262. An argumentation about their integration is done at an early stage of the development process and takes the geometric topology into account.

V. Conclusion

In this paper, we proposed a methodology to address EMC for safety-relevant automotive systems. For the methodology we consider contract-based specifications and a meta-model that allows to characterize functions and their relationship to distributed E/E components in a hardware topology with EMC properties. The proposed model is a possible extension of EAST-ADL and allows modelling of topological information for a hardware architecture and to determine the alignment of wires forming coupling paths. The model ensures traceability of related engineering as well as analysis and test information as required by the ISO 26262. By applying the proposed method an early validation of EMC characteristics is enabled on system level before E/E components are actually built and integrated. The dependency between OEM and supplier can therefore become less pessimistic. Safety-relevant EMC problems are identified and addressed on concept level. In contrast to existing methods that make assumptions on a system's implementation regarding EMI robustness in safety-analyses and consider robustness tests on lower design levels with the proposed method safety-relevant EMC characteristics are identified and validated at an early stage of the development process. They are systematically written down in requirements using contracts for involved components of a safety-relevant system and its physical environment. With the proposed method the effort needed to achieve EMC in a development process for a safety-relevant automotive system is focussed on safety-relevant EMI characteristics.

VI. Acknowledgment

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement Nr. 295311, and Nr 269335, as well as from the German Federal Ministry of Education and Research (BMBF) under the grant number 01IS11003L and the Austrian Research Promotion Agency FFG under the program ”Forschung, Innovation und Technologie für Informationstechnologien (FIT-IT)”.

References

[1] ISO, *Road Vehicles - Functional Safety*. International Standard Organization, November 2011, ISO 26262.

[2] W. Vesley, F. Goldberg, N. Roberts, and D. Haasl, *Fault Tree Handbook*. U.S. Nuclear Regulatory Commision, 1981.

[3] T. Peikenkamp, A. Cavallo, L. Valacca, B. Eckard, M. Pretzer, and E. M. Hahn, “Towards a Unified Model-Based Safety Assessment,” *Assessment*, pp. 275–288, 2006.

[4] IEC, *Functional safety of electrical/electronic/ programmable electronic safety-related systems*, iEC 61508, all parts.

[5] [Online]. Available: www.isograph.com

[6] ikv++ technologies ag, *Medini^{TM} analyze functional safety analysis for ISO 26262* , February 2013, Version 2.1.

[7] S. Alexandersson, “Automotive electromagnetic compatibility - Prediction and Analysis of Parasitic Components in Conductor Layouts,” Ph.D. dissertation, Lund University, 2008.

[8] [Online]. Available: www.emcos.com/

[9] [Online]. Available: www.feko.info

[10] [Online]. Available: www.comsol.com

[11] R. Kado, J. J. Nelson, and W. Taylor, “Impact of Functional Safety on EMC: ISO 26262,” in *Proceedings of SAE 2013 World Congress & Exhibition, Detroit, MI, USA*, April 2013, SAE Technical Paper 2013-01-0178, 2013, doi:10.4271/2013-01-0178.

[12] [Online]. Available: www.speeds.eu.com

[13] [Online]. Available: www.cesarproject.eu/

[14] [Online]. Available: http://spes2020.informatik.tu-muenchen.de/

[15] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger, and K. Larsen, “Contracts for Systems Design,” INRIA, Research Report N°8147, November 2012, project-Teams S4.

[16] [Online]. Available: www.mathworks.com

[17] [Online]. Available: www.modelica.org

[18] W. Damm, A. Baumgart, E. Böde, M. Büker, G. Ehmen, T. Gezgin, S. Henkler, H. Hungar, B. Josko, M. Oertel, T. Peikenkamp, P. Reinkemeier, I. Stierand, and R. Weber, “Architecture Modeling,” OFFIS, SPES2020 Project Result, March 2011.

[19] EAST-ADL Association, *EAST-ADL Domain Model Specification*, May 2013, version V2.1.11.

[20] M. Ungerer and O. Rabe, *Harness Description List (KBL)*, 2005, Version 2.3 SR-1.

[21] A. Baumgart, “A Contract-Based Installation Methodology for Safety-Related Automotive Systems,” in *Proceedings of SAE 2013 World Congress & Exhibition, Detroit, MI, USA*, April 2013, SAE Technical Paper 2013-01-0192, 2013, doi:10.4271/2013-01-0192.

[22] SPEEDS Project, *SPEEDS L-1 Meta-Model*, May 2009, SPEEDS WP.2.1 Deliverable D.2.1.5, Revision 1.0.1.

[23] IEC, *IEC62433-2-1, Ed.1 - EMC IC modelling Part 2-1: Theory of black box modelling for conducted emission*. International Electrotechnical Commission, 2009.

[24] ——, *IEC61000-2 Environment: description, classification; compatibility levels*. International Electrotechnical Commission, 2008.

[25] K. Pohl, H. Hönninger, R. Achatz, and M. Broy, *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*. Springer, 2012, ISBN 978-3642346132.

[26] EAST-ADL Association, “EAST-ADL Association,” Website, http://www.east-adl.info/, 2013.

[27] A. Baumgart and C. Ellen, “A Recipe for Tool Interoperability,” in *Proceedings of MODELSWARD 2014*. SCITEPRESS, 1 2014, pp. 300–308.

[28] Goble, *Control Systems Safety Evaluation and Reliability, 3rd Edition*. International Society of Automation (ISA), 2010.

[29] W. Damm, B. Josko, and T. Peikenkamp, “Contract based ISO CD 26262 safety analysis,” in *Proceedings of SAE 2009 World Congress & Exhibition, Detroit, MI, USA*, April 2009, SAE Technical Paper 2009-01-0754, 2009, doi:10.4271/2009-01-0754.

Agent-based Simulation of Information and Communications Technology Practicum Courses for Engineering Students at Instituto Tecnológico Metropolitano

Julian A. Castillo, Cristian F. Gallego, Yony F. Ceballos, Carlos O. Zapata.
Engineering faculty
Instituto Tecnológico Metropolitano
Medellin, Colombia
{castillojulian, pharsat, fceball}@gmail.com; carloszapata@itm.edu.co

***Abstract*—In this paper, a simulation of enlisting and use of free practicum courses for engineering students from Instituto Tecnológico Metropolitano (ITM) is presented. Currently, this university is interested in getting better results to rank more competitively among other universities. Although the current courses are free, a high quality offer is evidently more expensive and we need to maximize the impact of the students in the labor market. In Colombia, many universities certify engineering students in computer sciences so, in order to attend the current needs of our institution and the labor market, we would like to contribute to decision-making processes through agent-based simulation regarding what Information Communications Technologies courses may fit best for the community and the industrial clusters ITM students are and will be working in.**

Keywords-Agent-based modelling; Computer engineering; Simulation; Netlogo.

I. INTRODUCTION

The ITM is a public university whose mission aims to provide charge-free education in different areas. Besides, ITM is in touch with the needs of the labor market. However, the current techniques to identify educational mechanisms to meet labor market needs are based on online surveys applied to students during practicum and to recently graduated bachelors. This may become a reason to use simulations that help determine how these different techniques of information gathering on the educational offer will allow this population to impact positively the labor market in the future, making possible to see devise educational mechanisms for students to learn about new technologies in the industry and be able to found cutting-edge enterprises.

ITM bachelors have particular features. They are educated through propaedeutic cycles and belong to low socioeconomic sectors of the city (see Figure 1). Therefore, the institution makes efforts to form competitive bachelors allocating resources for them to have access to free training during their practicum stages, so that they are able to improve their quality of life and learn the skills necessary to compete in equal conditions with students from other highly qualified universities. In order to contribute to decision-making processes aligned with the university' mission and its population, modeling and agent-based simulations are proposed in this research to identify where it might be necessary to provide student training and decide how the university should invest resources to meet the needs of students and local enterprises.

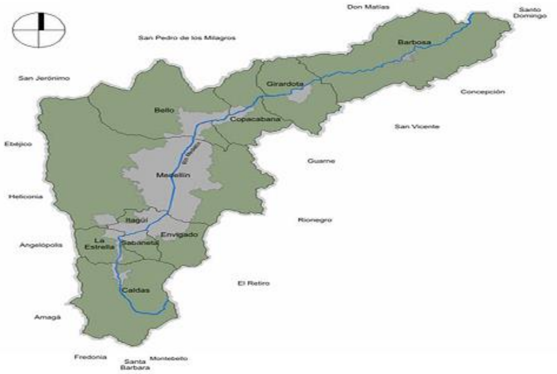


Figure 1. Valle de Aburrá Urban Area. Grey sectors represent bachelors geographical location in the Metropolitan area.

As the student and bachelor population of the institution is small and many simulation techniques usually need a huge amount of input data, we have used Netlogo. This software is used to provide low abstraction levels to characterize a small population in a simple and detailed manner, resulting in a reliable and easy way to validate a simulation [2]. Moreover, we have chosen this low-cost simulation tool that provides close-to-reality results regardless of small populations. Figure 2 displays the distribution of enterprises in the urban area of Medellin city, those who are the main information source for this model.

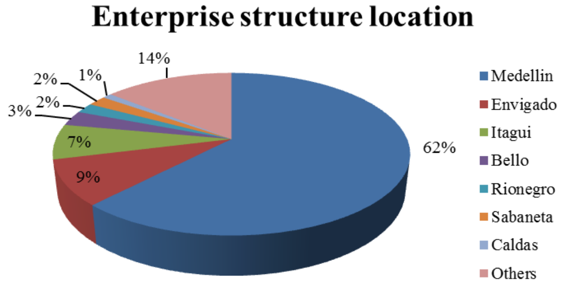


Figure 2. Distribution of enterprises in Valle de Aburrá urban area.

For this simulation, local industrial clusters were identified in the city [3], which generally hires bachelors from different higher education institutions. These clusters are leading organizations of software development groups. They are located in strategic coordinates of the city. Nearly

4-5 enterprises per cluster are development companies. From these, two of the most remarkable are the Tata Consultancy Services Ltd (TCS) cluster, located in Olaya Enterprise Center and Hewlett-Packard in Ruta N Cluster (see Figure 3).

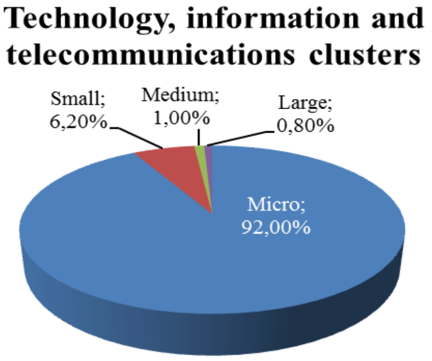


Figure 3. Typology of Enterprises in Medellin Urban Area [3].

For this simulation, we have also developed an agent-based model. The purpose was to know the relation between ITM bachelors' interaction in the mentioned market sectors (clusters) with the creation of training courses at ITM. These relations have given rise to a number course programs that respond to the wishes of students and companies to reach an intermediate point where the decisions in training and learning and training are the most appropriate for both agents.

Section III discusses the ODD (Overview, Design Concepts and Details) protocol [4] used to build the simulation. Then, we will present an overview of the description of the model. Next, in section IV, we will report the results of a set of preliminary simulation experiments. Finally, in the section V and VI, we will provide concluding remarks of our study and provide a summary of the future work that has been projected from this study.

II. JUSTIFICATION

The proposed research project is part of the modeling and simulation of systems by agents, which is an area that has been barely explored in Colombia due to its high computational requirements and random components [1].

This technique differs from other modeling techniques in the way that abstraction of the real system and consequently, the functional model, is constructed. In the models constructed using agent-based simulation, the basic components of the real system are explicitly and individually represented in the model. Thus, defining the basic components of the real system correspond to the boundaries that define the model agents, and the interactions that take place between the basic components of the real system correspond to the interactions that take place between agents in the model [5][6]. This direct correspondence contrasts with the traditional use of 'representative agents' and is able to increase the realism and scientific rigor of formal models built using this approach. Agent-based systems are characterized by comprising several agents that are to a greater or lesser degree, autonomous, heterogeneous and independent, each showing their own goals and objectives, and are generally able to interact with each other and with their environment [7].

They are usually characterized by the existence of a large number of relatively simple agents that can evolve over time to adapt to new environmental conditions or to new systems objectives. The agent-based methods facilitate the study and modeling of complex systems from their component units, allowing to build experimental models of reality from a different point of view to the traditional: from the simplest to the most complex. Furthermore, an intrinsic agent’s result is the idea of an emergency. Emergent phenomena are patterns that emerge from the decentralized interactions of individual simpler components [8].

What characterizes these emerging phenomena is that their presence or appearance is not apparent from a description of the system consisting of the specification of the behavior of the individual components and rules of interaction between them [9]. A typical example of an emerging phenomenon is the formation of distinct groups in the Schelling segregation model [10]; the emergence of clear patterns of segregation is not explicitly imposed in the model definition, but emerges from the local interactions of individuals with segregationist tendencies in weak times. Another example is the migration patterns of Sugarscape [11]. In the next section, we show the ODD protocol [12] used in agent-based models.

Given the small number of courses, the population size, geographic location and heterogeneous characteristics of the population, the simulation technique used is based on agents as it provides a complete mapping of the selection process.

III. SIMULATION PROCESS

We use an agent-based approach to build a simulation of the relationship between the ITM for bachelors with medium to low income levels in the Metropolitan Valley in Medellin – Colombia and practicum courses. This involves a relation made by the bachelor with the ITM and the company where the bachelor works for, the training courses that are made for them and how they make the decision based on a company request and their own preferences and if the course is available in the curricular offer. The process is described as the ODD protocol [4].

A. Purpose

The purpose of this model is to make decisions for the future position ITM a pioneer in the process of educating highly trained professionals that learn in relevant aspects needed in business and the Institution. This leads to making decisions based on the model to allow the institution to grow in terms of academic extension courses.

B. State variables and scales

Bachelors are located in the metropolitan area and are related as agents with a cluster. This determines their initial position as return values when the learning cycle ends. Each bachelor also has a preference for a specific course given by a vector of 10 positions on which each position determines their sympathy for the course of their position, e. g., if at the 3 position vector of sympathy there is a value close to 10, it means that their sympathy to take the course "3" is high; instead, if at position 4 there is a value close to zero, it means that their sympathy to take the course "4" is low. Both the student and the company for which they work (Cluster) have preferences and it is this relationship that determines the selection of a training course.

The cluster as agent has the ability to create its own sympathy for courses randomly, which is defined as a 10-position vector where each position represents a course. There is also a unique cluster number that identifies its own position.

The university as an agent is related to its two main sites that are identified by numbers. With one to two, each has a random vector of courses that are offered, which cannot be further repeated also defined in 16 weeks with a weekly intensity of 4 hours.

An array of preferences associated with each cluster and student is defined. After that, a new array is created using the addition from position to position of the elements of both arrays, of which a maximum value for the course is chosen. This value represents the most suitable for the cluster and student, simultaneously. To select this position, a run on the array of partial sums is made. The ITM has an estimated dropout rate with a maximum value of 30% of bachelors enrolled in training courses; this value is cumulative for each bachelor and accounts for a temporary variable that determines the possibility of dropping out from or a course.

The main attributes of these three classes of agents are shown in Figure 4.

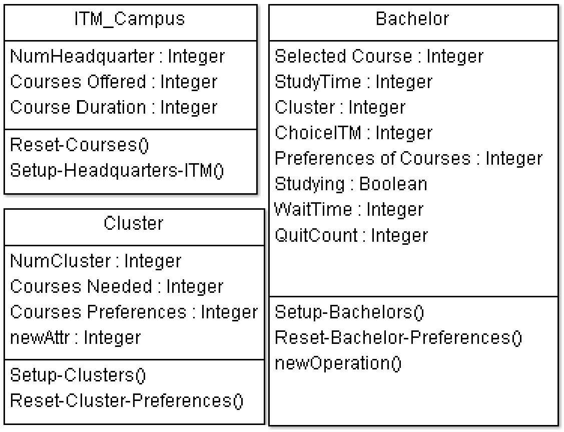


Figure 4. Class diagram agent.

All agents need to be initialized with values for every variable. The values are generated based on a set of hypothetical assumptions.

C. *Process overview and scheduling*

Description of the World: The world also known as 'work area' is defined by the location of the agents located in the Valle de Aburrá within the Department of Antioquia - Colombia. Valle de Aburrá has several locations that belong to the metropolitan area. This simulation considers the clusters closely located to the influence area of the ITM. The simulated clusters are located as seen in Figure 1.

According to the dictionary of technology companies registered in ClusterTic, these were referenced by 6 strategic points located spatially on xy coordinates on the map [1]. Each cluster can be identified as follows: 1 Cluster Ruta N, 2 Bello Cluster, 3 Poblado Cluster, 4 Olaya Centro Empresarial Cluster, 5 Candelaria or Downtown Medellín Cluster and 6 Envigado and Sabaneta Cluster. For the space, each patch is defined with a unit value and a 400 by 400 patches long map on which the image is overlaid with political delimitations of Valle de Aburrá. Each agent interacts in the map and each is spatially associated with each other. The loaded image below has been enhanced to visualize this datum using a filter process in 3 scales: gray, black and white (See Figure 5)

Agents Interaction: Agents that interact in the process are denominated: Bachelors, Clusters, and ITM's (two campuses). The central element on which the simulation is performed is the Bachelor. We aimed to simulate whether a bachelor takes a training course or not and whether the bachelor finishes it or drops it, Nonetheless, this information measurements are taken into a simulation process of 10 periods of 2 years interactions each.

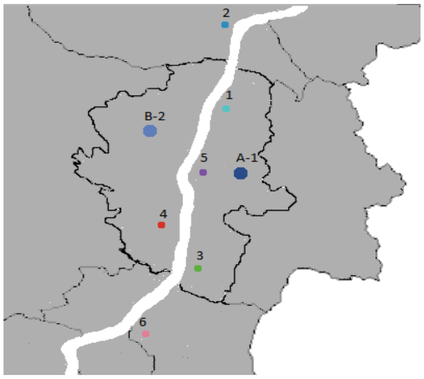


Figure 5. Location of cluster and university campus.

The main process is defined as the student’s empathy to choose one training course mixed with the enterprise needs they belong to, and this process is conditioned to make a decision about the ITM course offers.

Bachelors: Each bachelor is born randomly in each cluster; the number assigned to them is according to a max number of bachelors determined by a random variable ranging from 100 to 500 agents. For each bachelor, it is defined an array of 10 random values for training course selection empathy of each training course [2]. These values have no relation between them.

Cluster: Each cluster is an agent. A cluster is created spatially approximating to the actual position in the metropolitan area for a total of 6 clusters. Each has a number that identifies it, and has a need for knowing of same training courses. For instance, when an enterprise recommends that a bachelor employee (bachelors) take a certain course, there is a preference for courses that would affect the decision of the graduates.

The ITM: It is composed of two branches, each of them represented as agents, and they are symbolized by an identification number (Figure 5 B2 is for the Robledo Section in the west of the metropolitan area and the A1 is for the Fraternidad placed in the east of the same area). Each ITM campus offers a random number of training courses from 1 to 10 as maximum; one ITM campus cannot have all courses [2]. The relationship between each entity is shown in Figure 6.

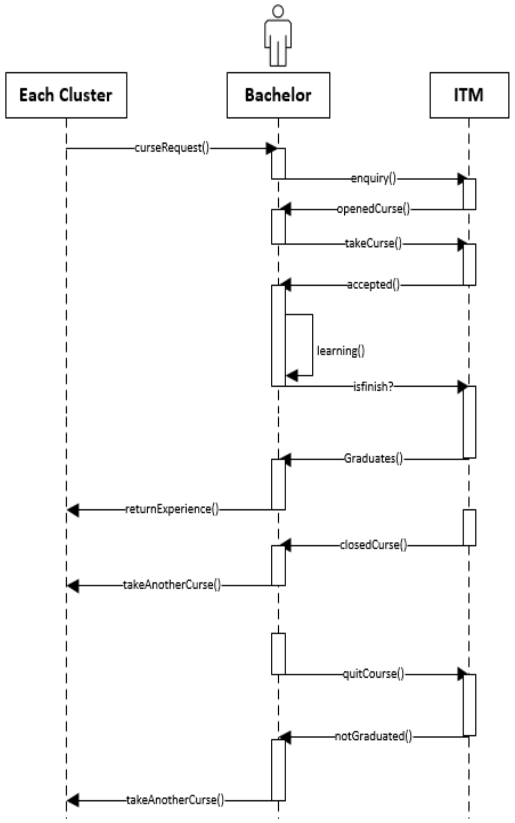


Figure 6. Sequence Diagram (Bachelor and training course flow).

Decision: As seen in the section of state variables and scales, the decision of a bachelor is taken according to the highest greater value of sympathy for a course added to the sympathy of the courses by the company. After this, the algorithm verifies that the student is not taking courses by measuring two bachelors' own variables, which are "chosen course" and "study time"; both must be zero in order to make the assignment of the course.

The decision is assigned to the variable "course chosen”, having the decision taken involving both the company and the students. Then, the bachelor proceeds to verify in which ITM campus this training course is offered.

Training course in any ITM campus must exist in order to move a bachelor for training, which will take 16 weeks decreasing with each time lapse in the software (Tick) one to one.

In order to do the training when a bachelor is assigned to an ITM, is assigned the course time left for complete it (16 weeks, 16 ticks) this number decreases with each tick; when it reaches to zero, then the bachelor leaves the ITM returning to the original cluster. When a bachelor returns to their own original cluster, their training course preferences refresh.

There is a possibility of that a bachelor not to take a course, however is needed a required value of sympathy for its selection. This happens when the training course chosen by the bachelor and by the enterprise is not offered by any ITM campus. In this case, the bachelor awaits a maximum of 5 weeks; then, the bachelor refreshes their own preferences.

Every 20 weeks training courses are refreshed by the ITM, also it holds a 30% bachelor desertion rate. In order to simulate this variable, we defined it by a pseudo-random value between 0 and 1. If this value is lower than the exogenous variable "likelihood of attrition" (0.3), it is excluded from the course. After this, training courses preferences are refreshed.

For this process, we defined 104 weeks or two years, being the average period for an extension academic program.

D. *Validation*

The fitting analysis of the data obtained by the simulation is performed using the information contained in [2], compared to historical data from the year 2013. In addition, expert knowledge is used in order to validate the seasonality of the series obtained together with the characteristics that define each agent and the taking courses frequency.

IV. RESULTS

The simulation setup was executed having the following parameters: the inclusion probability of each training course for bachelors was 50%, the probability of desertion was estimated on 15%, the amount of bachelors was 150, and the empathy probability with a training course was presented with 10%. For the simulation process, we set and run a 10 time round of simulations for periods of two years each one, meaning 104 weeks or ticks. For coding, the software Netlogo ® was used [13].

As seen in Figure 7 for the Robledo campus, there is an oscillation in the amount of bachelors in the training courses for all clusters. However, around week 54 there was a re-initialization of the bachelors who started doing another training course. This may have occurred because the courses are restarted at week 60 of the academic semester.

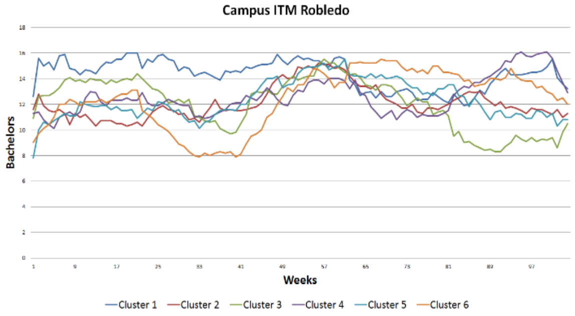


Figure 7. Bachelors taking a training course by cluster.

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-371-1

In the Figure 8, there is a change in the trend because in the description of the model courses are redefined based on information from the office of academic extension.

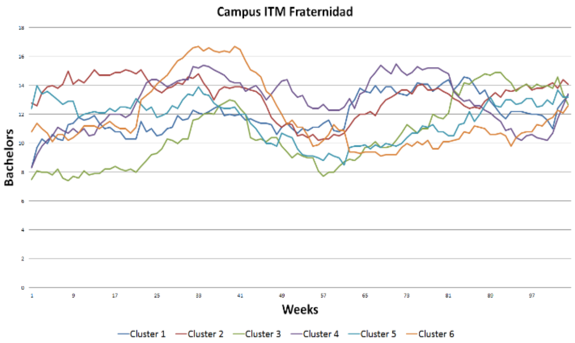


Figure 8. Bachelors taking a training course by cluster.

The graphs are related because the students in the courses belong to the same population and it is presented in Figure 7 as a valley and in Figure 8 as a peak and vice versa. This is caused by the courses offered per year at each campus are exchanged so that generates a migration of the number of students between campuses.

In Figure 9, we can see the student dropout, which is exacerbated in the valley of the simulation periods, as presented in the above figures is observed.

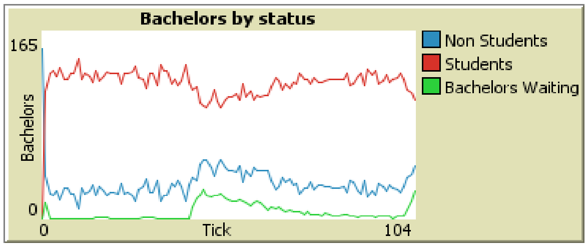


Figure 9. Bachelors taking a training course by cluster.

It is suggested and implementation of short courses are planned to capture students who drop out often, given the duration of 16 weeks of regular courses.

V. Conclusions

According to the results obtained for the simulation generates an inverse correlation in the number of students shown in both graphs, it means when there is increase of Fraternidad students will notice a decrease in the Robledo campus.

The variability of the courses influences the dispersion of students through them, and if the same training course is offered in both campuses the amount of bachelors decrease in the related University.

The flow on the increase and decrease of the number of students per campus is consistent with the distribution of students in courses on campus with a fluctuation of 20 weeks

To improve competitiveness in the local market of universities and the possibility for graduates to access a better labor compensation, will be promoted from the first half of 2015 a project based on this simulation, taking the information generated in the cluster and weeks in which the courses are taken. The offer will be screened with seasonality allowing a proper rotation of courses taught previously. Desertion of each student and students waiting will be considered for a course as well as seen in Figure 9. This is necessary to reduce costs because the university takes the fee of these courses.

VI. Further Work

For future work, the amount of students who change campus completing courses will be quantified. We will measured how many students quits courses, as is the average waiting time of a student with a chosen course without availability by the University, the preferred course selected by students and their relationship with their respective campus.

References

[1] AMVA (Área Metropolitana del Valle de Aburra), “Area Digital - Area urbana de Medellin,” Municipio de Medellin, 2014. Available in spanish http://www.clustertic.co/site/DescubraelemClusteremTIC/Qu%C3%A9esiClusteriTIC.aspx. [retrieved: august, 2014]

[2] I. Ramirez, “Characterization of college graduates ITM” “Caracterización de egresados de la institución universitaria ITM,” Medellin, 2013. Available in spanish http://www.itm.edu.co/autoevaluacioninstitucional/acreditacion/imagenes/Anexos/73.InformesCaracterizacionEgresados20122.pdf [retrieved: august, 2014]

[3] AMVA (Área Metropolitana del Valle de Aburra), “Cluster Technology, Information and Communication” “Cluster Tecnología, Información y comunicación,” Municipio de Medellin, 2014. Available in Spanish http://www.clustertic.co/site/Comunidad/Directoriodeempresas.aspx. [retrieved: august, 2014]

[4] V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, et al, “A standard protocol for describing individual-based and agent-based models,” Ecol. Modell., vol. 198, no. 1–2, Sep. 2006 pp. 115–126.

[5] B. Edmonds, A. Building, and A. Street, “The Use of Models-making MABS actually work,” Cent. Policy Model. pp. 15–32, 2000.

[6] Q. B. Le, S. J. Park, and P. L. G. Vlek, “Land Use Dynamic Simulator (LUDAS): A multi-agent system model for simulating spatio-temporal dynamics of coupled human–landscape system,” Ecol. Inform., vol. 5, no. 3, May 2010, pp. 203–221.

[7] I. S. Torsun and I. Torson, Foundations of intelligent knowledge-based systems. Academic Press, 1995.

[8] J. H. Holland, Emergence: From chaos to order. Oxford University Press, 1998.

[9] N. Gilbert, “Agent-based social simulation: dealing with complexity,” Complex Syst. Netw. Excell., 2004.

[10] T. C. Schelling, “Dynamics models of segregation,” J. Math. Sociol., no. 1, pp. 143–186, 1971.

[11] J. M. Epstein and R. L. Axtell, Growing Artificial Societies—Social Science from the Bottom Up. MIT Press, 1996.

[12] J. G. Polhill, D. Parker, D. Brown, and V. Grimm, “Using the ODD Protocol for Describing Three Agent-Based Social Simulation Models of Land-Use Change,” J. Artif. Soc. Soc. Simul., vol. 11, p. 3, 2008.

[13] U. Wilensky, “NetLogo: Center for Connected Learning Comp.-Based Modeling,” Evanston, IL: Northwestern Univ., 1999.

A Unified Framework for Uncertainty and Sensitivity Analysis of Computational Models with Many Input Parameters

Li Gu and C. F. Jeff Wu

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia
Email: lgu@gatech.edu and jeffwu@isye.gatech.edu

***Abstract*—Computational models have found wide applications in simulating physical systems. Uncertainties in input parameters of the system can greatly influence the outputs, which are studied by Uncertainty Analysis (UA) and Sensitivity Analysis (SA). As the system becomes more complex, the number of input parameters can be large and existing methods for UA and SA are computationally intensive or prohibitive. We propose a unified framework by using a hierarchical variable selection approach to connect UA and SA with one design. By incorporating the effect hierarchy principle and the effect heredity principle, the method works well especially when the number of input parameters is large. Since the procedure requires only one design, it is economical in run size and computationally efficient.**

Keywords–Uncertainty analysis; Sensitivity analysis; Screening; Effect hierarchy principle; Effect heredity principle; Polynomial chaos expansions.

I. Introduction

The understanding and analysis of complex physical systems relies increasingly on computer simulations. Simulations are based on computational models with input parameters. In many practical situations, there is uncertainty in the choice of the input parameter values. Thus, the understanding of uncertainties in the input parameters and their impact on the output becomes essential. Two major tools, Uncertainty Analysis (UA) and Sensitivity Analysis (SA), are often employed for this purpose.

UA studies how uncertainties in the input parameters can be mapped to uncertainties in the outputs. A typical method for UA is Monte Carlo simulation, which works as follows. The inputs are determined by generating random samples, following a distribution that characterize the uncertainties of the input parameters. Then analyze the empirical cumulative distribution function of the outputs. See [1] for a review. SA studies how the total output uncertainty can be attributed to uncertainties in each input parameter. It can be done locally or globally, but we only focus on global SA in this paper. Global SA methods can be gathered into two categories: direct computation and metamodeling. Methods in the first category directly use Monte Carlo simulation to compute sensitivity indices, like FAST [2] and Sobol' indices [3]; Methods in the second category build a metamodel to replace the computational model for subsequent statistical analysis. Various metamodels were considered for SA, such as linear regression model [4], Gaussian process model [5], polynomial chaos expansions [6], and smoothing spline [7].

SA becomes complicated when the number of input parameters is not small. Based on the effect sparsity principle [8], only a few (but unknown) parameters are significant among the many candidates. Therefore variable selection, which selects a short list of important parameters, is applied before SA to reduce the computational burden, and to improve the accuracy. Variable selection methods usually need extra function evaluations with the use of special designs, such as systematic fractional replicate design [9], sequential bifurcation [10], or elementary effect method [11].

One can choose the aforementioned methods to sequentially conduct UA, variable selection, and SA. However, this will require too many function evaluations because three separate methods and designs are involved. Especially for variable selection, existing methods are computationally intensive or prohibitive when the number of input parameters is very large and the computational model nonlinear.

In this paper, we propose a new framework, where only one design is used for the three steps. For UA, we run Monte Carlo simulation with Latin hypercube samples [12]. The *Polynomial Chaos* (PC) expansions is used as the metamodel. By keeping the same samples for UA, a variable selection approach designed to handle many input parameters is employed for selecting significant linear, nonlinear and interaction effects. Then, the sensitivity indices (Sobol' indices) of the selected parameters are computed *analytically* from the PC coefficients without any extra function evaluations [6]. Because we need only one design in the three steps, the proposed approach can drastically reduce the computational time in the simulation. Also, it works efficiently for high dimensions (*i.e.*, the number of input parameters p is large). We focus on the selection of input parameters, which is a bridge connecting UA and SA. Details of UA and SA can be found in [1] [6] [13], and are hence omitted.

The rest of the paper is organized as follows. Section II reviews the PC expansions, which is the metamodel we use. Two variable selection methods (the sure independence screening and the lasso) are reviewed in Section III. Section IV introduces a hierarchical variable selection approach which can handle many input parameters. A numerical example is given in Section V. Concluding remarks are given in Section VI.

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-371-1

II. Polynomial chaos expansions

Computational models are black-box functions, which may have complex relationships, such as nonlinear effects and interactions. In most cases, the linear model does not work well as a metamodel to approximate computational models. We consider the PC expansions as the metamodel. A brief review of the PC expansions is given in this section.

Denote the computational model by

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

where f is the computer code, $\boldsymbol{x} \in \mathbb{R}^p$ is the p-variate input, and y is the output. Suppose the basis $\Psi(\boldsymbol{x})$ is constructed by the tensor product of orthogonal polynomials $\phi(x_j)$, $j = 1, \ldots, p$, as follows:

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \phi_{\alpha_1}(x_1) \cdots \phi_{\alpha_p}(x_p), \tag{2}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)$, and α_j corresponds to the order of ϕ_{α_j}. The order of $\Psi_{\boldsymbol{\alpha}}$ is defined by $|\boldsymbol{\alpha}| = \sum_{j=1}^{p} \alpha_j$. (The choice of orthogonal polynomials varies depending on the distribution of x, which can be found in [14].)

Then, the PC expansions is

$$g(\boldsymbol{x}) = \sum_{k=0}^{\infty} \beta_k \Psi_{\boldsymbol{\alpha}_k}(\boldsymbol{x}), \tag{3}$$

where β_k are unknown coefficients. The order of $g(\boldsymbol{x})$ is defined by $\max_k |\boldsymbol{\alpha}_k|$. Note that $g(\boldsymbol{x})$ with a proper choice of β_k converges in quadratic mean to $f(\boldsymbol{x})$ [15]. The *truncated* PC expansions involving only a finite number of bases is used for practical computations, and is denoted by

$$g_P(\boldsymbol{x}) = \sum_{k=0}^{P} \beta_k \Psi_{\boldsymbol{\alpha}_k}(\boldsymbol{x}), \tag{4}$$

where P is usually decided by restricting the order of the PC expansions.

The coefficients β_k in (4) need to be determined. In stochastic mechanics, stochastic spectral methods using a Galerkin minimization technique that minimizes the residual in the balance equation have been widely used [16]. Such methods are intrusive, because an *ad hoc* modification on the computer code of each problem is required. Some non-intrusive methods were recently proposed for estimating coefficients, where the computation can be done with a set of deterministic model evaluations. Sudret [6] used the regression method. A projection method was discussed in [17]. Throughout this paper, we follow [6] to estimate coefficients. Suppose we have n inputs, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Then $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_P)^T$ in (4) are estimated by minimizing the sum of squares of residuals as follows:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\{ f(\boldsymbol{x}_i) - \sum_{k=0}^{P} \beta_k \Psi_{\boldsymbol{\alpha}_k}(\boldsymbol{x}_i) \right\}^2. \tag{5}$$

The solution is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^T \boldsymbol{y}, \tag{6}$$

where $\boldsymbol{\Psi}$ denotes the $n \times P$ matrix with (i,k)th entry $\Psi_{\boldsymbol{\alpha}_k}(\boldsymbol{x}_i)$, and $\boldsymbol{y} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^T$.

Variable selection is necessary when the number of bases P is large. Note that selecting input parameters is equivalent to selecting bases in the PC expansions. Blatman and Sudret [13] proposed a method based on stepwise regression to fit the PC expansions sequentially. A similar idea using least angle regression was discussed in [18]. However, these methods work only if the number of input parameters is small. For high dimensions, if we consider the PC expansions with order L, the number of candidate bases $P = (p+L)!/p!L!$ increases rapidly, and hence the computation becomes burdensome.

III. Brief review of the sure independence screening and the lasso

In this section, we review two variable selection methods for linear models with high dimensions and moderate dimensions respectively. These methods are embedded in the proposed approach after suitable modifications of the selection of bases. We should point out that the choice is not unique. That is, other variable selection methods [19] [20] [21] that work for linear models can perform the same function in the proposed approach.

Fan and Lv [22] proposed the *Sure Independence Screening* (SIS) on linear models for high dimensional variable selection at a relatively low computational cost. We slightly modify the SIS here to make it applicable on selecting bases, instead of linear predictors. It works as follows. Denote $\Psi_{\boldsymbol{\alpha}_k}(X) = (\Psi_{\boldsymbol{\alpha}_k}(\boldsymbol{x}_1), \ldots, \Psi_{\boldsymbol{\alpha}_k}(\boldsymbol{x}_n))^T$. Compute the correlation between $\Psi_{\boldsymbol{\alpha}_k}(X)$ and $\boldsymbol{y}$, which is given by

$$\omega_k = \mathrm{corr}(\Psi_{\boldsymbol{\alpha}_k}(X), \boldsymbol{y}), \quad \text{for } k = 1, \ldots, P. \tag{7}$$

Then for any given $\gamma \in (0,1)$, the selected model is

$$\mathcal{A}_\gamma = \{1 \le k \le P : |\omega_k| \text{ is among the first } [\gamma n] \text{ largest of all}\}, \tag{8}$$

where $[\gamma n]$ is the integer part of γn. $\gamma n = n - 1$ or $n/\log(n)$ were recommended.

When the dimension is moderate, the lasso [23] is commonly used for variable selection, which can be modified for selecting bases as follows:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\{ f(\boldsymbol{x}_i) - \sum_{k=0}^{P} \beta_k \Psi_{\boldsymbol{\alpha}_k}(\boldsymbol{x}_i) \right\}^2 + \lambda \sum_{k=0}^{P} |\beta_k|, \tag{9}$$

where λ is a tuning parameter that controls the number of selected bases, and usually is determined by cross-validation. The lasso does both variable selection and shrinkage due to the use of the L_1-penalty (last term in (9)). Note that the lasso is more accurate but computationally much heavier than SIS, and hence is more suitable for a moderate number of candidate bases.

IV. Hierarchical screening method connecting UA and SA

Using the PC expansions as the metamodel, we propose a variable selection approach for computational models with many input parameters. Since the method connects UA and SA using one design, we call it *Uncertainty-Sensitivity-Analysis* (or abbreviated as USA). The USA has a hierarchical structure with multiple layers. The following two principles give justification for the structure of the approach.

The effect hierarchy principle and the effect heredity principle [8] are commonly considered in variable selection, in order to reduce computational burden due to a large number of candidate bases. The effect hierarchy principle states that lower order effects are more likely to be important than higher order effects, which gives a reasonable reference for the ranking of importances of bases. By acknowledging the effect hierarchy principle, it becomes possible to arrange the sequence of bases under consideration. The bases that are more likely to be important will be assigned in lower layers, which gives them higher priorities to be tested. The effect heredity principle states that an interaction can be active only if one or all of its parent effects are also active. It can significantly reduce the number of bases to be considered in higher layers, by selecting candidate bases given the selected bases in lower layers. See [24] for a discussion on the principles.

The main idea of the USA is the following. Based on the effect hierarchy principle, the hierarchical structure of candidate bases is built to rank the priorities of the bases with different orders. We do variable selection from lower layers to higher layers. In each layer, depending on the number of candidate bases, the SIS and the lasso are used alternately. Then, the effect heredity principle transfers the selected bases to the next layer, thus connecting any two adjacent layers. The diagram of the USA is sketched in Figure 1.

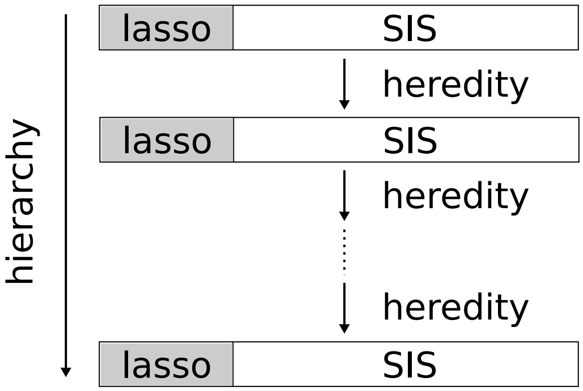


Figure 1. Diagram of the proposed approach.

Let $\mathcal{C} = \{\boldsymbol{\alpha}_k : k = 1, \ldots, P\}$ be the set of the candidate bases. (Note that $k = 0$ is the intercept and is always selected in the model.) Using effect hierarchy, we only consider bases with orders no more than l in the lth layer. Thus, let $\mathcal{M}^{(l)} = \{\boldsymbol{\alpha}_k : k = i_{k_1}, \ldots, i_{k_l};$ and $|\boldsymbol{\alpha}_k| \le l\}$ be the set of selected bases in the lth layer. Based on $\mathcal{M}^{(l-1)}$, we can generate a set of candidate bases $\mathcal{C}^{(l)}$ for the lth layer. The details will be discussed later. Then, the variable selection within the layer works in the following way. Consider the response $\boldsymbol{y}$. We apply the SIS on $\mathcal{C}^{(l)}$ to select a subset of bases $\mathcal{A}_1^{(l)}$ with a moderate size. Apply the lasso on $\mathcal{A}_1^{(l)}$ to obtain the set of selected bases $\mathcal{S}_1^{(l)}$. In the mth step, we have $\mathcal{S}_{m-1}^{(l)}$ selected in the previous steps. The response is changed to the residual $\boldsymbol{r}_{m-1}$, which is a vector with ith entry $f(\boldsymbol{x}_i) - \sum_{\boldsymbol{\alpha}_k \in \mathcal{S}_{m-1}^{(l)}} \hat{\beta}_k \Psi_{\boldsymbol{\alpha}_k}(\boldsymbol{x}_i)$. $\hat{\beta}$ is estimated by (5) and (6) with current selected bases. We then apply the SIS on $\mathcal{C}^{(l)} \setminus \mathcal{S}_{m-1}^{(l)}$ to get $\mathcal{A}_m^{(l)}$. The set of selected bases $\mathcal{S}_m^{(l)}$ is obtained by using the lasso on $\mathcal{S}_{m-1}^{(l)} \cup \mathcal{A}_m^{(l)}$. Keep doing this until some pre-specified stopping rule is satisfied. One reasonable choice is $\mathcal{S}_{m-1}^{(l)} = \mathcal{S}_m^{(l)}$. The set of bases that are selected when it stops is denoted by $\mathcal{S}^{(l)}$.

We connect two adjacent layers by incorporating two versions of the effect heredity principle [8] [25]. For an interaction to be active, at least one of its parents effects is required to be active in *weak heredity*; while all of its parents effects have to be active in *strong heredity*. Generally, there is no guideline on which version should be used in each specific application [24]. In the USA, the two versions are jointly utilized in a natural way for the purpose of simplifying screening without losing much accuracy. The move between two layers consists of two parts. First, at the beginning of each layer, the set of candidate bases $\mathcal{C}^{(l)}$ is generated by expanding $\mathcal{M}^{(l-1)}$ following weak heredity, which is given by

$$\begin{aligned}\mathcal{C}^{(l)} \;=\; & \{\boldsymbol{\alpha}_k : \exists\ \boldsymbol{\alpha}' \in \mathcal{M}^{(l-1)}, \boldsymbol{\alpha}'' \in \mathcal{C} \text{ s.t. } \boldsymbol{\alpha}' + \boldsymbol{\alpha}'' = \boldsymbol{\alpha}_k; \\ & \text{ and } |\boldsymbol{\alpha}_k| \le l\}.\end{aligned} \tag{10}$$

The reason is that it is very likely that some of the parent effects of significant interactions are not significant in lower layers. Second, at the end of each layer, we expand $\mathcal{S}^{(l)}$ following strong heredity. Denote the expanded set by

$$\begin{aligned}\mathcal{D}^{(l)} \;=\; & \{\boldsymbol{\alpha}_k : \exists\ \boldsymbol{\alpha}', \boldsymbol{\alpha}'' \in \mathcal{S}^{(l)} \text{ s.t. } \boldsymbol{\alpha}' + \boldsymbol{\alpha}'' = \boldsymbol{\alpha}_k; \\ & \text{ and } |\boldsymbol{\alpha}_k| \le l\}.\end{aligned} \tag{11}$$

Then use the lasso on $\mathcal{D}^{(l)}$ to screen and fit with the response $\boldsymbol{y}$. By doing so, the interactions between significant bases, which are likely to be significant but have been screened out by a quick variable selection method like the SIS can be re-examined by a more elaborate and accurate variable selection method like the lasso. Denote the set of the final selected bases by $\mathcal{M}^{(l)}$, which will be used to generate $\mathcal{C}^{(l+1)}$ in the next layer. The USA stops when it finishes the highest layer L.

The choice of L is straightforward. Note that the performance of the USA increases as L increases, so does the computational burden. Thus, one can keep increasing the number of layers until enough data is explained by the selected model (which can be assessed by some criterion, such as the coefficient of determination, the root-mean-square error, and so on) or the computational resource runs out. Based on our experience, $L = 3, 4, 5$ are good choices.

The USA is summarized by the following pseudo-code.

```
Set $\mathcal{M}^{(0)} = \varnothing$
for $l = 1, 2, \ldots, L$
    Expand $\mathcal{M}^{(l-1)}$ by (10) to obtain $\mathcal{C}^{(l)}$
    Set $\boldsymbol{r}_0 = \boldsymbol{y}$ and $\mathcal{S}_0^{(l)} = \varnothing$
    for $m = 1, 2, \ldots$
        Set $\boldsymbol{r}_{m-1}$ to be the response
        Apply the SIS on $\mathcal{C}^{(l)} \setminus \mathcal{S}_{m-1}^{(l)}$ to obtain $\mathcal{A}_m^{(l)}$
        Apply the lasso on $\mathcal{S}_{m-1}^{(l)} \cup \mathcal{A}_m^{(l)}$ to obtain $\mathcal{S}_m^{(l)}$
    endfor if converge; denote the last $\mathcal{S}_m^{(l)}$ by $\mathcal{S}^{(l)}$
    Expand $\mathcal{S}^{(l)}$ by (11) to obtain $\mathcal{D}^{(l)}$
    Set $\boldsymbol{y}$ to be the response
    Apply the lasso on $\mathcal{D}^{(l)}$ to obtain $\mathcal{M}^{(l)}$
endfor
```

V. A SIMULATION STUDY

We use the Morris function [11] to study the performance of the USA.

$$\begin{aligned} y &= \beta_0 + \sum_{i=1}^{20} \beta_i w_i + \sum_{i<j}^{20} \beta_{ij} w_i w_j \\ &\quad + \sum_{i<j<l}^{20} \beta_{ijl} w_i w_j w_l + \sum_{i<j<l<s}^{20} \beta_{ijls} w_i w_j w_l w_s, \end{aligned} \tag{12}$$

where

$$w_i = \begin{cases} 2(1.1X_i/(X_i+0.1)-0.5), & \text{for } i = 3,5,7, \\ 2(X_i-0.5), & \text{otherwise,} \end{cases} \tag{13}$$

and $X_i \sim U(0,1)$. The β_i are assigned as

$$\begin{array}{ll} \beta_i = 20, & \text{for } i = 1,\ldots,10, \\ \beta_{ij} = -15, & \text{for } i,j = 1,\ldots,6, \\ \beta_{ijl} = -10, & \text{for } i,j,l = 1,\ldots,5, \\ \beta_{ijls} = 5, & \text{for } i,j,l,s = 1,\ldots,4. \end{array} \tag{14}$$

The remaining first- and second-order coefficients are independently generated from a standard Gaussian distribution; the remaining third- and fourth-order coefficients are zero.

The true model has 20 X_i input parameters, out of which the first ten are significant. 980 dummy parameters were added. So we had $p = 1000$ parameters in total. n simulations were conducted each time. The design matrix was an arbitrary Latin hypercube design with n rows and p columns.

For the SIS, $[\gamma n]$ was set to be $[n/\log(n)] = 81$. For the lasso, the number of selected bases was chosen by cross-validation. We use one simulation with $n = 500$ to illustrate how the USA works. Step-by-step details are given in Table I. For simplicity, only the results in the first and the final iterations in each layer are listed. From Table I, all significant input parameters had been already selected in the third layer. Although we can still try more layers to get better performance of the PC expansions, the PC expansions with order three can explain more than 90% variation of the data [13]. Thus, we chose three as the highest layer L in the example. Another important point is that strong heredity at the end of the second and third layers played a key role in selecting significant bases and removing insignificant bases.

The results of 20 replications with $n = 300, 500, 1000$ are listed in Table II, which show the average values of the numbers of true positives and false positives. Standard deviations are given in parentheses. The sensitivities indices can be computed analytically after all significant input parameters are found. See [13].

TABLE II. RESULTS FOR MORRIS FUNCTION WITH $p = 1000$

n	*True Positives*	*False Positives*
300	7.11(1.66)	9.58(4.05)
500	9.26(0.87)	2.95(2.83)
1000	10(0)	1.36(0.56)

By comparing the results with different n, it is clearly seen that as n increases, the performance became better. With $n = 500$ runs, the USA can identify almost all significant input parameters. The false positives, at the same time, were very low. Moreover, the USA exactly identified all significant variables each time when $n = 1000$, which is still a light computation.

We compare the performance of the USA with that of the Morris method [11]. The Morris method requires at least $r(p+1)$ function evaluations, where r is the number of levels of each input parameter. Based on the results of 20 replications, the Morris method identified 99.5% significant variables when $r = 4$, $n = 4004$. When $r = 2$, $n = 2002$ (the minimal number of runs required), only 75.5% significant variables were identified. In contrast, the USA with $n = 1000$ performed perfectly. Considering the design used in the Morris method is a special design, which cannot be used for neither UA nor SA, the USA is much better than the Morris method in terms of saving functional evaluations.

VI. CONCLUSION

In this paper, we have proposed a hierarchical variable selection dubbed the USA, for computational models with many input parameters. The major feature of the USA is that it unifies UA and SA with the use of one design, which significantly reduces the number of simulations required. By incorporating the effect hierarchy principle and the effect heredity principle, the number of candidate bases in each layer is reduced to an acceptable level for variable selection. By applying the SIS and the lasso alternately, the USA balances the computation and the performance. The performance was shown with a numerical example.

As we mentioned in Section II, the coefficients in the PC expansions can be estimated in several ways. Although we only consider the regression method, the USA is independent of the method for estimating coefficients, and hence can be extended to other methods, which is left for future research.

(Note: An expanded version of this paper, including an application to building energy simulation, will be forthcoming as a joint work with Yuming Sun and Godfried Augenbroe.)

ACKNOWLEDGMENT

This work is supported by NSF-EFRI-SEED Award 1038248 and DOE grant DE-SC 0010548.

REFERENCES

[1] J. C. Helton and F. J. Davis, “Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems,” Reliability Engineering & System Safety, vol. 81, no. 1, 2003, pp. 23–69.

[2] R. Cukier, C. Fortuin, K. E. Shuler, A. Petschek, and J. Schaibly, “Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. i theory,” The Journal of Chemical Physics, vol. 59, no. 8, 1973, pp. 3873–3878.

[3] I. M. Sobol’, “Sensitivity estimates for nonlinear mathematical models,” Mathematical Modeling and Computational Experiment, vol. 1, no. 4, 1993, pp. 407–414.

[4] A. Saltelli, K. Chan, and E. M. Scott, Sensitivity analysis. Wiley New York, 2000, vol. 134.

[5] J. E. Oakley and A. O’Hagan, “Probabilistic sensitivity analysis of complex models: a bayesian approach,” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 66, no. 3, 2004, pp. 751–769.

[6] B. Sudret, “Global sensitivity analysis using polynomial chaos expansions,” Reliability Engineering & System Safety, vol. 93, no. 7, 2008, pp. 964–979.

[7] S. Touzani and D. Busby, “Smoothing spline analysis of variance approach for global sensitivity analysis of computer codes,” Reliability Engineering & System Safety, vol. 112, 2013, pp. 67–81.

[8] C. F. J. Wu and M. S. Hamada, Experiments: planning, analysis, and optimization. John Wiley & Sons, 2009, vol. 552.

TABLE I. ILLUSTRATION OF THE USA WITH $n = 500$ AND $p = 1000$

Layer	*Stage*	*Bases*
1	Weak heredity	N/A; $\mathcal{C}^{(1)} = 1000$ bases
1	First: SIS	$\mathcal{A}_1^{(1)} = 81$ bases
1	First: Lasso	$\mathcal{S}_1^{(1)} = 19$ bases: $3, 5, 7, 8, 9, 10,$ $93, 203, 284, 286, 362, 434, 510, 511, 623, 815, 896, 940, 941$
1	Last: SIS, Lasso	$\mathcal{S}^{(1)} = 17$ bases: $3, 5, 7, 8, 9, 10,$ $93, 203, 284, 286, 362, 434, 510, 511, 556, 896, 966$
1	Strong heredity	N/A; $\mathcal{M}^{(1)} = \mathcal{S}^{(1)}$
2	Weak heredity	$\mathcal{C}^{(2)} = 16711$ bases
2	First: SIS	$\mathcal{A}_1^{(2)} = 81$ bases
2	First: Lasso	$\mathcal{S}_1^{(2)} = 23$ bases: $5, 7, 8, 9, 10, 286, 1\&3, 1\&5, 2\&5, 3\&4, 3\&6, 4\&6$ $161\&284, 163\&556, 167\&896, 198\&284, 198\&544, 238\&284,$ $284\&858, 284\&885, 510\&592, 510\&966, 673\&792$
2	Last: SIS, Lasso	$\mathcal{S}^{(2)} = 19$ bases: $5, 7, 8, 9, 10, 286, 1\&3, 1\&5, 2\&5, 3\&4, 3\&6,$ $93\&108, 161\&284, 163\&556, 167\&896,$ $198\&284, 238\&284, 510\&592, 510\&869$
2	Strong heredity	$\mathcal{M}^{(2)} = 16$ bases: $7, 8, 9, 10, 1\&2, 1\&3, 1\&5, 1\&6,$ $2\&4, 2\&5, 2\&6, 3\&4, 3\&6, 4\&6, 7\&7, 163\&556$
3	Weak heredity	$\mathcal{C}^{(3)} = 15744$ bases
3	First: SIS	$\mathcal{A}_1^{(3)} = 81$ bases
3	First: Lasso	$\mathcal{S}_1^{(3)} = 21$ bases: $7, 8, 9, 10,$ $1\&2, 1\&3, 1\&4, 1\&5, 1\&6, 2\&4, 2\&5, 2\&6, 3\&4,$ $3\&6, 4\&5, 4\&6, 5\&6, 7\&7, 3\&4\&421, 5\&574, 7\&32$
3	Last: SIS, Lasso	$\mathcal{S}^{(3)} = 16$ bases: $7, 8, 9, 10,$ $1\&2, 1\&3, 1\&4, 1\&5, 1\&6, 2\&4, 2\&5, 2\&6, 3\&4, 3\&6, 4\&6, 7\&7$
3	Strong heredity	$\mathcal{M}^{(3)} = 20$ bases: $7, 8, 9, 10,$ $1\&2, 1\&3, 1\&4, 1\&5, 1\&6, 2\&4, 2\&5, 2\&6, 3\&4,$ $3\&6, 4\&6, 7\&7, 3\&4\&5, 3\&5\&5, 7\&7\&7, 10\&10\&10$

[9] S. C. Cotter, “A screening design for factorial experiments with interactions,” Biometrika, vol. 66, no. 2, 1979, pp. 317–320.

[10] B. Bettonvil, Detection of important factors by sequential bifurcation. Tilburg University Press, 1990.

[11] M. D. Morris, “Factorial sampling plans for preliminary computational experiments,” Technometrics, vol. 33, no. 2, 1991, pp. 161–174.

[12] M. D. McKay, R. J. Beckman, and W. J. Conover, “Comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” Technometrics, vol. 21, no. 2, 1979, pp. 239–245.

[13] G. Blatman and B. Sudret, “Efficient computation of global sensitivity indices using sparse polynomial chaos expansions,” Reliability Engineering & System Safety, vol. 95, no. 11, 2010, pp. 1216–1229.

[14] D. Xiu and G. E. Karniadakis, “The wiener–askey polynomial chaos for stochastic differential equations,” SIAM Journal on Scientific Computing, vol. 24, no. 2, 2002, pp. 619–644.

[15] C. Soize and R. Ghanem, “Physical systems with random uncertainties: chaos representations with arbitrary probability measure,” SIAM Journal on Scientific Computing, vol. 26, no. 2, 2004, pp. 395–410.

[16] R. G. Ghanem and P. D. Spanos, Stochastic finite elements: a spectral approach. Springer, 1991, vol. 41.

[17] T. Crestaux, O. Le Maître, and J.-M. Martinez, “Polynomial chaos expansion for sensitivity analysis,” Reliability Engineering & System Safety, vol. 94, no. 7, 2009, pp. 1161–1172.

[18] G. Blatman and B. Sudret, “Adaptive sparse polynomial chaos expansion based on least angle regression,” Journal of Computational Physics, vol. 230, no. 6, 2011, pp. 2345–2367.

[19] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” Journal of the American Statistical Association, vol. 96, no. 456, 2001, pp. 1348–1360.

[20] H. Zou, “The adaptive lasso and its oracle properties,” Journal of the American Statistical Association, vol. 101, no. 476, 2006, pp. 1418–1429.

[21] E. Candes and T. Tao, “The dantzig selector: statistical estimation when p is much larger than n,” The Annals of Statistics, 2007, pp. 2313–2351.

[22] J. Fan and J. Lv, “Sure independence screening for ultrahigh dimensional feature space,” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 70, no. 5, 2008, pp. 849–911.

[23] R. Tibshirani, “Regression shrinkage and selection via the lasso,” Journal of the Royal Statistical Society: Series B (Methodological), vol. 58, no. 1, 1996, pp. 267–288.

[24] M. Yuan, V. R. Joseph, and Y. Lin, “An efficient variable selection approach for analyzing designed experiments,” Technometrics, vol. 49, no. 4, 2007, pp. 430–439.

[25] H. Chipman, “Bayesian variable selection with related predictors,” Canadian Journal of Statistics, vol. 24, no. 1, 1996, pp. 17–36.

An Agent-based Model to Support Measuring Drug Choice and Switch Between Drug Types in Rural Populations

Georgiy Bobashev
RTI International
Research Triangle Park, NC
Bobashev@rti.org

Eric Solano
RTI International
Research Triangle Park, NC
Solano@rti.org

Lee Hoffer
Case Western Reserve University
Cleveland, OH
lee.hoffer@case.edu

***Abstract*—In rural areas availability, price, and legal consequences can force some drug users to switch between primary drugs of choice. For example, as a consequence of stricter law enforcement policy in rural Ohio we observed a shift from methamphetamine use to heroin and prescription opiate use. We propose a polydrug agent-based model that describes drug users interconnected in a network. Behavior rules are based on our ethnographic research. The drug selection mechanisms are dictated by drug liking, drug availability, drug cost, perception of health and other life consequences, perception of potential punishment and pressure from the peers. The model produces time series of users' choices of one or concurrent drugs. Modeling and ethnographic data collection are interlinked i.e. model results lead to the improvements in quantitative measurements, which in turn improve the model. Polydrug trends are of particular interest to policy makers because short-term interventions can lead to long-term adaptation.**

Keywords- decision making, drug use modeling, drug switch, agent-based model, model-based survey

I. Introduction

National and regional drug use patterns have been changing, reflecting changes in individual choices of drugs, for example, marijuana in the late 1960s and 1990s, heroin epidemic waves of the early 1950s and 1970s (1971–1977), the crack epidemic in the late 1980s, and the methamphetamine epidemic in late 2000s [1]. These trends are difficult to predict because they reflect several adaptive factors including individual choices, public policy, public acceptance, and the adaptation of the supply side of the market. Despite the “war on drugs,” drug use patterns and trends surprisingly persist [2]. Individual choices and drug trajectories have been extensively described by a number of researchers [3,4,5]. On the other hand, market adaptation has been extensively studied from a cost-effectiveness point of view [6]. However, these two components are seldom put together with a few exceptions [7,8]. Understanding is a critical component in policy and decision making. By “understanding,” here, we mean the identification of causal patterns and feedbacks that could predict qualitatively or quantitatively the response resulting from the actions taken. Such understanding is complicated by a lack of available data and a lack of experience in collecting the right data. In this sense, to develop the right understanding, modeling and data collection should complement each other in an iterative manner. Data collection about illegal and adaptive behavior could benefit from model-based suggestions about which aspects of behavior to include as data items, and conversely, the collected data should educate the model about unexplained events and possible adaptations. In our NIDA-funded study we employ this approach. Based on initial ethnographic information we have developed a behavioral model which is now providing suggestions about specific aspects of ethnographic information that needs to be collected to further develop the model. The model is theoretical, meaning that the goal is to formally reflect our qualitative “understanding” of rural drug use patterns. The model is not supposed to reproduce numerically the observed data but rather identify qualitative trajectories of individual and collective response to external interventions.

Our agent-based model describes individual drug choices under the conditions of drug availability, perceived risk of drug use, impact of social network, and drug use “burnout.” Although there are a few models that describe the process of drug use and operation of drug markets [4,6,10,12], there was little focus on modeling the switch between drug use. We considered two models: a simplified model focused on the specific phenomenon of switching between methamphetamine and opiates and an extended model that considers a greater variety of drug choices as well as the evolution of the supply side of the market. In this paper we focus on the simplified model because it serves the specific purpose of the study and provides a basis for the extended model under development. In the next section we describe the model's assumptions and the basic rules that are followed by drug using and market agents. Then we present our preliminary results and discuss the implication for data collection, policy analysis, and future work.

II. Methods

To provide suggestions about specific aspects of ethnographic information that needs to be collected for understanding switches between markets we developed a theoretical agent-based model that describes our understanding of market functioning. The model is built using the Overview Design concepts and Details (ODD) protocol that was introduced and standardized by Railsback and Grimm [9]. The ODD is designed to demystify ABMs and provide the reader with a clear description of the logic, structure, and components of the model. The rest of the section is structured according to this protocol.

A. *Purpose*

The purpose of this model is to describe the manifold drug use patterns leading to community response to drug use interventions. The model will help to understand how drug preference, drug-acquiring effort levels, drug use consequences, and drug-sticking factors in a networked population affect the decision to switch the use of one drug for another. Drug preference identifies how much an individual likes a specific drug at a specific time.

The model translates ethnographic observations and narratives into formal causal rules and parameter values. The model can then be used to generate simpler aggregate system dynamics and statistical models that in turn could be calibrated and validated to “hard” survey data.

B. *Entities, State Variables, and Scales*

The model includes individuals who are drug users. There is a constant population of drug users during the entire simulation.

The simpler model contains only three reinforcers: Heroin, Opiates, and Other. Note that “Other” could include other drugs such as alcohol and marijuana, but also non-drug reinforcers such as sporting activities. For simplicity we separate methamphetamine and opiates from the other reinforcers.

The agents in the model use one, two, or all three reinforcers. We do not distinguish between occasional and regular use; however, we distinguish between never use, first use, and use of the reinforcer. The reason for such distinction is that the effect of the drug on individual perception of liking can change after first use and then after consequent use. Thus, in relation to each of the drugs an agent could be in one of the following non-overlapping states: Never used, tried it, use it, and used in the past. The extended version of the model also considers occasional and regular use (Figure 1).

Each individual can be in the community or institutionalized (incarcerated or in treatment). Each drug state is characterized by attitudes toward the drug (i.e., drug liking) and a number of external factors. The transition between the stages is additionally governed by external factors such as peer pressure.

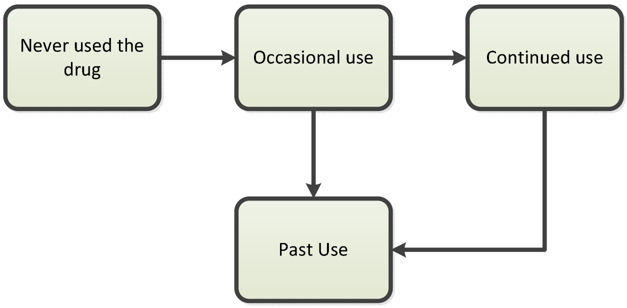


Figure 1. State chart for an agent in the multiple drug user model

The population of drug users is interconnected. Drug users are linked to a number of other drug users during the simulation. The linkage between two drug users is maintained during the entire simulation.

C. *Process Overview and Scheduling*

This model proceeds in monthly time steps. Within each month or time step, four phases are processed in the following order: drug selection probability calculations, selected drugs identification, drug use state update, and drug preference update.

At each time step an individual considers a number of competing priorities, and evaluates them according a combination of six drug factors:

- Drug liking
- Drug availability
- Drug cost
- Perception of health and other life consequences
- Perception of potential punishment
- Pressure from peers

These factors can change in time according to external or internal factors. For example, external influences can be related to drug availability, activities of drug cartels, or the perception of punishment, which could increase due to formation of a police task force to fight a specific drug. Internal influences are associated with the increase of “drug liking” after prolonged use. We also consider that in time some drugs could lead to “burnout” and decrease in liking. Although we use the term “drug liking” for the description of both positive and negative reinforcement, we implicitly distinguish between them by assuming that occasional use is a result of positive reinforcement and dependence is a result of negative reinforcement. The resource can be stable over time or get reduced because of loss of a job caused by addiction. Thus, a number of complex feedback loops impact the formation and development of individual and community drug use patterns in the drug market environment.

Agents can get institutionalized depending on perceived punishment. Although the punishment perception should follow the severity of the punishment we assume that these factors are immediately correlated and controlled externally by an observer or external input.

Timeline. We are more focused on long-term changes rather on daily behavior. The time horizon in the simulation is 10 years. The time steps are of 1-month duration.

Scheduling. At each timestep each agent first evaluates the external and internal factors and then makes a decision about which drugs to use for the next timestep. Among external factors we consider the prevalence of drug use among agents' peers. In a simplest implementation we assume that the community is closed and everyone is connected to everyone, so rather than considering a complex social network we average the drug use in the community to create the "peer pressure" factor. In a more realistic setting we consider two overlapping networks: heroin and methamphetamine. The assessment of the network is done before any individual state changes.

D. *Design Concepts*

The model design assumes that each person is interconnected with a number of people in the population. The model hypotheses are that changes of use from one type of drug to another depends on drug preference, drug-acquiring efforts levels, drug use consequences, and drug-sticking factors.

The model design assumes that drug supply is a commodity that is always available. Drug suppliers are different for different drugs. Drug preference changes over time for each type of drug. If a person has null preference for a drug, the only way to change it to a positive preference is through social pressure exerted by the person's network.

Emergence. We do not consider explicit emergent behavior, but rather expect that emergent behavior could form on its own from existing rules. We might expect to see that the impact in one limiting parameter such as increase in punishment for a single drug without providing alternative reinforcement option for "no use" could lead to the increase in other drugs becoming dominating depending on the punishment and availability of the new drug. Thus, with the delay in response we might expect long-term waves in drug use that have been historically observed in the United States and worldwide.

Agent objectives. Agents have their resources and are willing to spend them depending on the current assessment of the situation. They do not have any long-term objectives, but rather make a choice according to the criteria based on a combination of expected effect and risks.

In real life individuals have resources such as time and funds that they spend on their priorities. The resources act as limiting factors that control the mix of drugs an individual can use. By considering other reinforcers we cover the entire space of choice possibilities that compete for these resources. Rather than have an explicit set of resources we consider a relative score which is defined as a scaled absolute score and the agent makes choices based on the values of that relative score. For each choice we define an absolute score Xi as follows:

$$X_i = \frac{Liking * Availability * PeerInfluence}{(1+Price)(1+HealthProb)(1+Punishment)}, \qquad (1)$$

where each of the components is defined on a scale from 0 to 100. The relative score is defined as $S_i = \frac{X_i}{\sum_i X_i}$. We also provide the rule for making the choice to try and to use the drug. If for a specific drug the score S_i>*T1*, where *T1* is an upper threshold, then the individual will become a user of that choice. If the relative score is between *T1* and *T2*, where *T2* is a lower threshold, then the user will try the reinforcer, and if the score is less than *T2* then the user will not be using that drug. The natural low limit for *T2* is 1/3 given that we have 3 choices of reinforcers. This will guarantee that at least one choice will be made. The example of the relative score is presented in Figure 2.

Adaptation. Agents assess their social network and include the assessment in the objective function.

Learning. Agents' learning ability is represented in the form of drug-liking dynamics. The more an agent uses the drug the more it "likes" the drug. We use the term "liking" as a surrogate of the actual liking and dependence. Although one of the characteristics of drug addiction is negative reinforcement (i.e., an individual uses a drug to reduce discomfort) we consider that limited to the dynamics of the drug-liking function (i.e., they can move from occasional to regular users). Future model versions will have the affinity to their drug using group as a part of generating "comfort" as an additional factor impacting the choice of drug.

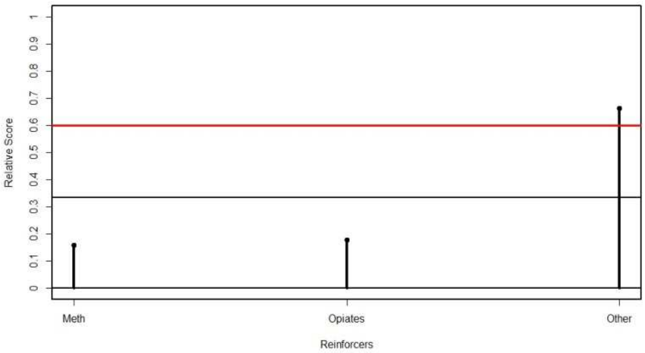


Figure 2. Example of the relative score diagram. "Other" reinforcement shows the highest score thus this agent will choose the "Other" reinforcement.

We distinguish growth in drug use liking after first use and tempering out with time reflecting the "burn out" in drug use. This is done by multiplying the liking by the burnout function which has a logistic shape with characteristic time. The functional form for the growth and burnout in liking is represented with a function with interpretable coefficients:

$$L_i = \frac{at^b}{(1+at^b)} \frac{1}{\left(1+ce^{\left(d(t-t')\right)}\right)}, \qquad (2)$$

where t is the time since the first try, a, b, c, d and t' are scaling parameters with a corresponding to the speed of initial growth and is used to distinguish between occasional and regular use. Parameter b is characteristic of different drugs, parameter c corresponds to the end point for the liking, d corresponds to the speed at which the liking is reduced, and t' to the timing of the decline in liking. The largest possible liking corresponds to the value of 100.

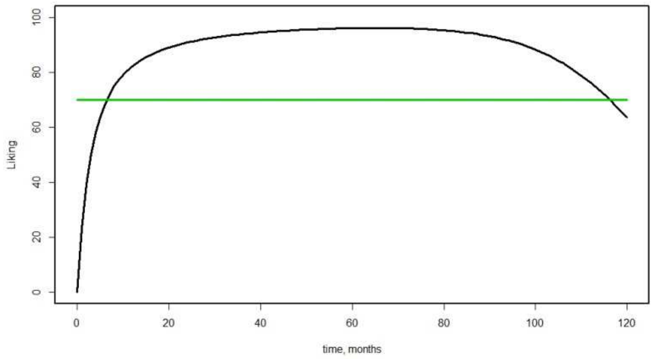


Figure 3. Liking function for heroin.

In Figure 3 we present the shapes of the liking function for a drug such as heroin as it progresses in time. It takes a short period of use to get addicted but after a long period of use the liking is lowered because of the detraining health and lifestyle effects. The characteristic times of the liking dynamics could be defined as times when the liking reaches 70 of the maximum value. In Figure 3 these times will correspond to points where the liking function crosses the horizontal (green) line.

Prediction. The agents do not make any prediction. All prediction is described by an immediate assessment of health and punishment risks.

Sensing. Agents sense the state of the members of their network (currently everyone).

Interaction. Agents do not directly interact with each other, they only sense the state of network members.

Stochasticity. In the current version agents are deterministic by choosing the "best" combination of drugs; however, there is an option to randomly make the stochastic choice with probability of choosing being proportional to the relative score of the drug choice. The stochastic component is introduced at the level of creating connections in a social network. The potential usage of a drug is a binary variable that depends on the individual's social index. The social index is computed as a function of the individual's previous drug use and the previous drug use of the individual's peers. The potential use is then computed using a Bernoulli distribution with social index as the parameter. Initial drug preference values and initial drug use states are randomly selected from a set of four possible states, each with a given probability.

Collectives. Collectives are defined according to the type of reinforcer they use. Thus, an individual could be in more than one collective at a time.

Observation. The observer has options to impact drug availability and perceived drug punishment. At each step we record the individual's drugs of use which are recorded in the output file and then can be analyzed.

Initialization. A randomly linked network is initially created and used for each simulation. Each drug user is initialized by assigning a random number of linked people and the values of drug preferences and drug characteristics. A number of other structured networks has been considered, e.g. scale-free and a small world, but actual validation of the network structure is not feasible at this point, thus we use the simplest random arrangement.

III. Model Applications

The model was developed as a standalone Java application as well as a NetLogo model. The models were verified through a system of sequential tests: (1) a collection of individual users, no market agent (drug characteristics do not change), no networks; (2) addition of networks, no market agent; and (3) network and agents. Althought the direct validation of the model has not been feasible becase of the lack of direct measurements in the real world setting, the face validity of the resutls were discussed with the ethmographers and law enforcement respresentatives.

The main application of this model resulted in the design of a set of six questions aimed to calibrate the model. These questions corresponded to the items in the score and were defined at a 10-scale level. This questionnaire was included into a longer ongoing ethnographic survey of drug users. Upon the completion of the study the data will be used for model calibration and validation on historic data.

Here we present the results of a hypothetical scenario. After the market locally stabilizes, the action of law enforcement leads to the increase in consequences of the local methamphetamine market, which in turn leads to the reduction in meth use and increase in other reinforcements available in the social network. Because heroin has a higher addictive potential than "other" reinforces it eventually wins over the community with a temporary dominance of other reinforcements. Figure 4 shows the results of running the simulation for 2 years.

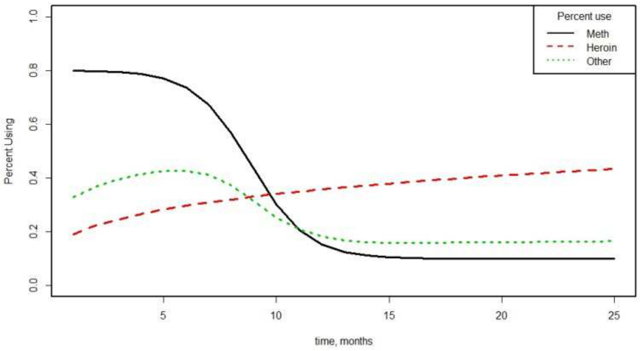


Figure 4. Drug use trends corresponding to the scenario where law enforcement activities led to the reduction in methamphetamine use and the increase in heroin use.

After users try heroin they get quickly addicted and by the time methamphetamine is reintroduced to the community many former meth users are on heroin. Addition of methamphetamine increases the chance of using both.

IV. Discussion

We present a model that describes individual priorities in the choice of drug and the relationship between the individual choices, drug supply adaptation and the impact of community and law enforcement which sometimes leads to short time success followed by long-term resistance and opposite results. Our model provides a possible explanation on why law enforcement strategies often fail. When the law enforcement attempts don't consider long-term market adaptation and with limited resources the success is quickly diminished by the adaptation in a different dimension. These adaptations sometimes make drug markets more efficient as in [10] or dominated by a more addicted drug as in the case of Summit County considered in this paper. Historical precedents support the need for a systems approach. For example, the increase in border control between the United States and Mexico in 1969 (Operation Intercept) led to an increase in the use of other drugs (76% of students and 84% percent of patients reported that they increased their consumption of one or more other drugs (including alcohol) because of the unavailability of marijuana) [11]. Additionally, smugglers have found more efficient ways to smuggle the drug into the United States and domestic marijuana growth has increased, which made the illegal market more efficient [11].

Although our paper presents a simple theoretical model, ethnographic research can play a key role in a realistic understanding of feedback processes and consequences of interventions. Ethnographic research allows one to collect data on the actual reasoning and causal relationships that are often hidden from standardized surveys. Although limited to a smaller number of individuals, ethnographic data provide the basis for the development of theoretical mental models. The process of building ABMs allows one to convert these mental models into formal rules and parameters. By simulating trajectories from these ABMs one can obtain understanding of the main consequences and adaptations one can expect from the interventions. Although more difficult to calibrate and validate microsimulation have an advantage of capturing non-linearities that occur due to complex decision making and networks structures. As shown in [13] a simplification of an ABM by aggregating over the population agents and time requires careful analysis, otherwise a simplified aggregated model will not adequately represent the dynamics of the full model. After selecting the best theoretical scenarios ABMs can provide the basis for population-based data collection through more reliable means such as representative surveys. The advantage of such an approach is that now the surveys are grounded in realistic causal pathways and thus their predictive validity becomes much more prominent.

ACKNOWLEDGMENT

This research was funded by a grant from the National Institutes of Health, National Institute on Drug Abuse (DA025163)

REFERENCES

[1] Hunt, D., S. Kuck, and L. Truitt, (2006) Methamphetamine Use: Lessons Learned, final report to the National Institute of Justice, (NCJ 209730), available at www.ncjrs.gov/pdffiles1/nij/grants/209730.pdf.

[2] Global commission on War on drugs National report June 02 2011. Available online at: http://www.scribd.com/doc/56924096/Global-Commission-Report

[3] Anglin, M.D., et al., (2001). Drug treatment careers: Conceptual overview and clinical, research, and policy applications. In: F. Tims, C. Leukefeld, and J. Platt (Eds.), Relapse and Recovery in Addictions. New Haven, CT: Yale University Press, pp. 18-39.

[4] Agar M. (2005) Agents in Living Color: Towards Emic Agent-Based Models. *Journal of Artificial Societies and Social Simulation* vol. 8, no. 1

[5] Anderson T. (1998) A cultural-identity theory of drug abuse. Sociology of Crime, Law, and Deviance, Volume 1, pp 233-262.

[6] Caulkins, J P. and MacCoun R. (2005), Analyzing Illicit Drug Markets When Dealers Act with Limited Rationality. In Francesco Parisi and Vernon L. Smith (eds.) The Law and Economics of Irrational Behavior, Standard University Press, Stanford, CA, pp.315-338.

[7] Perez P., Dray A., Moore D., Dietze P., Bammer G., Jenkinson R., Siokou C., Green R, Hudson S.L., Maher L., (2011) SimAmph: an agent-based simulation model for exploring the use of psychostimulants and related harm amongst young Australians. The International journal on drug policy. 06/2011; 23(1):62-71

[8] Hoffer, L. (2006) Junkie Business: The Evolution and Operation of a Heroin Dealing Network, Belmont, CA: Thompson Wadsworth.

[9] Steven F. Railsback & Volker Grimm (2012) Agent-Based and Individual-Based Modeling: A Practical Introduction Princton University Press.

[10] Hoffer, L. D., Bobashev, G. V., & Morris, R. J. (2011). Simulating patterns of heroin addiction within the social context of a local heroin market. *In Computational neuroscience of drug addiction* (pp. 313–331). New York: Springer.

[11] McGlothlin W., K. Jamison, and S. Rosenblatt, (1970) Marijuana and the Use of Other Drugs, *Nature* 228: 1227-1229

[12] Lamy F., Bossomaier T., PerezP., SimUse: modeling recreational polydrug use through an agent-based model *Proceedings of the 12th International Conference on Modeling and Applied Simulation*, Athens, Greece 2013: 61-69

[13] Heard D., Morris J., Bobashev G.V., (2014) Reducing the Complexity of an Agent-Based Local Heroin Market Model, *PLoS One.* DOI: 10.1371/journal.pone.0102263