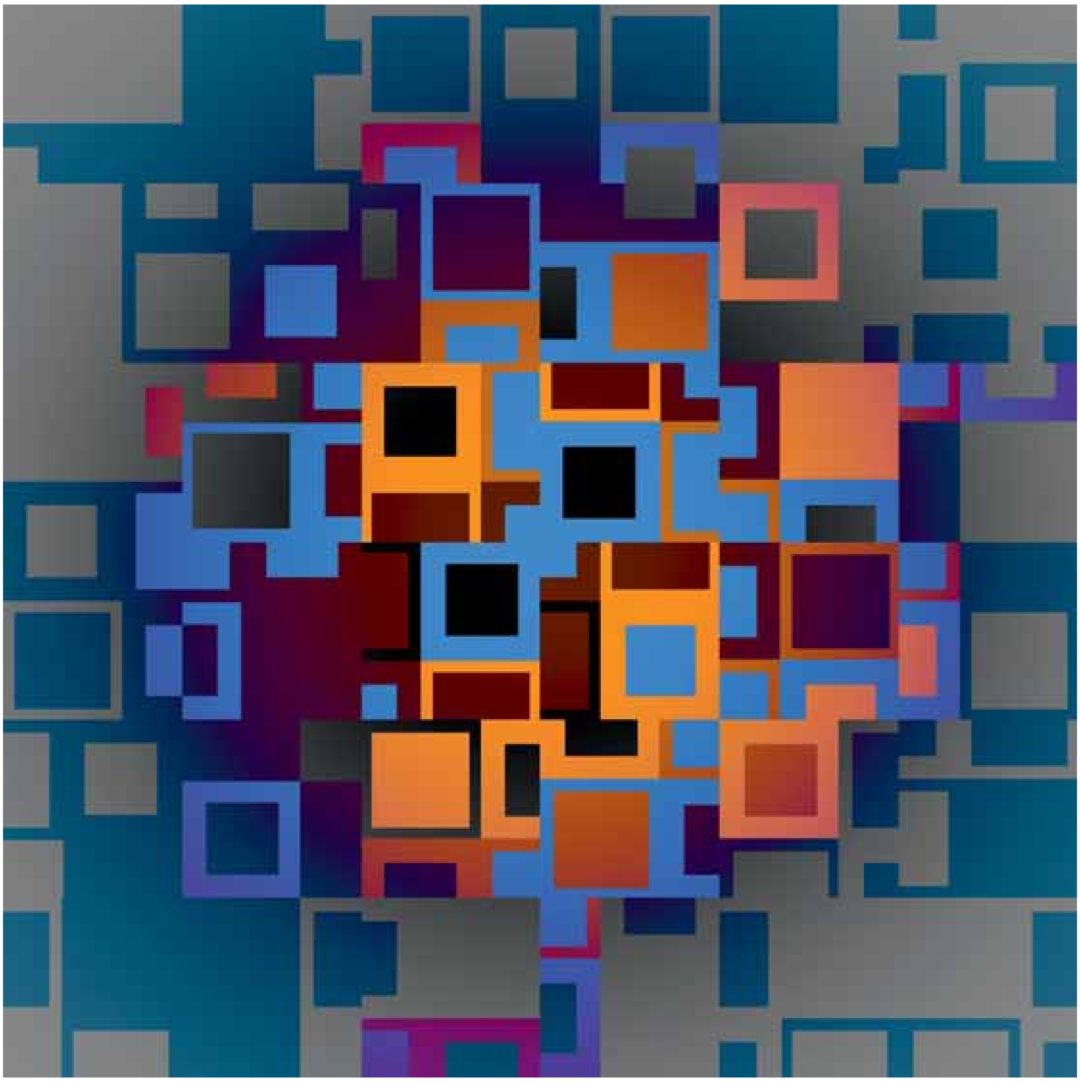
SNE SIMULATION NOTES EUROPE



Journal on Developments and Trends in Modelling and Simulation

EUROSIM Scientific Membership Journal

Vol. 28 No.2, June 2018

ISSN Online 2306-0271
ISSN Print 2305-9974

DOI 10.11128/sne.28.2.1041
ISBN Print 978-3-903024-71-4

TU Verlag an der Technischen Universität Wien

ARGESIM

ASIM ASIM ASIM

ASIM - Buchreihen / ASIM Book Series

Proceedings

Simulation in Production and Logistics 2017 – 17. ASIM Fachtagung Simulation in Produktion und Logistik
Sigrid Wenzel, Tim Peter (Hrsg.); kassel university press GmbH, Kassel, 2017; ISBN 978-3-7376-0192-4 (print), ISBN 978-3-7376-0193-1 (e-book); ASIM Mitteilung AM164.

Simulation in Production und Logistics 2015 - 16. ASIM-Fachtagung Simulation in Produktion und Logistik
M. Raabe, U. Clausen (Hrsg.); ISBN 978-3-8396-0936-1, Stuttgart: Fraunhofer Verlag, 2015.

Simulation in Produktion und Logistik 2013: Entscheidungsunterstützung von der Planung bis zur Steuerung
W. Dangelmaier, C. Laroque, A. Klaas (Hrsg.); ISBN 978-3-942647-35-9, HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut, Paderborn, 2013.

Modellierung, Regelung und Simulation in Automotive und Prozessautomation – Proc. 5. ASIM-Workshop Wismar 2011. C. Deatcu, P. Dünow, T. Pawletta, S. Pawletta (eds.), ISBN 978-3-901608-36-0, ASIM/ARGESIM, Wien, 2011.

Simulation in Produktion und Logistik 2010: Integrationsaspekte der Simulation - Technik, Organisation und Personal. G. Zülch, P. Stock, (Hrsg.), ISBN 978-3-86644-558-1, KIT Scientific Publ. Karlsruhe, 2010.

Monographs

Simulation und Optimierung in Produktion und Logistik – Praxisorientierter Leitfaden mit Fallbeispielen.
L. März, W. Krug, O. Rose, G. Weigert , G. (Hrsg.); ISBN 978-3-642-14535-3, Springer, 2011.

Verifikation und Validierung für die Simulation in Produktion und Logistik - Vorgehensmodelle und Techniken.
M. Rabe, S. Spieckermann, S. Wenzel (eds.); ISBN: 978-3-540-35281-5, Springer, Berlin, 2008.

Qualitätskriterien für die Simulation in Produktion und Logistik – Planung und Durchführung von Simulationsstudien. S. Wenzel, M. Weiß, S. Collisi – Böhmer, H. Pitsch, O. Rose (Hrsg.); ISBN: 978-3-540-35281-5, Springer, Berlin, 2008.

Series Fortschrittsberichte Simulation

Patrick Einzinger: **A Comparative Analysis of System Dynamics and Agent-Based Modelling for Health Care Reimbursement Systems.** FBS 24; ISBN 978-3-901608-75-9, ASIM/ARGESIM Vienna, 2014; www.tuverlag.at – print-on-demand.

Martin Bruckner: **Agentenbasierte Simulation von Personenströmen mit unterschiedlichen Charakteristiken.** FBS 23; ISBN 978-3-901608-74-2 (ebook), ASIM/ARGESIM Vienna, 2014; www.tuverlag.at – print-on-demand

Stefan Emrich: **Deployment of Mathematical Simulation Models for Space Management.** FBS 22; ISBN 978-3-901608-73-5 (ebook), ASIM/ARGESIM Vienna, 2013; www.tuverlag.at – print-on-demand

Xenia Descovich: **Lattice Boltzmann Modeling and Simulation of Incompressible Flows in Distensible Tubes for Applications in Hemodynamics.** FBS 21; ISBN 978-3-901608-71-1 (ebook), ASIM/ARGESIM Vienna, 2012; www.tuverlag.at – print-on-demand

Florian Miksch: **Mathematical Modeling for New Insights into Epidemics by Herd Immunity and Serotype Shift.** FBS 20, ISBN 978-3-901608-70-4 (ebook), ASIM/ARGESIM Vienna, 2012; www.tuverlag.at – print-on-demand

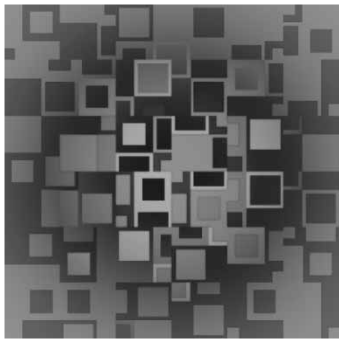
Shabnam Tauböck: **Integration of Agent Based Modelling in DEVS for Utilisation Analysis: The MoreSpace Project at TU Vienna.** FBS 19; ISBN 978-3-901608-69-8 (ebook), ASIM/ARGESIM Vienna, 2012; www.tuverlag.at – print-on-demand

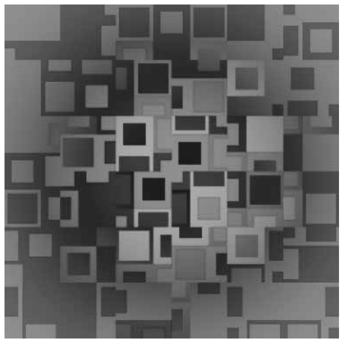
Christian Steinbrecher: **Ein Beitrag zur prädiktiven Regelung verbrennungsmotorischer Prozesse.**
FBS 18; ISBN 978-3-901608-68-1 (print), 978-3-901608-72-8 (ebook), ASIM/ARGESIM Vienna, 2010.

Olaf Hagendorf: **Simulation-based Parameter and Structure Optimisation of Discrete Event Systems**
FBS 17; ISBN 978-3-901608-67-4 (print), 978-3-901608-99-5 (ebook), ASIM/ARGESIM Vienna, 2010.

Download (some books) via ASIM webpage - **www.asim-gi.org** **Info:** info@asim-gi.org

Editorial





Dear Readers – This issue SNE 28(2) presents very interesting contributions, for instance the first Project Note contribution, and a Benchmark Note contribution of new style.

SNE Volume 28 started offering the Project Note contribution for publication of project work, either as summary, or as report for (parts of) results, or as sketch of ongoing work, also aiming for discussion and feedback. Back in 1990, SNE started with publication of one- (and later two-) page solutions of the ARGESIM Benchmarks. In 2017, the publication possibilities for benchmark solutions have been extended. Depending on the benchmark, three types of 'solutions' can be submitted: i) a Benchmark Solution with concise description of model implementation and experimentation tasks (2 pages), ii) a Benchmark Report with sufficient detailed description of model implementation with variants and adequate experiment formulations (4 - 6 pages), and iii) a Benchmark Study presenting e.g. alternative or comparative modelling approaches and sketching analysis variants or supplemental model experiments (6 – 10 pages).

This issue also continues with Vlatko Čerić's algorithmic art as design for SNE cover pages. At right the cover pictures for SNE 28(1) and SNE 28(2) from the series BIRTH - the technique used is mapping and overlapping of squares emerging from a centre and fading out in colour.

I would like to thank all authors for their contributions to SNE 28(2) showing the broad variety of simulation. And thanks to the editorial board members for review and support, and to the organizers of the EUROSIM conferences for co-operation in post-conference contributions. And last but not least thanks to the SNE Editorial Office for layout, typesetting, preparations for printing, with special thanks for support co-operation with Vlatko Čerić.

Felix Breitenecker, SNE Editor-in-Chief, eic@sne-journal.org; felix.breitenecker@tuwien.ac.at

Contents SNE 28(2)

Online SNE 28(2), DOI 10.11128/sne.28.2.1041, ARGESIM Publ.
Print SNE 28(2) ISBN 978-3-903024-71-4,
Print-on-Demand, TU Verlag Vienna, www.tuverlag.at

SNE Contact & Info

→ *www.sne-journal.org*
office@sne-journal.org, eic@sne-journal.org
SNE Editorial Office, Andreas Körner
ARGESIM/Math. Modelling & Simulation Group,
Vienna Univ. of Technology /101,
Wiedner Haupt
strasse 8-10, 1040 Vienna , Austria

SNE Simulation Notes Europe

SNE Online ISSN 2306-0271, SNE Print ISSN 2305-9974

WEB: → *www.sne-journal.org*, DOI prefix 10.11128/sne

Scope: Developments and trends in modelling and simulation in various areas and in application and theory; comparative studies and benchmarks (documentation of ARGESIM Benchmarks on modelling approaches and simulation implementations); modelling and simulation in and for education, simulation-based e-learning; society information and membership information for EUROSIM members (Federation of European Simulation Societies and Groups).

Editor-in-Chief: Felix Breitenecker, Vienna Univ. of Technology, Math. Modelling and Simulation Group
Felix.Breitenecker@tuwien.ac.at, *eic@sne-journal.org*

Layout / Administration: A. Körner, A. Mathe, J. Tanzler, R. Leskovar, et al.; *office@sne-journal.org*

Print SNE and Print-on-Demand: Grafisches Zentrum and TU-Verlag, TU Wien, Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria – *www.tuverlag.at*

Publisher: ARGESIM Arbeitsgemeinschaft Simulation News c/o Math. Modelling and Simulation Group, TU Wien/101, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria; *www.argesim.org*, *info@argesim.org* on behalf of ASIM *www.asim-gi.org* and EUROSIM → *www.eurosim.info*
© ARGESIM / EUROSIM / ASIM 2018

SNE - Aims and Scope

Simulation Notes Europe (SNE) provides an international, high-quality forum for presentation of new ideas and approaches in simulation - from modelling to experiment analysis, from implementation to verification, from validation to identification, from numerics to visualisation - in context of the simulation process.

SNE seeks to serve scientists, researchers, developers and users of the simulation process across a variety of theoretical and applied fields in pursuit of novel ideas in simulation and to enable the exchange of experience and knowledge through descriptions of specific applications. **SNE** follows the recent developments and trends of modelling and simulation in new and/or joining application areas, as complex systems and big data. **SNE** puts special emphasis on the overall view in simulation, and on comparative investigations, as benchmarks and comparisons in methodology and application. For this purpose, **SNE** documents the **ARGESIM Benchmarks** on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions. SNE welcomes also contributions in education in/for/with simulation.

A News Section in **SNE** provides information for **EUROSIM** Simulation Societies and Simulation Groups.

SNE, primarily an electronic journal, follows an open access strategy, with free download in basic layout. **SNE** is the official membership journal of **EUROSIM**, the *Federation of European Simulation Societies and Simulation Groups* – www.eurosim.info. Members of **EUROSIM** societies are entitled to download **SNE** in an elaborate and extended layout, and to access additional sources of benchmark publications, model sources, etc. **Print SNE** is available for specific groups of **EUROSIM** societies, and starting with Volume 27 (2017) as print-on-demand from TU Verlag, TU Wien. **SNE** is DOI indexed by CrossRef, identified by DOI prefix 10.11128, assigned to the **SNE** publisher **ARGESIM** (www.argesim.org).

Author's Info. Individual submissions of scientific papers are welcome, as well as post-conference publications of contributions from conferences of **EUROSIM** societies. **SNE** welcomes special issues, either dedicated to special areas and/or new developments, or on occasion of events as conferences and workshops with special emphasis.

Authors are invited to submit contributions which have not been published and have not being considered for publication elsewhere to the **SNE** Editorial Office.

SNE distinguishes different types of contributions (*Notes*), i.e.

- **TN** Technical Note, 6 – 10 p.
- **EN** Education Note –6 – 8 p.
- **PN** Project Note 6 – 8 p.
- **SN** Short Note, max. 6 p.
- **SW** Software Note , 4 – 6 p.
- **BN** Benchmark Note, 2 – 8 p.
- **ON** Overview Note *ON* – only upon invitation, up to 14 p.

Further info and templates (doc, tex) at **SNE**'s website.

www.sne-journal.org

SNE Editorial Board

SNE - Simulation Notes Europe is advised and supervised by an international scientific editorial board. This board is taking care on peer reviewing of submission to SNE. At present, the board is increasing (see website www.sne-journal.org):

David Al-Dabass, *david.al-dabass@ntu.ac.uk*,
Nottingham Trent University, UK
Felix Breitenecker, *Felix.Breitenecker@tuwien.ac.at*
Vienna Univ. of Technology, Austria, Editor-in-chief
Maja Atanasijevic-Kunc, *maja.atanasijevic@fe.uni-lj.si*
Univ. of Ljubljana, Lab. Modelling & Control, Slovenia
Aleš Belič, ales.belic@sandoz.com
Sandoz / National Inst. f. Chemistry, Slovenia
Peter Breedveld, *P.C.Breedveld@el.utwente.nl*
University of Twenty, Netherlands
Agostino Bruzzone, *agostino@itim.unige.it*
Universita degli Studi di Genova, Italy
Francois Cellier, *fcellier@inf.ethz.ch*
ETH Zurich, Switzerland
Vlatko Čerić, *vceric@efzg.hr*
Univ. Zagreb, Croatia
Russell Cheng, *rchc@maths.soton.ac.uk*
University of Southampton, UK
Roberto Cianci, *cianci@dime.unige.it*, Math. Eng. and Simulation,
Univ. Genova, Italy
Eric Dahlquist, *erik.dahlquist@mdh.se*, Mälardalen Univ., Sweden
Horst Ecker, *Horst.Ecker@tuwien.ac.at*
Vienna Univ. of Technology, Inst. f. Mechanics, Austria
Vadim Engelson, vadim.engelson@mathcore.com
MathCore Engineering, Linköping, Sweden
Edmond Hajrizi, *ehajrizi@ubt-uni.net*
University for Business and Technology, Pristina, Kosovo
András Jávor, *javor@eik.bme.hu*,
Budapest Univ. of Technology and Economics, Hungary
Esko Juuso, *esko.juuso@oulu.fi*
Univ. Oulu, Dept. Process/Environmental Eng., Finland
Kaj Juslin, kaj.juslin@vtt.fi
VTT Technical Research Centre of Finland, Finland
Andreas Körner, andreas.koerner@tuwien.ac.at
Technical Univ. Vienna, E-Learning Dpt., Vienna, Austria
Francesco Longo, *f.longo@unical.it*
Univ. of Calabria, Mechanical Department, Italy
Yuri Merkuryev, *merkur@itl.rtu.lv*, Riga Technical Univ.
David Murray-Smith, *d.murray-smith@elec.gla.ac.uk*
University of Glasgow, Fac. Electrical Engineering, UK
Gasper Music, *gasper.music@fe.uni-lj.si*
Univ. of Ljubljana, Fac. Electrical Engineering, Slovenia
Thorsten Pawletta, *pawel@mb.hs-wismar.de*
Univ. Wismar, Dept. Comp. Engineering, Wismar, Germany
Niki Popper, *niki.popper@dwh.at*
dwh Simulation Services, Vienna, Austria
Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Univ. Tirana, Inst.f. Statistics, Albania
Thomas Schriber, *schriber@umich.edu*
University of Michigan, Business School, USA
Yuri Senichenkov, *sneyb@dcn.infos.ru*
St. Petersburg Technical University, Russia
Oliver Ullrich, oullrich@cs.1u.edu
Florida International University, USA
Siegfried Wassertheurer, *Siegfried.Wassertheurer@ait.ac.at*
AIT Austrian Inst. of Technology, Vienna, Austria
Sigrid Wenzel, *S.Wenzel@uni-kassel.de*
Univ. Kassel, Inst. f. Production Technique, Germany

A Solution to ARGESIM Benchmark C21 'State Events and Structural-dynamic Systems' based on Modelica Components

Jan-Philipp Disselkamp, Peter Junglas*, Alexander Niehüser, Phillip Schönfelder

Department of Engineering "Dr. Jürgen Ulderup", PHWT Vechta/Diepholz, Schlesierstr. 13a, 49356 Diepholz, Germany
*peter@peter-junglas.de

SNE 28(2), 2018, 39-48, DOI: 10.11128/sne.28.bn21.10411
Received: March 10, 2018 (Draft); Revised May 15, 2018 (Final)
Accepted: May 20, 2018
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The ARGESIM C21 benchmark 'State Events and Structural-dynamic Systems' adresses difficulties that appear in the modelling and simulation of discrete systems with state and structure-changing events. The solution presented here uses Modelica and a component based approach. It shows that even though Modelica may have conceptual problems modelling such systems, it is capable to deal with all the tasks of the benchmark in a straightforward way.

Introduction

The ARGESIM C21 benchmark [1] deals with systems showing state events or even structural-dynamic behaviour. It requires to investigate three different examples: a bouncing ball, an RLC circuit with a diode and a rotating pendulum with a free flight phase. The solution shown in the following applies a component based modelling approach using the Modelica language [2] and its standard library MSL.

Since the benchmark is quite complex and consists of several subtasks, we concentrate here on the concrete tasks defined in the benchmark itself. The definition of the studied example systems in full detail can be found in [1]. Different approaches all based on Modelica components have been compared in [3], together with a discussion of underlying conceptions and encountered problems.

With Modelica one has a choice between several simulation programs. The results presented here have been obtained using MapleSim 2017-3 from Maplesoft under Kubuntu 16.04. Using Dymola from Dassault Systemes leads to identical results in most cases. Some implementation problems that showed up in one or both systems as well as minor numerical deviations are described in [3].

The models and scripts necessary to reproduce all results presented here are available from [4].

1 Case Study Bouncing Ball

The Bouncing Ball example is a model for a falling mass with or without air resistance that is reflected when hitting the ground. The reflection is either described as a simple timeless event or as a continuous process using a spring-damper model for the deformation of the ball.

1.1 Event contact model

Description of model implementation. The 'bouncing ball' model uses concepts and components of the `Mechanics.Translational` and `Blocks` parts of the MSL. One creates a component for each force acting on the falling mass, including a `Hardstop` component that is responsible for the bounce (cf. Figure 1). Except for the hardstop all components are standard or easily implemented and produce the continuous equations of the system.

For the implementation of the hardstop two different versions have been studied: A simple one, based on [5, p. 57], is defined by the following Modelica code:

```
model Hardstop1d
    parameter Real mu = 0.9;
    Position s;
```

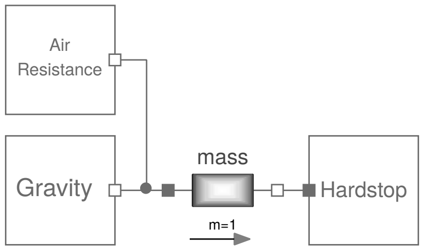


Figure 1: Bouncing ball with event contact

```
    Velocity v;
    Flange_a flange_a;
equation
    s = flange_a.s;
    v = der(s);
    flange_a.f = 0;
    when s <= 0 then
        reinit(v, -mu * pre(v));
    end when;
end Hardstop1d;
```

The `when` construction is the basic method in Modelica to create an event, the `reinit` restarts the solver with new initial values. This implementation leads to the notorious fall through problem near the Zenon point.

To cope with this one has to define another event `flying` (like in [2, pp. 96f]) and add a counter force at the hardstop

```
flying = not (s <= 0 and v <= 0);
flange_a.f = if flying then 0 else -m*g;
```

The resulting `Hardstop1dA` components works properly, the mass comes to rest after a large (but finite) number of bounces.

Simulation until last bounce – scattering prevention. With the given parameters for the free fall case one computes from [1, eq (16)] the bouncing time limit $t_{B,\infty} = 27.1290$ s. Using `Hardstop1dA` and standard solver parameters the simulation gives $t_{B,\infty} = 27.1287$ s. Adding air resistance results in $t_{B,\infty} = 25.5894$ s.

For the defective case of `Hardstop1d` the benchmark suggests adding a maximal height event and stopping the bouncing accordingly. This can be implemented in Modelica in the following way:

```
isFalling = v < 0;
isAtTop = edge(isFalling);
when isAtTop and s <= stopHeight then
  reinit(s, 0);
  reinit(v, 0);
  flange_a.f = -m*g;
elsewhen s <= 0 then
  reinit(v, -mu * pre(v));
  flange_a.f = 0;
end when;
```

The resulting limit time of course depends on the value of the parameter `stopHeight`. Interestingly, the simulation works even for the value `stopHeight` = 0 and reproduces the former result with air resistance, while in the free fall case the bounces stop earlier at $t_{B,\infty} = 27.1166$ s.

Testing accuracy of event handling. To determine the bounce times the `Hardstop1dB` component contains variables for the number and time of the last bounce that are updated at the bounce event. Figure 2 shows the difference between the theoretical values and the simulation results for a model without air resistance.

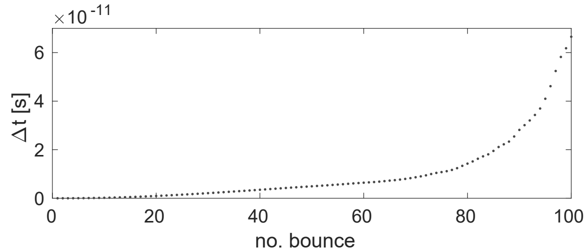


Figure 2: Accuracy of bounce times

Compensation of linear model deviation. This task asks to compensate for the later bouncing of the linear model (i. e. the one without air resistance) by introducing an initial velocity v_0. Doing so for the linear model one cannot reach identical final bounce times, because a simple calculation shows that it is bounded below by

$$t_{B,\infty}^{min} = \frac{2}{1-\mu}\sqrt{\frac{2x_0\mu}{g}} = 27.09\,\text{s}$$

for the given values.

Therefore one has to introduce an initial velocity into the nonlinear model. To compute it, one can solve its ODE analytically (which is easily possible outside the bounces), use a small Matlab script to add up the bounces and compute the final bounce time as function of v_0. Finally an application of fzero gives the requested value

$$v_0 = 4.39563\,\mathrm{m/s}$$

Figure 3 shows the solutions of the original linear and the shifted nonlinear model.

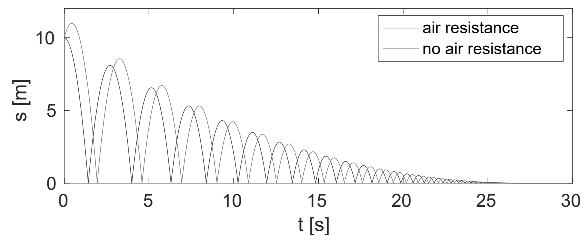


Figure 3: Compensation of linear deviation.

1.2 Model with continuous contact

Description of model implementation. The implementation of the bouncing ball model with continuous contact is very similar to the event based model, only the `Hardstop` component has been exchanged by an `ElastoGap` component (cf. Figure 4).

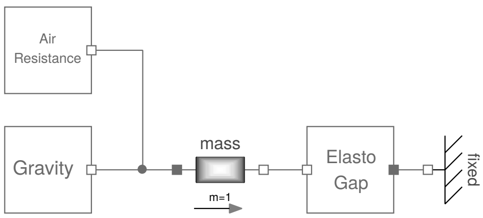


Figure 4: Bouncing Ball with continuous contact.

The MSL library already contains an `ElastoGap`, but it is more complicated in order to cope with unphysical situations. It is easy to adapt it to the benchmark requirements by using the following equations:

```
equation
  s_rel = flange_b.s - flange_a.s;
  v_rel = der(s_rel);
  y = s_rel + w;
  hasContact = (y <= 0);
  fc = if hasContact then
     -c*s_rel - d*v_rel else 0;
  flyRestarted = (fc <= 0);
  der(w) =
    if (hasContact and not flyRestarted)
      then -v_rel else -(c/d)*w;
  flange_a.f = fc;
  flange_b.f = -fc;
```

The events are defined implicitely through the `if` expressions and the corresponding logical variables. As always in Modelica the complete set of equations of the model is collected from all components and connections, then preprocessed and simplified. Only after these transformations the simulation program chooses appropriate state variables. This procedure makes it difficult to compare it with the general approaches defined in [1]. But since the total number of variables and equations is fixed in a Modelica model, one would probably describe it best as a 'maximal state space approach'.

Dependency of results from algorithms. MapleSim offers the choice of three variable-step solvers, which are all well known: the Runge-Kutta-Fehlberg solver RKF45, a Cash-Karp solver CK45 and a Rosenbrock solver ROS of third-fourth order. They have all been adapted by Maplesoft to cope with DAE systems. The model has been simulated with all three solvers and the following parameters: ε_{abs} = 1e-6, ε_{rel} = 1e-6, N_{plot} = 30001. Reference values with higher accuracy have been created using RKF45 and the parameters ε_{abs} = 1e-12, ε_{rel} = 1e-12. Using a different solver for the reference values leads to almost identical results. Creation of additional output points at events has been switched off to get output values at fixed times.

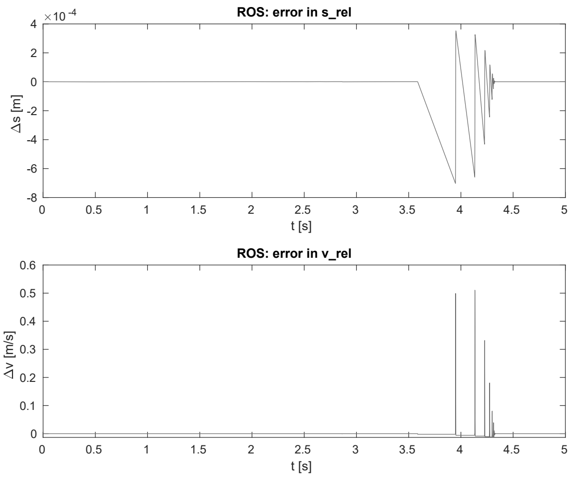


Figure 5: Errors for solver ROS.

All error plots are very similar, a typical result is shown in Figure 5. The much higher error of the velocity is due to its large slope together with deviations in the event times (cf. Figure 6).

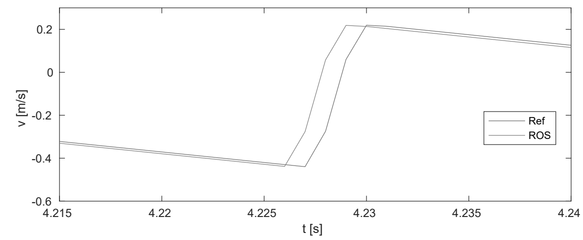


Figure 6: Velocity results near a bounce.

The maximal errors against the reference solution are given in Table 1. The standard solver CK45 is the worst here, one should use the ROS solver instead, which is generally recommended for stiff problems, and gives much better results than the other two. This is no surprise, since DAEs generally require stiff solvers.

	RKF45	CK45	ROS
s [1e-3 m]	3.7940	11.8984	0.7023
v [m/s]	2.3443	2.6007	0.5104

Table 1: Absolute errors for different solvers.

Investigation of contact phase. In order to output values of the maximal height h_{max} und maximal depression w_{max} the component `ElastoGapA` has been extended to create additional output events. For a closer look at the contact phase the model is simulated with higher accuracy (ε_{abs} = 1e-10, ε_{rel} = 1e-10, N_{plot} = 100000) using the Rosenbrock solver.

The results of the state and output variables (including the contact force) can be seen in Figure 7 and Figure 8 for the first and second contact phases and in Figure 9 for the second flight phase.

Table 2 shows the values for the maximal height and maximal depression. After the first 10 bounces the ball doesn't reach another flight phase, but oscillates while remaining in contact phase.

Parameter studies. Increasing k by a factor 100 leads to much more bounces since the contact time and the energy loss per bounce are small. Decreasing k by 100 leads to a sticking behaviour, the energy loss is too high for a second flight phase. A decrease or increase of d by a factor 10 leads to similar results for the same reasons (cf. Figure 10).

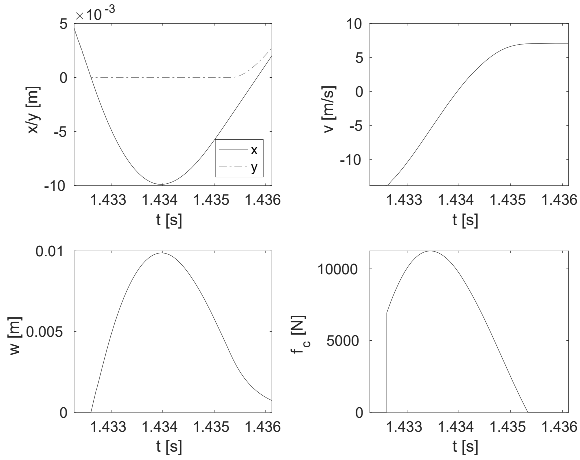


Figure 7: First contact phase.

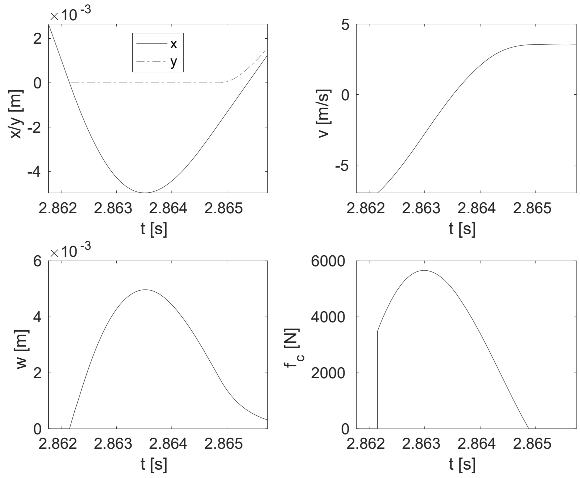


Figure 8: Second contact phase.

The results for changing d by a factor F and changing k by a factor $1/F^2$ are almost identical (cf. Figure 11). This can be explained easily by solving the simple contact equation [1, eq. 21] analytically, which shows that the percentage of energy loss per bounce depends on k/d^2 [6].

Bouncing Ball on Mars. Due to the lower gravity the bouncing ball behaviour on Mars is stretched in time, but otherwise similar (cf. Figure 12). The small differences are due to the different air resistances.

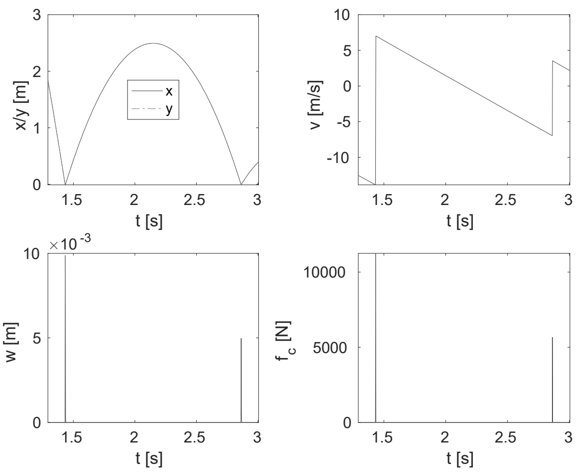


Figure 9: Second flight phase.

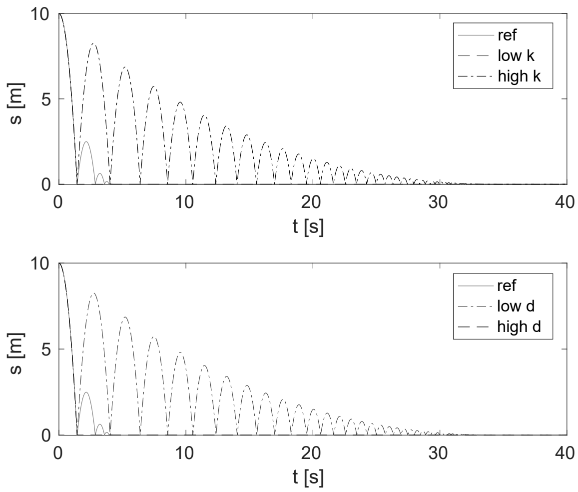


Figure 10: Variation of k and d.

2 Case Study RLC Circuit with Diode

The second example is a simple RLC circuit with a diode, where different diode models are to be investigated. It is constructed easily with standard components from the `Electrical.Analog` part of the Modelica Standard Library (MSL) (cf. Figure 13), which even contains two simple diode components. This model is used throughout this section, only the implementation of the diode component is changed.

Description of model implementations. To simplify the construction of electrical components, the

n	h_{max} [m]	w_{max} [mm]
1	10.00000000000	9.87387827
2	2.49466100656	4.97183608
3	0.63276017545	2.51178619
4	0.16058207486	1.26910129
5	0.04046397059	0.64031534
6	0.01002858933	0.32201325
7	0.00239917584	0.16085257
8	0.00053012002	0.07922586
9	0.00009490838	0.03783789
10	0.00000588204	0.01701446
11	-0.00000660874	0.01123246
12	-0.00000917794	0.01009085
13	-0.00000968520	0.00986545
14	-0.00000978536	0.00982095
15	-0.00000980527	0.00981176

Table 2: Maximal heights and depressions.

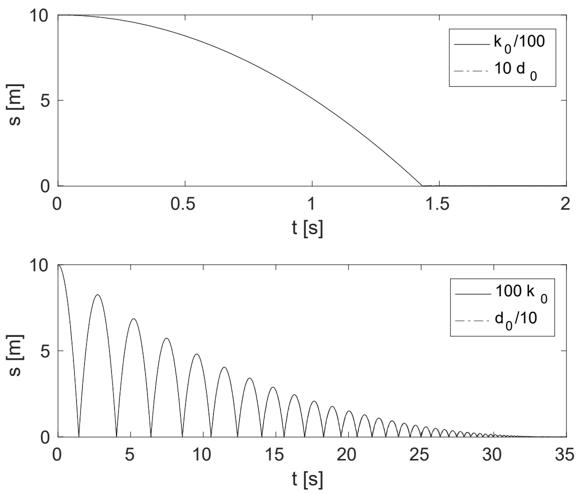


Figure 11: Comparison of k and d changes.

MSL contains a partial model `OnePort` that defines external connection points and the internal variables i und v. One creates a concrete model by inheriting from `OnePort` and adding the equation that defines the connection between i and v. This mechanism will be used in the following for all diode models.

The shortcut diode can be implemented using the `IdealDiode` from MSL. A simpler version can be

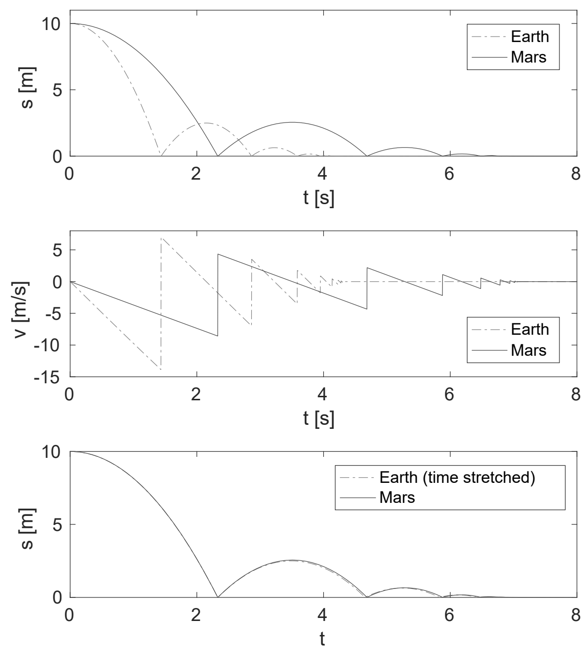


Figure 12: Bouncing ball on Earth and on Mars.

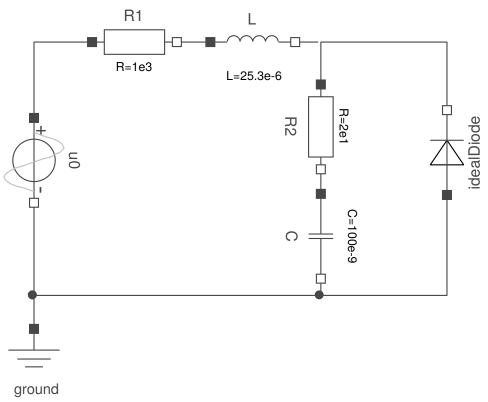


Figure 13: RLC circuit with diode.

found in [5, p.56], which uses a standard trick to cope with the non-functional relation between *i* and *v*:

```
model DIsc1 "short-cut diode"
  extends OnePort;
  Real s;
  Boolean off;
equation
  off = s < 0;
  v = if off then s else 0;
  i = if off then 0 else s;
end DIsc1;
```

The Shockley diode `DIshu1` is even simpler, it defines parameters I_S and U_T and the equation

```
i = if v < 0 then 0 else IS*(exp(v/UT)-1);
```

For the implementation of the approximated Shockley diode one writes a simple linear interpolation function and has

```
model DIas1 "approximated Shockley diode"
  extends OnePort;
  parameter Real IS = 1e-8;
  parameter Real UT = 26e-3;
  parameter Integer N = 10;
  parameter Voltage uMax = 3.84e-2;
  Real up[N] = linspace(0, uMax, N);
  Real ip[N] = IS*(exp(up/UT) - ones(N));
  Boolean off;
equation
  off = v < 0;
  i = if off then 0
           else linInterp(v, up, ip);
end DIas1;
```

The final model is the 'explicit Shockley diode', which is defined by differentiating the algebraic equation of the complete RLC model. In the context of a component based environment used here, one can only differentiate the *i*-*v* relation to get the explicit component `DIesu1` defined by the equations

```
  off = v < 0;
  if off then
    i = 0;
  else
    der(i) = (Is/UT)*exp(v/UT)*der(v);
  end if;
```

The events are again defined implicitely by the if-expressions. According to the terminology of [1], one may call this a 'switching model parts' approach. The only noteworthy detail is in the explicit diode, where the variable *i* is a state variable (differentiated) in one branch, and a simple algebraic variable in the other. Such a situation often presents problems for the simulation environment, but MapleSim works nicely here. The well-known Dymola program can't cope with this component and stops the simulation, when the first locking phase appears, claiming to hit upon a singular linear system of equations. The obvious workaround – substitute `i = 0` by `der(i) = 0` – makes *i* a state variable always and saves the day of the Dymola user.

Dependency of results from algorithms. The general procedure (using reference values of high accuracy) and the general solver parameters are the same as in Section 1.2.

For the interesting variables i_L, u_C, i_D and u_D relative errors have been computed by comparing to the reference solution and scaling by maximal absolute values of the variable. The results for the different solvers are displayed in Table 3 for the shortcut diode and in Table 4 for the Shockley diode.

	RKF45	CK45	ROS
ε_{iL}	425.90	440.30	305.94
ε_{uC}	4.65	3.48	4.13
ε_{iD}	549.64	328.32	243.72
ε_{uD}	254.71	163.66	19.29

Table 3: Shortcut diode: Relative errors [in 1e-6].

	RKF45	CK45	ROS
ε_{iL}	521.08	494.31	42.67
ε_{uC}	5.59	5.97	5.82
ε_{iD}	215.95	199.58	18.82
ε_{uD}	283.15	302.26	27.23

Table 4: Shockley diode: Relative errors [in 1e-6].

Again the Rosenbrock solver gives the highest accuracy, but in the shortcut model the difference to the other solvers is only marginal (except for u_D), whereas it is an order of magnitude in the Shockley model.

The behaviour of the errors over time is similar for all variables and both models. An example for the shortcut diode and variable i_D is shown in Figure 14.

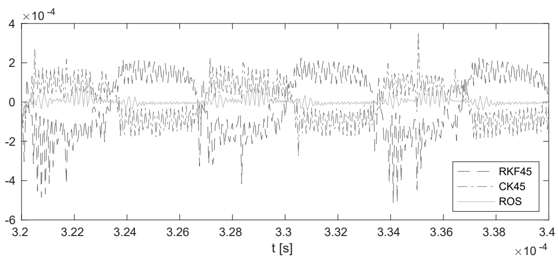


Figure 14: Shortcut diode: Relative errors for i_D.

Comparison of shortcut and Shockley diode model. Figure 15 shows the behaviour of the relevant variables for models with shortcut resp. Shockley diode over two switching periods starting at 0.3 ms to get rid of initial effects. The very small values of the diode current i_D for the Shockley diode are due to its rather large value of U_T leading to a high resistance in conducting phase.

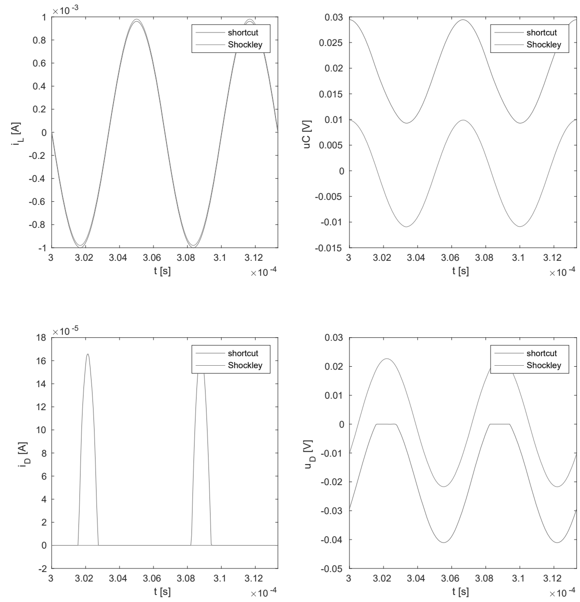


Figure 15: Comparison of shortcut and Shockley diode.

Simulation times have been obtained by performing seven runs each and computing mean values of the last five, thereby minimizing initial loading time effects. MapleSim outputs timing values for different stages of the computation, which shows that the largest part here is not the integration itself, but a task described as ‘preparing for integration’. This part is done much faster for the Shockley model than for the shortcut model, reducing the total computation time by 44%.

Approximation of Shockley diode model. The model using the approximated Shockley diode with different numbers of interpolation points almost reproduces the results of the Shockley diode, the only notable difference being the diode current i_D (cf. Figure 16). The corresponding plot of the absolute errors nicely shows the approximation points. Relative errors for the other variables are small, a typical behaviour is displayed in the lower plot.

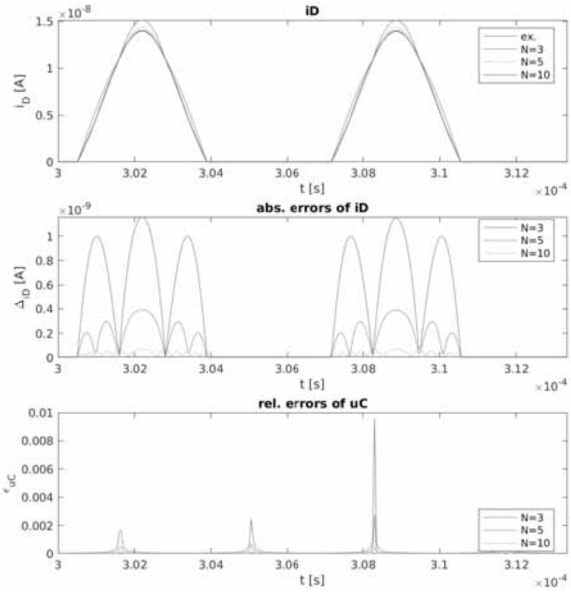


Figure 16: Comparison of Shockley and approx. Shockley diode.

Relevance of choice of algebraic state. It's easy to use the inverse relation $v(i)$ for the Shockley diode writing

```
if v < 0 then
  i = 0;
else
  v = UT * log(i / IS + 1);
end if;
```

This leads to identical simulation results. But interestingly, the computing times are different drastically: The new variant is 25 times slower in MapleSim. In Dymola the simulation times do not differ at all.

Investigation for real-time simulation. MapleSim provides a few fixed-step solvers, in the following the RK4 solver is used with a step size of 1e-8. Furthermore the constraint projection has been switched off and the number of event iterations set to 1. Compared to the standard solver parameters this leads to identical results for the shortcut and shockley diodes, whereas the model using the "explicit Shockley" diode `DIesu1` differs in the diode current i_D: It drifts to negative values during the locking phases (cf. Figure 17).

As described above the equation used inside the diode component is

```
der(i) = (Is/UT)*exp(v/UT)*der(v);
```

The alternative component `DIesi1` uses the derivative of the inverse relation $v(i)$ and leads to slightly different deviations. Figure 17 compares both explicit versions with the correct Shockley diode.

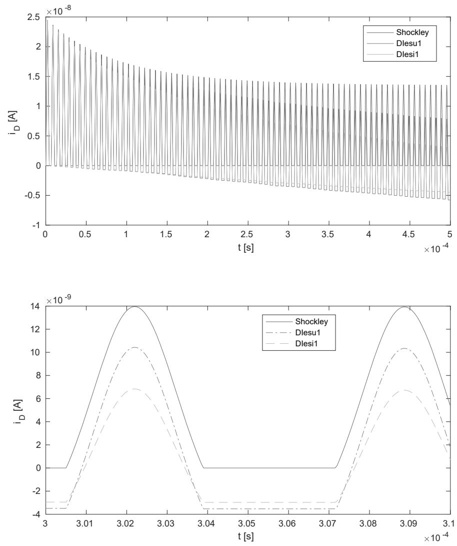


Figure 17: Comparison of Shockley and explicit Shockley diodes.

3 Case Study Rotating Pendulum With Free Flight Phase

The last example is a point mass with air resistance on a rope of fixed length. Its movement switches between swinging and free fall phases according to the direction of the force acting on the mass.

Description of model implementations. The implementation of the rotating pendulum consists of separate blocks for the two different system configurations and a `SystemSwitch` that alternatively activates one or the other system, depending on the state of the active system (cf. Figure 18). The block below computes some additional plot variables.

The switch contains the event functions h^F, h^S as defined in [1] and computes the initial state at a system change:

```
h1 = -g*m*cos(state1[1])+m*l*state1[2]^2;
h2 = l^2 - state2[1]^2 - state2[2]^2;
when h1 < 0 then
  active1 = false;
```

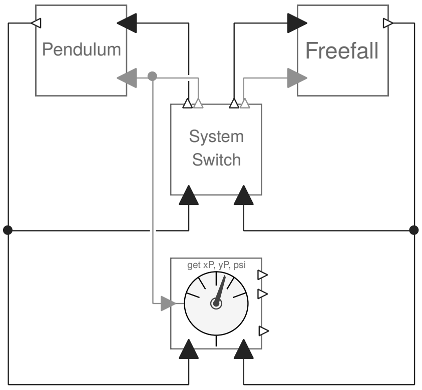


Figure 18: Rotating pendulum model.

```
elsewhen h2 < 0 then
  active1 = true;
end when;
active2 = not active1;

new1[1] = atan2(state2[1],state2[2]);
new1[2] = (state2[2]*state2[3]
          - state2[1]*state2[4])/l^2;
new2[1] = l*sin(state1[1]);
new2[2] = l*cos(state1[1]);
new2[3] = l*state1[2]*cos(state1[1]);
new2[4] = -l*state1[2]*sin(state1[1]);
```

To facilitate the implementation of the two systems a partial model `SwitchableSystem` has been defined that contains the state, the inputs and outputs and the triggering:

```
partial model SwitchableSystem
  parameter Integer N = 2
  parameter Real[N] s0 = {pi/4, 15};
  RealInput[N] newState;
  BooleanInput active;
  RealOutput[N] sOut;
  Real[N] state(start=s0, each fixed=true,
      each stateSelect=StateSelect.always);
equation
  when active then
    reinit(state, pre(newState));
  end when;
  sOut = if active then state else zeros(N);
end SwitchableSystem;
```

Using this the implementation of a concrete system only needs the definition of the state equations, usually in form of an ODE. Even a graphical approach is possible [3], but MapleSim cannot cope with such a model.

Finally one adds a few lines of explicit Modelica code to identify the (inherited) `state` variable with corresponding variables from the concrete model. For the pendulum this is as simple as

```
state[1] = pi/2 - revolute.phi;
state[2] = -revolute.w;
```

The attribute `stateSelect=StateSelect.always` of `state` guarantees that these variables will be used by the solver as the actual states.

This model looks exactly like a hybrid decomposition (cf. Figure 13 of [1]) and for the purpose of constructing the model it really is one: Both submodels can be created independently and almost in the same way as standalone systems. But formally this again is a maximal state space approach: The variables of an inactive system are simply ignored or their derivatives set to zero. In any case they always exist, enlarge the total state space and have to be computed always albeit trivially. On the other hand Modelica compilers routinely handle large systems with lots of trivial equations, so this should not be a large burden.

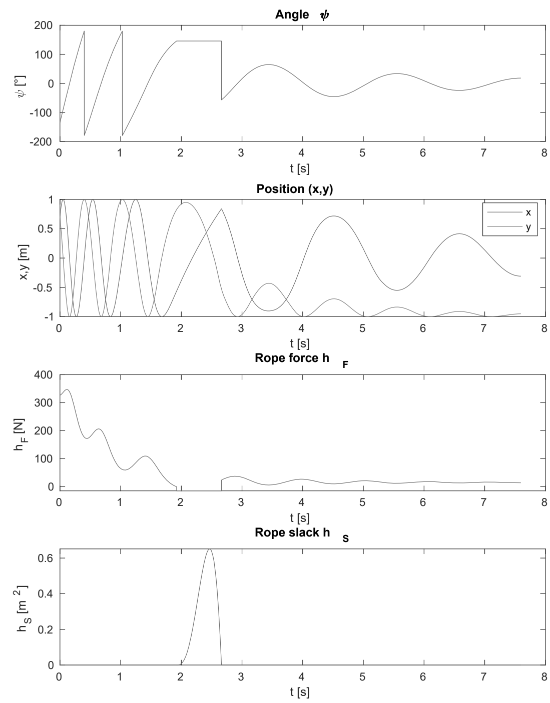


Figure 19: Results of pendulum model.

Basic simulation of phases. Using a simple `when` construction the basic model can be easily extended to stop after the amplitude is below $\pi/10$. This happens at t = 7.5962714 s, the corresponding values of state and event variables are displayed in Figure 19. The angle ψ shown there is measured against the lower equilibrium point and reduced to the interval $[-\pi,\pi]$

Dependency of results from algorithms. The procedure for comparing results that has been used twice before is employed again. The maximal absolute errors for the variables x, y and ψ are given in Table 5. They show that the Rosenbrock solver again has the highest accuracy, while the default solver CK45 performs worst. Figure 20 displays exemplary plots of the error over time.

	RKF45	CK45	ROS
x [10^{-6} m]	0.4804	0.8476	0.1080
y [10^{-6} m]	0.2597	0.5187	0.1941
ψ[10^{-6} rad]	0.4879	0.8801	0.2020

Table 5: Absolute errors (compared to reference solution).

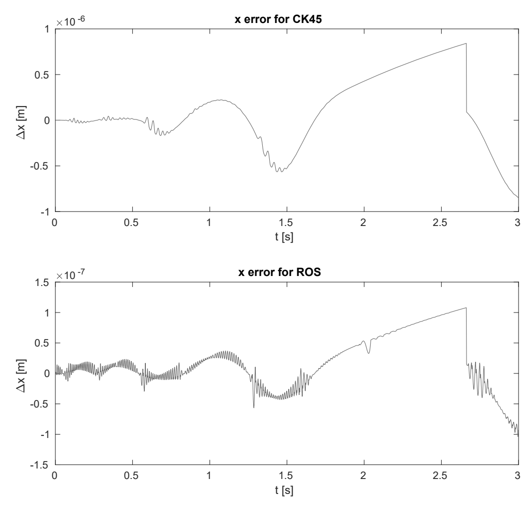


Figure 20: Absolute errors in x for two solvers.

External energy supply. To add the events necessary for the addition of the 'kick' – i. e. a jump of the angular velocity by a factor γ at the lowest point – one needs a `when-elsewhen` construction. This did not work due to a bug in MapleSim, but a simple workaround could be found [3].

The kick value γ is defined by the initial conditions $\psi_0 = 0$ and (unknown) ω_0 and given end conditions ψ_f and ω_f. To find it a simple model `RPkiAux1` has been used that integrates the time inverted pendulum ODE until the (initial) point $\psi = 0$ is reached. γ is then given by the ratio of ω_0 and the known value of ω before the kick. Table 6 shows the final values defining the three cases in [1] and the corresponding values of γ.

The definition of case (iii) "the swinging phase makes two rotations" is ambiguous, it could mean anything between 1.5 to 2.5 rotations before a fall from the top. The given value corresponds to the shortest path.

case	ψ_f	ω_f	γ
(i)	$-(5/4)\,\pi$	15	-21.9911
(ii)	$-\pi$	$\sqrt{g/l}$	-10.1501
(iii)	-3π	$\sqrt{g/l}$	-14.1982

Table 6: Final values and kick factor.

References

[1] Körner A, Breitenecker F. State Events and Structural-dynamic Systems: Definition of ARGESIM Benchmark C21. *Simulation Notes Europe*. 2016; 26(2): 117–122. doi: 10.11128/sne.26.bn21.10339.

[2] Modelica Association. *Modelica® - A Unified Object-Oriented Language for Systems Modeling - Language Specification Version 3.4, April 10, 2017.* Online: `https://modelica.org/documents/ModelicaSpec34.pdf` (called 2018-05-23).

[3] Disselkamp JP, Junglas P, Niehüser A, Schönfelder P. Implementing the Argesim C21 benchmark with Modelica components. In Loose T, editor. *Tagungsband Workshop 2018 ASIM/GI-Fachgruppen*; 2018 Mar; Heilbronn. 197–202.

[4] Junglas P. Argesim C21 models and scripts. Online: `http://www.peter-junglas.de/fh/simulation/argesimc21.html` (called 2018-05-23).

[5] Fritzson PA. Principles of Object-Oriented Modeling and Simulation with Modelica 3.3. Wiley & Sons, New York, 2015.

[6] Nagurka M, Huang S. A mass-spring-damper model of a bouncing ball. In *Proc. American Control Conference*; 2004 Jul; Boston. 499–504.

The RESIN Climate Change Adaptation Project and its Simple Modeling Approach for Risk-oriented Vulnerability Assessment

Daniel Lückerath, Manfred Bogen, Erich Rome, Betim Sojeva, Oliver Ullrich, Rainer Worst, Jingquan Xie

Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS
Schloss Birlinghoven, 53757 Sankt Augustin, Germany; **daniel.lueckenrath@iais.fraunhofer.de*

SNE 28(2), 2018, 49 - 54, DOI: 10.11128/sne.28.pn.10412
Received: April 10, 2018 (Selected ASIM GMMS/STS 2018 Postconf. Publ.), Accepted: May 15, 2018
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Urban population centers are especially vulnerable to extreme weather events and climate change. Local decision makers demand standardized processes, methods, and tools enabling them to design and implement climate adaptation strategies for their specific contexts. The project "Climate Resilient Cities and Infrastructures – RESIN" aims at providing such a set of methods and tools; it utilizes the impact chain modeling approach to capture and represent cause-effect relationships underlying risks and vulnerabilities in urban population centers, enabling a systematic analysis and evaluation. Highlighting on a number of concepts, such as hazard, exposure, stressors, coping capacity, and vulnerability, these impact chains constitute the base for further quantitative modeling steps.

Introduction

High concentrations of residents and economic assets render urban population centers especially vulnerable to the impact of extreme weather events and consequences of climate change (see [1]). Such disasters endanger large numbers of residents and critical infrastructure systems at the same time, thereby also impacting inter-regional and global economic networks (see [2]). The trend towards increasing urbanization in Europe – by 2050 82% of all Europeans are expected to live in urban population centers (see [3]) – and the increasing dependencies of infrastructure components make it necessary for municipalities to develop proactive strategies to increase their resilience against climate- related disasters.

Climate change adaptation measures, in contrast to climate protection efforts that are mainly aimed at reducing greenhouse gas emissions, are designed to reduce the impact of climate change on social and biological systems, such as urban population centers. Unfortunately, not many standardized methods and toolsets exist today that enable municipal decision makers to plan, assess, and implement adaptation measures, thereby helping them to consider, analyze, and evaluate risks and vulnerabilities under specific, climate change related scenarios. In the context of such a systematic scenario analysis, the application of a simulation model can be of considerable benefit. The foundation of such a model – and by extension of effective adaptation measures – is a comprehensive understanding of the risks and vulnerabilities themselves.

This paper describes characteristics and generation of impact chains representing cause-effect relationships that form risks and vulnerabilities of urban population centers in the context of the ongoing EU project "Climate Resilient Cities and Infrastructures – RESIN" (see [6]). The project is aimed at developing practical and applicable methods and tools to support municipalities in designing and implementing climate adaptation strategies for their local contexts. RESIN also systematically compares and evaluates methods for climate change adaptation in order to move towards a formal standardization of adaptation strategies.

RESIN is one of several interdisciplinary, practice-based research projects investigating climate resilience in European cities.

The EU project "Reconciling Adaptation, Mitigation and Sustainable development for citiES – RAMSES" (see [4]) that was concluded recently, developed methods and tools to quantify evidence of the impacts of climate change and the costs and benefits of adaptation measures to cities. The EU project "Smart Mature Resilience – SMR" (see [5]) aims at developing a resilience management guideline to support city decision-makers in developing and implementing resilience measures.

The paper continues with a short introduction to the RESIN project and its fundamental process (see Section 1). It then goes on to describe the impact chain modeling approach used to capture and represent cause-effect relationships underlying risks and vulnerabilities in urban population centers, enabling a systematic analysis and evaluation (see Section 2). The paper concludes with a short description of the steps necessary to complete the modeling process based on the generated impact chains (see Section 3).

1 The RESIN Project

RESIN investigates climate change adaptation practices in European cities in order to develop standardized methods and decision support tools that decision makers can use to develop local adaptation strategies.

The project builds on previous research by combining existing approaches to climate change adaptation and disaster risk management while taking into account all of the core elements of the urban system and their interrelations. One of the central aims of RESIN is to provide standardized methods and tools for comprehensive risk-oriented vulnerability assessments of an urban population center. These efforts are based on the latest state of discussion and conceptual approaches of the Intergovernmental Panel on Climate Change (IPCC) Assessment Report 5 (AR5, see [7]). RESIN views a city as a "system of systems" comprised of complex social, ecological, and technical sub-systems that overlap and interact with one another.

All RESIN methods and tools are being developed by means of co-creation with the cities of Bilbao (Spain), Greater Manchester (United Kingdom), Paris (France), and Bratislava (Slovakia). The first co-creation process with Bilbao, aiming at producing a vulnerability and risk mapping on a neighborhood (barrios) scale, started in July 2016 and concluded in September 2017.

Process, methods, and tools are standardized and can be applied to more European urban population centers, but at the same time support the tailoring to the specific needs of a municipality, depending on the varying degrees of maturity of their adaptation processes.

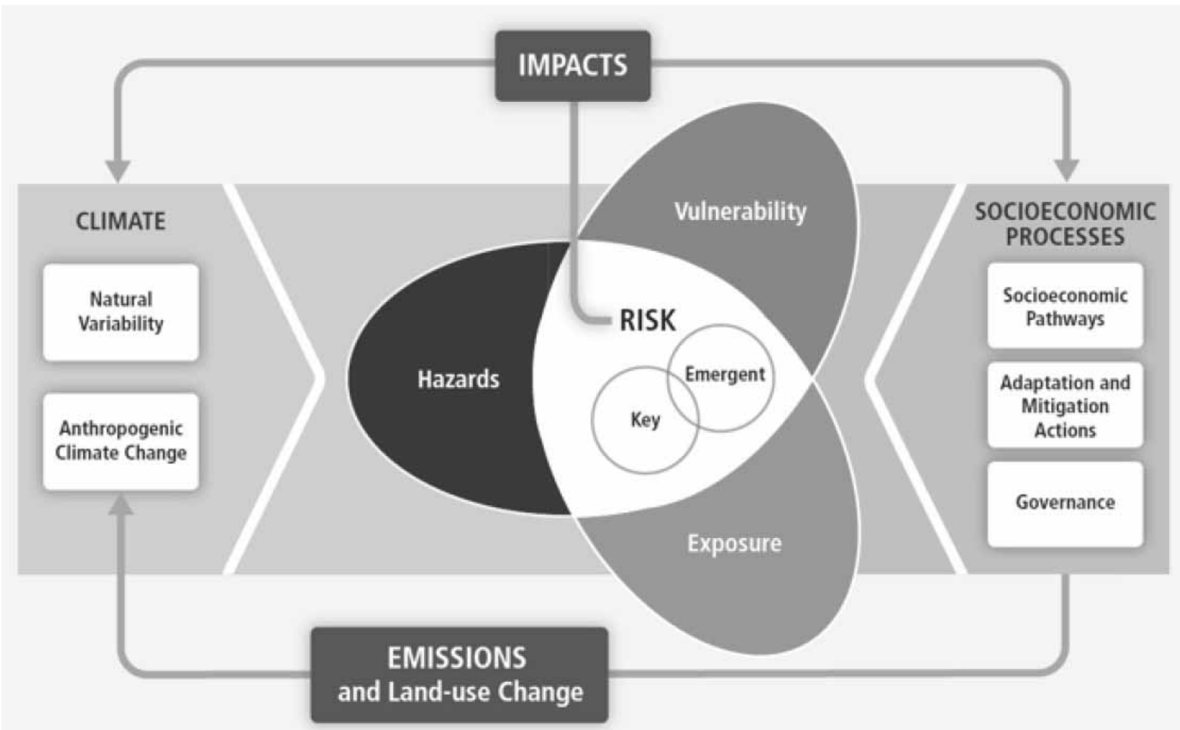


Figure 1. Risks as compositions of hazards, exposure, and vulnerability (source: [7]).

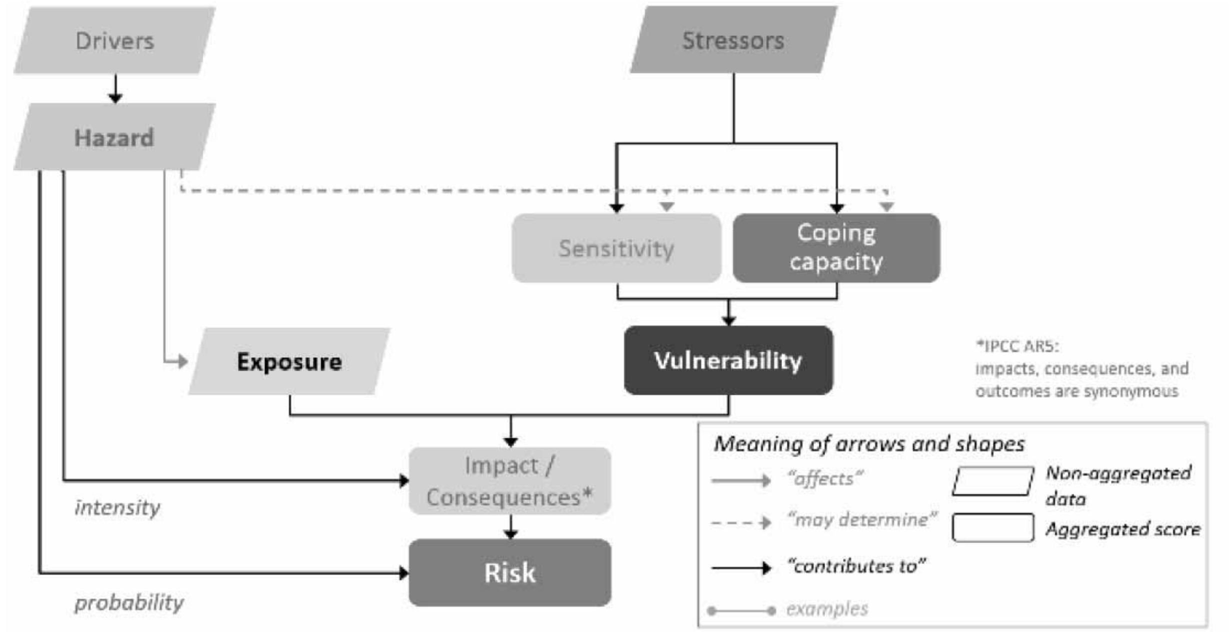


Figure 2. A risk-oriented vulnerability assessment schema (source: [11]).

The application of the RESIN conceptual framework follows four principal stages (see [8]):

1. Assessing climate risks
2. Developing adaptation objectives
3. Prioritizing adaptation options
4. Developing an implementation plan

The impact chain modeling technique described in the following section is part of the modelling process during the first stage of this adaptation planning process.

In RESIN, the design chosen for the vulnerability assessment process is based on the indicator-based framework provided by "The Vulnerability Sourcebook" (see [9]). Since the sourcebook is based on IPCC AR4 (see [10]), the framework had to be adapted to account for the conceptual move to a risk-based approach by IPCC AR5 (see [7]). Compared to AR4, AR5 defines and uses terms like *vulnerability* and *exposure* in different ways: Where exposure to climate change hazards was once considered to be part of vulnerability – alongside sensitivity to hazards and capacity to adapt –, the move to risk has separated out exposure.

As a result, risk is now regarded by the IPCC as a function of climate hazard, exposure, and vulnerability (see Figure 1).

2 Using Impact Chains to Model Risks and Vulnerabilities

Impact chains are tools (described in [9]) for capturing and structuring the components of a particular causeeffect relationship. They describe the basic connections between the elements, and prepare the selection of quantitative indicators for many of the components. Experiences during the co-creation processes with local experts from Bilbao, Bratislava, Manchester, and Paris show that impact chain diagrams are easy to grasp and apply, and that participants appreciate the structured modeling approach. As a result, impact chains are not exhaustive, but describe the common understanding of the stakeholders present at the workshops. Often, experts found that impact chain diagrams gave them first clues towards potential adaptation measures.

As part of impact chain modeling, RESIN utilizes a number of concepts (see Figure 2) to derive overall risk estimations: drivers, hazard, exposure, stressors, sensitivity, coping capacity, vulnerability, and impacts.

A *hazard* is defined as "…the potential occurrence of a natural or human-induced physical event or trend, or physical impact that may cause loss of life, injury, or other health impacts, as well as damage and loss to property, infrastructure, livelihoods, service provision, and environmental resources" (see [7]).

A climate-related hazard is a special case that is (at least partially) caused by climatic *drivers*. Examples include flooding, heatwave, drought, and water scarcity.

Exposure refers to the objects or systems that are exposed: The presence of people, livelihoods, species or ecosystems, environmental services and resources, infrastructure, or economic, social, or cultural assets in specific places that could be adversely affected.

Non-climatic trends and events, which are called *stressors*, can have an important effect on an exposed system. Examples are population growth or change of land-use; a larger percentage of sealed surface will in general increase the susceptibility to flooding events and thus the vulnerability of all exposed objects.

Different objects are more or less sensitive to a hazard. This is captured by the concept of sensitivity, defined as the degree to which an exposed object, species or system could be affected by the considered hazard. As such, sensitivity towards a hazard can be perceived as a property of an exposed object in regard to a specific hazard. Examples for sensitivity include the degrees of surface sealing, age and density of a population, household-income, or elevation and density of buildings.

Coping capacity is defined as "the ability of people, institutions, organizations, and systems, using available skills, values, beliefs, resources, and opportunities, to address, manage, and overcome adverse conditions in the short to medium term" (see [12]). Examples include the draining capacity of sewer systems, a dike's height, education and awareness of the population, and availability of early warning systems, while examples for adaptive capacity include diversity of economic activities, state of the city infrastructure, network redundancy, diversity of land-use, or availability of hospital beds.

Vulnerability is derived from the interplay of stressors, sensitivity, and coping capacity. It contributes directly to the impact or consequences that a hazard causes to the exposed objects.

Risk is classically computed by multiplying the probability of an adverse event with the magnitude of the expected consequences (see [12]). A risk assessment takes into account the characteristics and intensity of the considered hazard, as well as the set of objects exposed to it. The probability of a hazard affecting the set of objects may be estimated from extrapolating historical data or simulation results concerning the frequency of the hazard and the development of the objects. The vulnerability of the objects exposed to the hazard then determines the consequences.

The RESIN vulnerability assessment process starts with a systematic analysis and selection of hazards, drivers, and stressors relevant to the urban area under examination. These results serve as a base for the detailed planning of the assessment and ensures that the – usually limited – resources available for the assessment are spent on the most pressing current and future hazards, and no other threats or possible dependencies between different hazards are overlooked. In addition, a thorough documentation of the rationale for selecting hazards, drivers, and stressors is recommended to ensure that future assessments or re-evaluations yield comparable results.

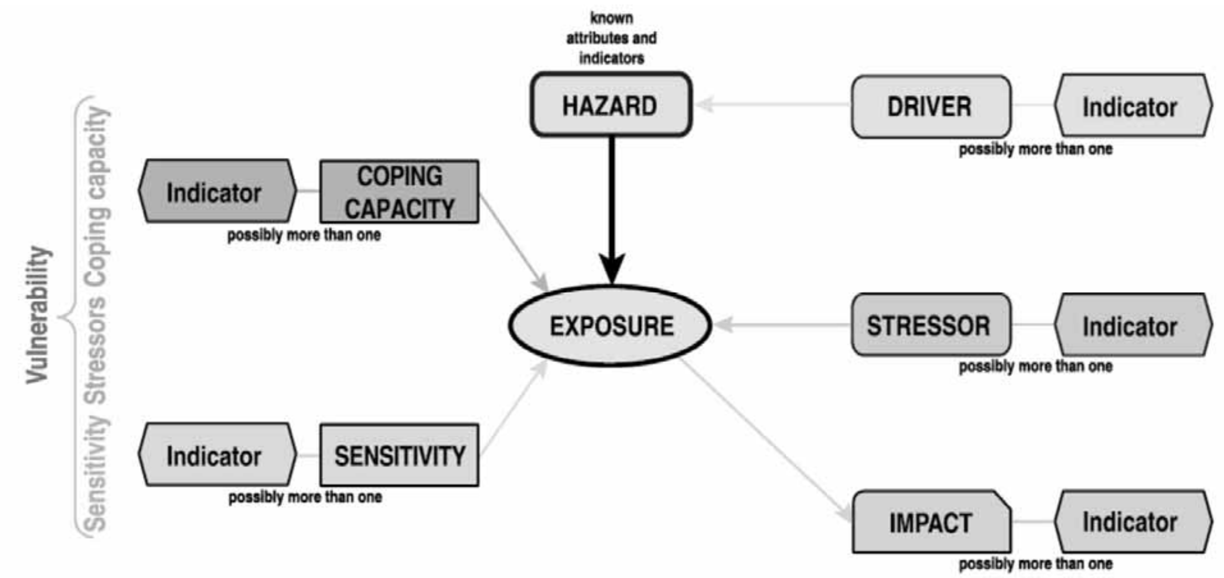


Figure 3. Generic impact chain diagram schema (source: [11]).

Once the vulnerability assessment is completed, impact chain diagrams (see Figure 3) are developed. The diagrams then visually illustrate cause-effect relationships between the elements contributing to the impact of a given combination of hazards and exposed objects.

RESIN impact chain diagrams explicitly differentiate between hazards on the one hand and impacts and consequences on the other (see Figure 4) – thus representing cause and effect. In addition, the diagrams model relationships between sensitivity and coping capacity, and exposed object. Each element of an impact chain may be described in a qualitative way by specifying attributes, such as "green infrastructure" for coping capacity, and later also in a quantitative way by assigning measurable indicators, such as "percentage of green area per city district". Usually, impact chain diagrams are developed during collaborative workshops with experts and stakeholders.

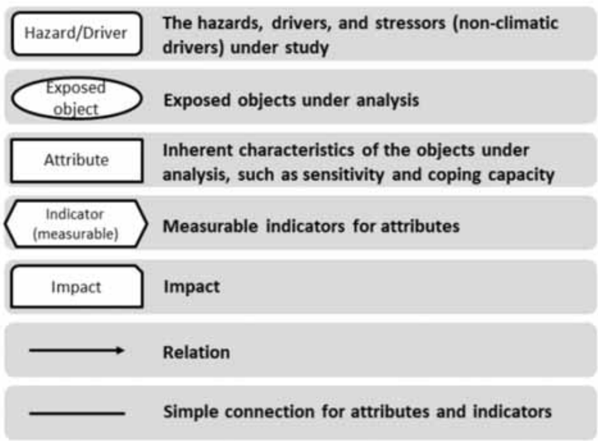


Figure 4. Elements used in impact chain diagrams (source: [11]).

Figure 5, developed during a co-creation workshop with local experts, shows an example impact chain for the hazard-exposure combination heatwave on public health for the city of Bilbao. Here, a heatwave is defined as a day when the average temperature exceeds 32°C. Both the coping capacity and the sensitivity indicators contain a mixture of infrastructure related measures as well as social indicators, such as the amount of green infrastructure and the percentage of elderly people. On the right side of the diagram, the impact indicators cover mainly health related and economic consequences.

The impact chain development (without the definition of measureable indicators) concludes the qualitative part of the assessment process and is a very valuable outcome itself. End-users without the necessary resources – both in terms of personnel and knowledge – may decide to end the vulnerability assessment at this point. Others may opt to go for the quantitative part outlined below.

3 Further Steps

Based on the insights gained during this phase, impact chains are annotated by further identifying measureable indicators for all identified elements, and by gathering necessary data for their calculation. To ease the indicator selection process, established directories of standard indicators can be employed. Such directories can be found, for example, in the annex of the Vulnerability Sourcebook (see [9]), the annex of the Covenant of Mayors for Climate and Energy Reporting Guidelines (see [13]) or the indicator database of the European Union Framework Programme 7 project MOVE (Methods for the Improvement of Vulnerability Assessment in Europe, see [14]). These annotated impact chains are then utilized as a base for evaluation and simulation models of impacts, hazards, sensitivity, coping capacity, or stressors – which in turn are applied to move from risk assessments based on historic data to risk assessment of future (simulated) scenarios.

Acknowledgments

The authors thank their partners in the RESIN consortium for their valuable contributions during the development and test process. This paper is based in part upon work in the framework of the European project "Climate Resilient Cities and Infrastructures – RESIN". This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 653522.

The sole responsibility for the content of this publication lies with the authors. It does not necessarily represent the opinion of the European Union. Neither the EASME nor the European Commission are responsible for any use that may be made of the information contained therein.

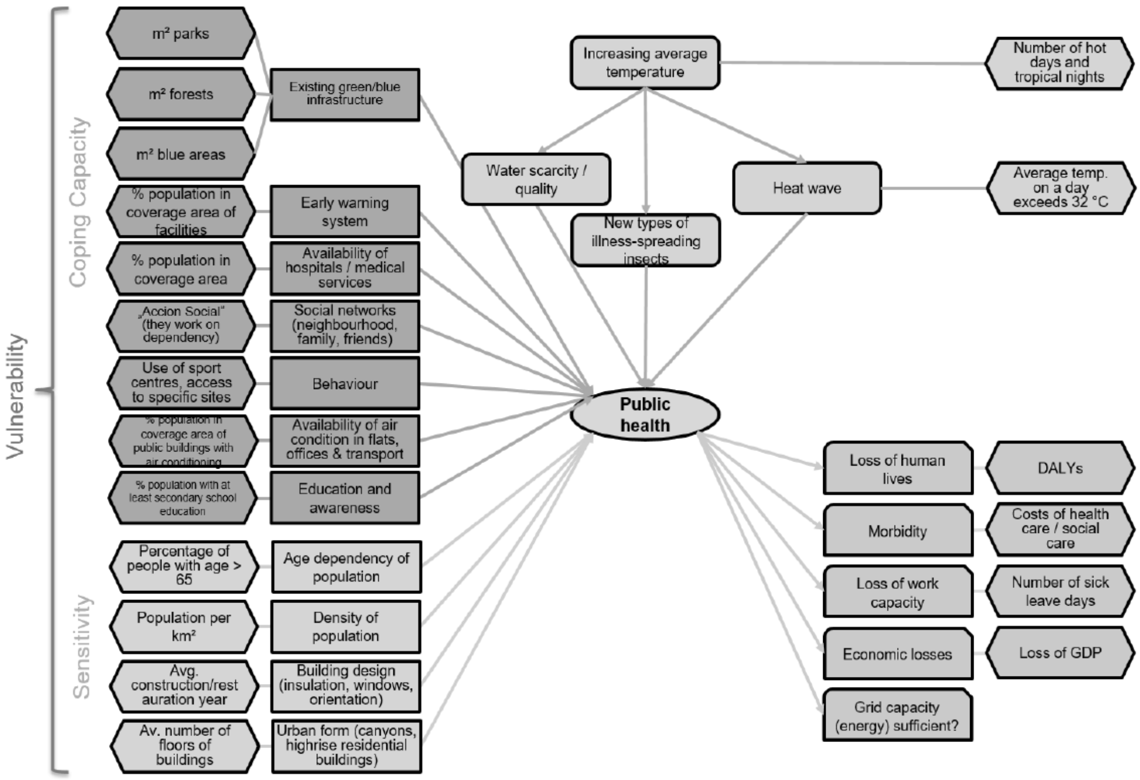


Figure 5. Example impact chain for the hazard-exposure combination heat wave on public health in the city of Bilbao (source:[11]).

References

[1] Coletti A, De Nicola A, Villani ML. *Building climate change into risk assessment*. In: Natural Hazards, Vol. 84, Issue 2, pp. 1307–1325, 2016.

[2] Burke M, Hsiang SM, Miguel E. *Global non-linear effects of temperature on economic production*. In: Nature, Vol. 527, pp. 235–239, 2015.

[3] United Nations, Department of Economic and Social Affairs, Population Division. *World Urbanization Prospects: The 2014 Revision, Highlights*. United Nations, 2014.

[4] RAMSES – Reconciling adaptation, mitigation and sustainable development for cities. URL, visited on Dec 05, 2017, http://www.ramsescities.eu/

[5] SMR – Smart Mature Resilience. URL, visited on Dec 05, 2017. http://smr-project.eu

[6] RESIN – Climate Resilient Cities and Infrastructures. URL, visited on Nov 11, 2017: http://www.resin-cities.eu/

[7] Intergovernmental Panel on Climate Change (IPCC). *Summary for Policymakers*, in: Climate Change 2014: Impacts, Adaptation, and Vulnerability. Part A: Global and Sectoral Aspects*.* Cambridge University Press, Cambridge, United Kingdom, and New York, NY, USA, pp. 1–32, 2014.

[8] Carter J, Connelly A, Handley J, Hincks S, Karvonen A. *Deliverable D1.3 Conceptual Framework*. EU H2020 Project RESIN, Manchester, UK, 2016.

[9] German Federal Ministry for Economic Cooperation and Development. *The Vulnerability Sourcebook. Concept and guidelines for standardised vulnerability assessments.* Deutsche Gesellschaft für Internationale Zusammenarbeit, Bonn and Eschborn, Germany, 2014.

[10] IPCC. *Climate Change 2007: Synthesis Report.* Pachauri, R.K and Reisinger, A. (eds.) IPCC, Geneva, Switzerland, 2007.

[11] Rome E, Bogen M, Worst R, Lückerath D, Sojeva B, Voss H, Voss N, Xie J. *Deliverable D2.3 Realisation and implementation IVAVIA.* EU H2020 Project RESIN, Sankt Augustin, Germany, 2017.

[12] IPCC. *Annex II: Glossary*, in: Climate Change: Synthesis Report*.* R.K. Pachauri and L.A. Meyer (eds.) IPCC, Geneva, Switzerland, pp. 117-130, 2014.

[13] Neves A, Blondel L, Brand K, Hendel Blackford S, Rivas Calvete S, Iancu A, Melica G, Koffi Lefeivre B, Zancanella P, Kona A. *The Covenant of Mayors for Climate and Energy Reporting Guidelines*. EUR 28160 EN; doi:10.2790/586693, 2016.

[14] MOVE Indicator Database Visualiser of the FP7 project MOVE. URL, visited on Oct 25, 2017: http://www.gi4drr.org/move/

Modeling and Simulation-based Development of Autonomy Features for Drones

Shihui Chen[1*], Umut Durak[1,2], Sven Hartmann[2]

[1]TU Clausthal, Institute of Informatics; **shihui.chen@tu-clausthal.de*

[2]German Aerospace Center (DLR), Institute of Flight Systems

SNE 28(2), 2018, 55 - 60, DOI: 10.11128/sne.28.tn.10413
Received: April 15, 2018 (Selected ASIM GMMS/STS 2018 Postconf. Publ.), Accepted: May 15, 2018
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. In the last decade, more and more aerial robotics researchers show interests in developing autonomy features for drones to solve problems in different areas. But the development of autonomy features is complex and labor intensive. Accordingly, model-based design and simulation-based verification is becoming an industry standard in development of autonomous airborne systems. This we call modelling and simulation-based development. However, commercial model-based design and simulation tools and supporting testing environments require a considerable amount of investment. In oder to provide a more economic and efficient solution, this paper investigates a pipeline for modeling and simulation-based development of autonomy features for drones using open source software and hardware stacks. In this context, a generic drone architecture is being designed based on open source hardware platforms, namely CC3D and Raspberry Pi. In the software stack, LibrePilot, an open source software suite to control multicopters is extended to support the designed architecture. The design of the autonomy features is developed using the model-based design in Scilab/Xcos. Xcos Reuseable and Customizable Code Generator is utilized for automatic code generation. The software stack will also include a generic plant model. The workflow starts from autonomy feature modeling and ends with flight testing through Model-in-the-Loop (MiL) testing, Software-in-the-Loop (SiL) testing, target deployment, Hardware-in-the-Loop (HiL) testing. The approach is demonstrated with a simple case study about an autonomous landing feature.

Introduction

Overview

An aircraft without a human pilot aboard is called an unmanned aerial vehicle (UAV), commonly known as a drone. A UAV is regarded as an essential part of an unmanned aircraft system (UAS). The other parts of the system are ground control system (GCS) and the communication system between the UAV and GCS. The usage of UAS has increased sharply in the recent years. Varies researchers have developed different autonomy features of drones to solve problems in many fields, such as health care emergency response. Especially in some dangerous situations, UAS that incorporate a high level of autonomy has the ability to accomplish the missions more efficiently without risking lives.

The architecture of autonomous systems is very important. It is regarded as a method to structure the algorithms for creating functionalities. Figure 1 depicts the general autonomy architecture [15] for UAS.

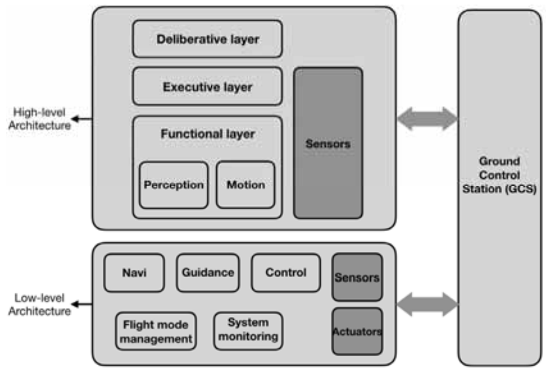


Figure 1. General autonomy architecture for UAS (Adapted from [15]).

The low-level architecture implements the basic functions like navigation and control algorithms that keep the UAV stable in the air and listen to commands from the high level. In this paper, the low-level architecture is realized using OpenPilot CC3D controller [5].

Clough [4] and Merz [13] have elaborated the difference among automatic, autonomous and intelligent systems. An automatic system will exactly do as the programmings say while an autonomous system has the capabilities to make decisions for achieving the missions. An intelligent system can do whatever an autonomous system does and it can produce the goals by its own motivations without any instructions and influence from the outside world. In this paper, system development towards autonomous drones will be discussed.

Modeling and simulation-based development

Model-based design [8] and simulation-based verification are becoming an industry standard in development of autonomous airborne systems. This we call modeling and simulation-based development (Figure 2).

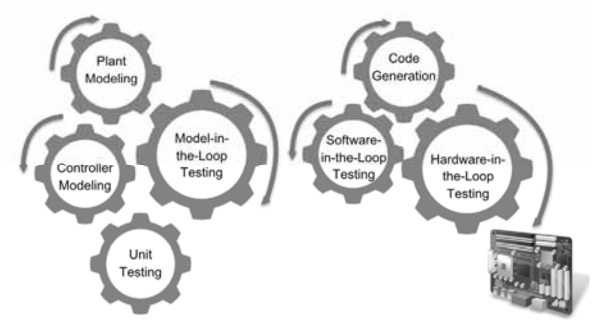


Figure 2. Modeling and simulation based development.

A plant [9] is often desired with a transfer function which indicates the relation between the input signals and the output signals of a system without feedback, commonly determined by physical properties of the system. Usually a plant model is identified by collecting and processing raw data from the real world. We could define the plant model by using mathematical equations or creating a block diagram model that implements known differential-algebraic equations governing plant dynamics. This is called plant modeling.

The mathematical model conceived from the plant is applied to identify dynamic characteristics of the system. According to those characteristics, a control algorithm that can be executed under the condition which the physical processes are controllable is derived and a suitable controller is chosen. The controller has two levels, the supervisory control that determines the mode transition structure and the lowlevel control that decides the time-based inputs to the plant [12]. For UAS, low-level controller corresponds to low-level architecture and supervisory control is a part of high-level architecture.

To verify the system flexibility, we use simulation-based verification. Each unit and subsystem should be tested and finally achieve a Model in-the-Loop (MiL) testing. Then it leads to Code Generation (CG) [7]. Software in-the-Loop (SiL) testing follows CG to verify the generated code by checking its conformance to the model. Then it comes to the Hardware in-the- Loop (HiL) testing where the generate code is tested using the target hardware.

1 Autonomy Feature Development Pipeline

1.1 Architecture

This paper aims at proposing a process to enable developing complex autonomy features for drones by using open source software and hardware.

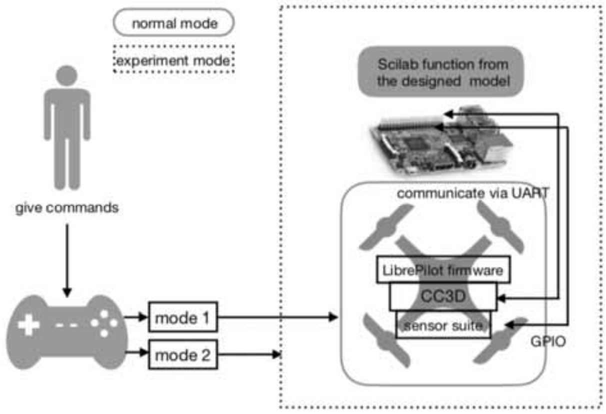


Figure 3. Testbed architecture.

In this paper, regarding their availability, accessibility, cost and flexibility, 250mm class racing drones are used as testbeds. Figure 3 illustrates the testbed architecture. CC3D running the LibrePilot firmware is utilized as the Flight Controller (FC). In the normal mode, users can give command to the Remote Controller (RC) to control the drone. Once the flight mode is switched to experiment mode, Raspberry Pi 3, the target hardware platform of the testbed, will take charge of the controlling while the RC will be disabled. Scilab/Xcos will be utilized as the modelbased design and simulation environment. Xcos Reuseable and Customizable Code Generator [16] is used for generating Scilab scripts from Xcos model. For the use case, the generated Scilab script for the autonomous landing feature is deployed to Raspberry Pi 3. It gets data from ultrasonic distance sensor and CC3D controller, computes the next command and send it to CC3D controller as the new command.

The communication between Raspberry Pi 3 and CC3D is physically established by Universal Asynchronous Receiver/Transmitter (UART) interface where SciPy [11] is used to execute Scilab scripts on target platform through. Related Python libraries are used for interface implementations. UART can control the series device that attached to the computer interface. It provides the computer with the RS-232C Data Terminal Equipment (DTE) interface so that it can "talk" to and exchange data with modems and other serial devices.

1.2 Open source software and hardware stacks

Scilab/Xcos Scilab [3] is a open source software for numerical computation providing a powerful computing environment for engineering and scientific applications. It is a platform to be utilized for model simulation, loading, design, saving and compilation using a graphic editor called Xcos. Xcos has some core features like standards palettes and blocks, model building and modification, model customization and simulation. For some special requirement blocks which are not provided in the Xcos palette browser, users can create their own module by toolbox skeleton to achieve the specific goals.

LibrePilot LibrePilot [17] is an open source research project which focuses on research and development of software and hardware to be utilized for different applications like vehicle control and stabilization, unmanned autonomous vehicles and robotics.

LibrePilot includes hardware and software elements (Figure 4). In the hardware side, UAV is manly controlled by a RC called transmitter. The transmitter has a paired receiver for signal receiving. This receiver also connects with the FC and directly control the actuators. The role of FC is to interpret the control command from RC and runs control algorithm and flight code on the aircraft. If FC is connected to PC where runs LibrePilot Ground Control Station (GCS), users are able to monitor and log flight telemetry data of their vehicle in a real-time environment. This is the software system of LibrePilot which includes GCS software and flight firmware. The flight firmware is implemented in C and C++ using the FreeRTOS [1] embedded real time operating system and typically runs on ARM architecture micro controllers. The communication between GCS and FC is implemented via UAVTalk protocol.

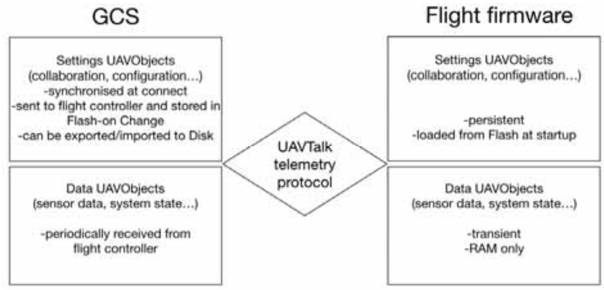


Figure 4. Elements of LibrePilot (adapted from [6]).

UAVTalk is a highly efficient, extremely flexible and completely open binary protocol designed specifically for communication with UAVs [6]. It implements the low level communication between the GCS and the autopilot. It acts as a transportation tool for the data structures defined by the UAVObjects, a data container written in XML format for all of the telemetry data. This protocol does not need to know the details of the data structure, its mission is to send byte arrays and routing received byte arrays to specified object for dealing with the data. For example, all of the RC commands are stored in an UAVObject called ManualControlCommand. Meanwhile, the states of the vehicle can be easily accessed from UAVObjects including accelerate states and attitude states. This is also the way to establish the communication between CC3D and Raspberry Pi 3.

Raspberry Pi Integration. Raspberry Pi [14], series of small single-board computers, could be equipped with operating system. Raspberry Pi 3 Model B is the third generation of Raspberry Pi family. It has the quad core 64bit CPU that has the best performance. To physically connect Raspberry Pi and CC3D, we use the main port of CC3D which can be configured as a serial port and GPIO pin module of Raspberry Pi as it shown in Figure 5. Moreover, To achieve an autonomous system, we utilize an ultrasonic distance sensor to measure distance between the drone and the at ground. An ultrasonic distance sensor transmit from and receive an ultrasonic wave with a single ultrasonic transmitting and receive element to measure proximate distances such as vehicle floor heights or distances to obstacles or pedestrians approaching relative to a vehicle [10]. For the testbed, a low-cost sensor called HC-SR04 is selected. However, The ECHO pin of the sensor is rated at 5V while the GPIO input pins are rated as 3.3V. Therefore two resistors are added to protect the GPIO module. To access the telemetry data in Python, UAVTalk protocol is applied. For instance, below is a code excerpt to get attitude state of the drone.

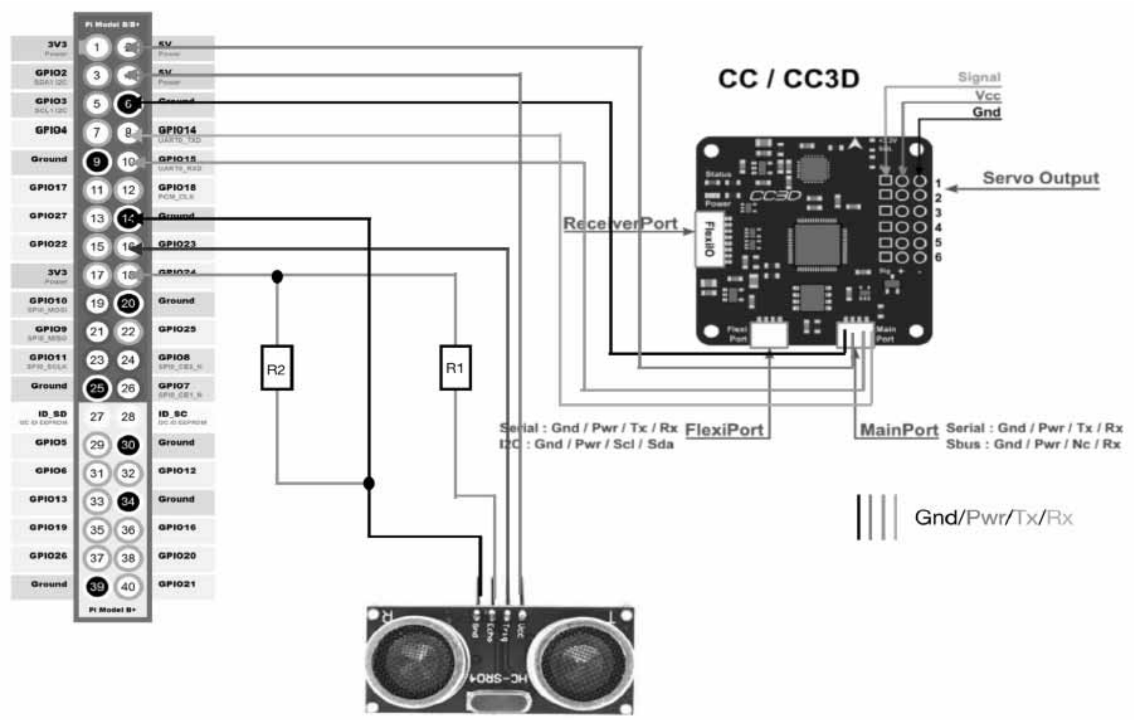


Figure 5. Hardware connection diagram.

```
self.objMan.AttitudeState.metadata.telemetry
    Update-
    Mode=UAVMetaDataObject.UpdateMode.PERIOD
    IC
self.objMan.AttitudeState.metadata.telemetry
    UpdatePeriod.value=50
self.objMan.AttitudeState.metadata.updated()
...
Yaw=self.objMan.AttitudeState.Yaw.value
Pitch=self.objMan.AttitudeState.Pitch.value
Roll=self.objMan.AttitudeState.Roll.value
```

Until here, all of the open source hardwares are integrated together for data communication between the Raspberry Pi 3 and the CC3D.

2 Demonstration

2.1 Workflow

To demonstrate the pipeline that promotes modelling and simulation-based development using open source software and hardware stacks, we developed an autonomous landing controller.

A generic Scilab/Xcos quadcopter model called Generic Quadcopter Simulation (GQS) that employs a proportional-derivative flight controller as a low-level architecture is used as a plant model. GQS model is based on [2].

This generic model can be tailored using parameters to represent a specific platform. The high level architecture is designed in a model-based fashion using Scilab/Xcos. MiL testing is to optimize and verify the controller design for the autonomy feature. Xcos Reuseable and Customizable Code Generator [16] performs as a mean of generating code for the autonomous landing model and to evaluate how good the generated code functions are, a SiL simulation will be tested. For HiL testing, Raspberry Pi 3 executes the code that automatically generated from code generator, A second Raspberry Pi 3 is used as a realtime simulation computer target that enables an UART communication between the plant model and the controller. Finally, to verify the autonomy feature, all of the hardwares stacks is assembled on the drone and flight testing is conducted.

Figure 6 explains the modeling and simulation based workflow applied:

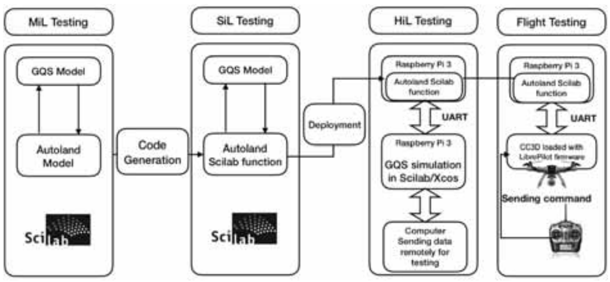


Figure 6. Demonstration workflow.

2.2 Model in-the-loop testing

In order to verify the autonomous landing controller, a MiL simulation (Figure 7) is conducted.

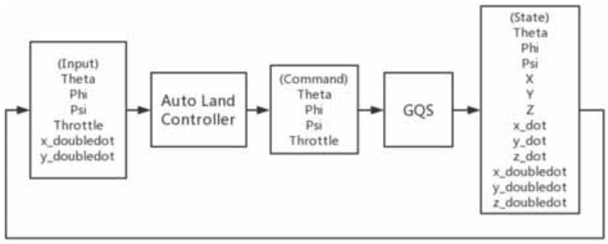


Figure 7. MiL simulation.

We prepared eight scenarios to test that in different situations this system will be working. Figure 8 shows the results for one of the scenarios.

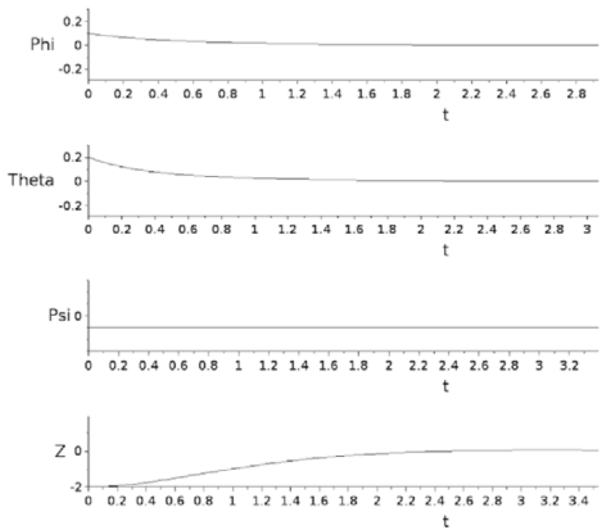


Figure 8. MiL simulation result.

The system can land for all the eight scenarios that means the designed autonomous landing controller is working properly.

2.3 Software in-the-loop testing

After MiL testing, the next step for evaluation is to generate source code out of autonomous landing controller model. As mentioned before, we use Xcos Reuseable and Customizable Code Generator for generating Scilab scripts. The auto generated code is then reintroduced in MiL schema. The same eight scenarios are executed and the results are compared with the MiL results.

2.4 Hardware in-the-loop testing

The HiL testing is essential a further step in testing the autonomous landing feature.

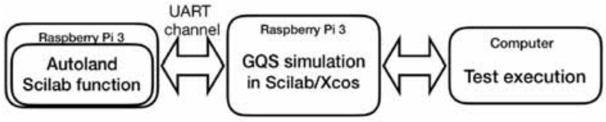


Figure 9. HiL simulation process.

Figure 9 depicts the test setup. The middle Raspberry Pi 3 runs the GQS simulates the drone with controller while the target Raspberry Pi 3 on the left side will execute the Auto Land Scilab function for further verification. It is quite convenient for two Raspberry Pis to establish UART communication by GPIO using RX and TX pins. To achieve UART between Pi and Xcos model, a ATOM toolbox named serial Xcos IO module [18] is used. This module provides a Xcos block to interface real hardware platform for a Xcos simulation via serial ports. It can also be applied into HiL simulation. Originally, the block supports Arduino and provides the bidirectional way to receive C structure input signal from an embedded system and then send back a C structure output signal to the embedded system. On the other hand, to execute autonomous landing Scilab function in Raspberry Pi, a python module called Scilab2Py [19], a mean to seamlessly call Scilab functions and scripts from Python is applied. The result for HiL simulation for the same eight scenarios matched with the MiL results.

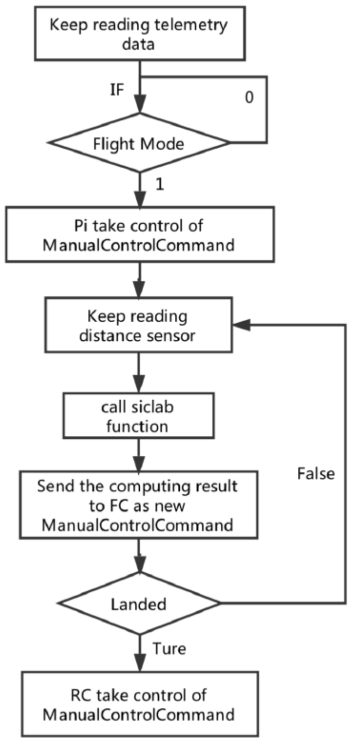


Figure 10. Target execution logic.

2.5 Flight testing

Figure 10 describes the execution logic of auto generated Scilab script on the target platform for achieving the autonomous landing.

Raspberry Pi is listening to the flight mode from FC, once the flight mode changes to experiment mode, Pi will take control of the drone and receive the required system states which are sent to Scilab function for computing the movement of autonomous landing. When landed, RC will take back the control of FC.

Once the code is running on the testbed platform the drone is ready to fly (Figure 11).

Figure 11. Flight testing.

3 Conclusion

Since the usage of drones is sharply growing while the modeling and simulation based development gets popular in many fields as well, it is quite meaningful to investigate a methodology to combine those two.

In this paper, we utilize the free open source software stacks including Scilab/Xcos which serves as a model design and simulation environment and LibrePilot. We also use open source hardware stacks like Raspberry Pi, HC-SR04 sensor and CC3D. They are sold with low price tags in the market and easy to acquire. The result shows that the pipeline is able to be used for simple autonomy feature design. However, since of all the resources are open source and low-end, the whole system does not perform perfectly. For example, the sensor can be easily broken so that the sensed data is not correct. For the future work, we will be using more reliable products with this pipeline to expect better performance.

After building the pipeline for the Simulation and Model-based development of autonomy features for drones using only open source software and hardware, more students who have interests in aerospace domain can take this pipeline as an guidance for developing their own autonomy features.

References

[1] Barry R. Using the FreeRTOS real time kernel: a practical guide. Real Time Engineers, 2010.
[2] Bouabdallah S. Design and control of quadrotors with application to autonomous flying. PhD thesis, 2007.
[3] La Vern Campbell S, Chancelier JP, and Ramine Nikoukhah. Modeling and simulation in Scilab/Scicos. Springer, 2006.
[4] Clough BT. Metrics, schmetrics! how the heck do you determine a uav's autonomy anyway. Technical report, Air Force Research Lab, 2002.
[5] LibrePilot/OpenPilot community. Coptercontrol/cc3d/atom hardware setup, 2016.
[6] LibrePilot/OpenPilot community. Librepilot documentation, 2017.
[7] Erkkinen T. Model style guidelines for flight code generation. In AIAA Modeling and Simulation Technologies Conference, 2005.
[8] Erkkinen T, Potter B. Model-based design for do-178b with qualified tools. In AIAA Modeling and Simulation Technologies Conference and Exhibit, 2009.
[9] Franklin GF, Powell JD, Emami- Naeini A. Feedback control of dynamic systems, volume 3. Addison-Wesley Reading, MA, 1994.
[10] Iwabuchi M, Ohzawa S. Ultrasonic distance sensor, April 17 1990. US Patent 4,918,672.
[11] Jones E, Oliphant T, Peterson P. {SciPy}: open source scientific tools for {Python}. 2014.
[12] Lee EA, Seshia SA. Introduction to embedded systems: A cyber-physical systems approach. MIT Press, 2016.
[13] Merz T. Building a system for autonomous aerial robotics research. In Proc. of the IFAC Symp. on Intelligent Autonomous Vehicles, 2004.
[14] Raspberry Pi—Teach. learn, and make with raspberry pi. Raspberry Pi, 2016.
[15] Viguria A. Autonomy architectures. In: Encyclopedia of Aerospace Engineering. 1–14, 2016. Proc.
[16] https://forge.scilab.org/index.php/p/xcos-code-generator/
[17] https://www.librepilot.org
[18] https://atoms.scilab.org/toolboxes/wgserialxcosio
[19] https://pypi.python.org/pypi/scilab2py

Model-driven Development and Simulation of Integrated Modular Avionics (IMA) Architectures

Björn Annighöfer

Institute of Aircraft Systems, University of Stuttgart; *bjoern.annighoefer@ils.uni-stuttgart.de*

SNE 28(2), 2018, 61 - 66, DOI: 10.11128/sne.28.tn.10414
Received: April 10, 2018 (Selected ASIM GMMS/STS 2018 Postconf. Publ.), Accepted: May 15, 2018
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Presented is a model-driven development method for avionics systems comprising of a domain-specific model, mathematical optimization, and an attached network simulation. For Integrated Modular Avionics (IMA) the degree of freedom in choosing the system architecture is so high that determining the optimum by hand is hardly possible for large aircraft. A domain-specific model was created with the Eclipse Modeling Framework (EMF) holding system requirements and architecture variants, such that it can automatically be validated and evaluated. Moreover, combinatorial optimization is used to determine optimal architectures by algorithm for single and multiple objectives. Optimization on civil aircraft and a space launcher revealed improvements of up to 30% in single design objectives. Moreover, the architecture model can automatically be converted in configuration stubs and an AFDX network simulation.

Introduction

Integrated Modular Avionics (IMA) are state-of-the-art for large civil and military aircraft. The concept of IMA is that computing, memory, and IO resources are shared between several safety-critical and non-critical system functions. System functions are, for instance, cabin pressure control and landing gear retraction. Those are loaded as segregated software partitions. The major portion of the avionics system's hardware is standardized. Computing and IO modules are configured in software to fulfil their purpose in multiple system functions.

IMA reduced the hardware, cost, weight, and space[1,2]. Two challenges arising are first design freedom and second the massive number of configuration parameters. Considering design, questions like, what is the lightest architecture, how many modules do I need, or what is the optimal IO distribution per module, can no more be answered optimally by hand. Considering configuration, current IMA systems require so many parameters that the process became inefficient and error-prone [3, 4, 5].

To improve the situation, a domain-specific model was developed for computer-aided design of avionics architectures. It holds the complete architectures, but in addition the system requirements, like functions, resource needs, and safety constraints. Architectures shall be automatically validated, evaluated and compared. It is generic in terms that no precise avionics technology or function architectures are predefined.

Requiring only a minimum mandatory information makes it applicable to the design phase. Nevertheless, it can be automatically converted to mathematical optimization problems. For instance, function assignment, routing, and module sizing. Moreover, the model is used to derive configuration stubs or simulations automatically. All is implemented in a seamless model-driven IMA design method depicted in Figure 1.

The remainder of this article is organized as follows. Section 1 introduces the domain-specific avionic architecture model. Section 2 explains a multi-objective optimization approach and Section 3 shows an example of an AFDX simulation derived from the model. The article ends with a conclusion and outlook.

1 Avionics Modeling

A standardized avionics system is a distributed computing platform, which provides computing resources and I/Os to aircraft system functions running as software.

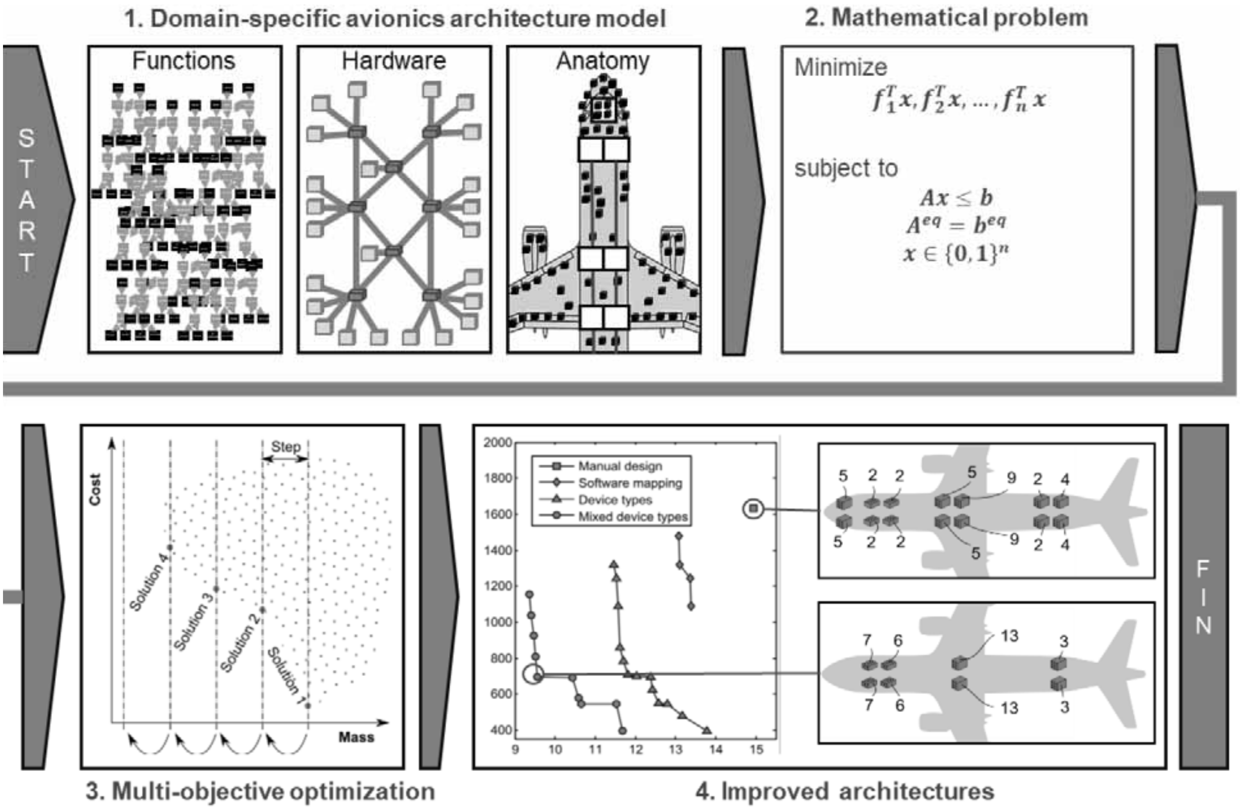


Figure 1. Workflow of the model-driven IMA architecture design method.

Since the number of different module types is kept as small as possible in order to reduce development costs, there is usually a large number of equal hardware modules, which could technically host the same functions. Whether a certain module should host a function or not, often depends on the position, the distance to required sensors and actuators, and safety consideration. Overall, there is a high degree of freedom, e.g., the dimension of modules, the installation locations, the assignment of functions, the network topology, and the routing of signals.

A domain-specific model was developed especially for the purpose of avionics system design. Therefore, it generically captures the capabilities and resources of hardware, without requiring a certain module type or technology. Moreover, it generically covers the software as atomic building blocks, so called tasks, signals, and their resource requirements. In addition, segregation, symmetry, location, and power constraints can be attached to single tasks or task groups.

The first instance of an avionics architecture model [6] was used in several air and space research programs. At the end of 2017, a second generation of the domain-specific model has been finalized [7], which is more, considering bus systems and hardware. It has been made available as Open Source as the **Open Avionics Architecture Model (OAAM)** [8].

OAAM is designed in nine almost independent layers. This matches the concurrent development process of IMA systems, which is distributed over multiple parties. The nine layers are Library, Scenarios, Systems, Functions, Hardware, Anatomy, Capabilities, Restrictions, Mapping.

The four main layers are Function, Hardware, Anatomy, and Mapping.

The **Functions layer** holds all tasks to be assigned to the avionics system and the signals that must be routed. In addition, it includes timing and safety constraints.

The **Hardware layer** allows modelling device instances and interconnection topologies without physical dimensions.

Within the **Anatomy layer**, the installation locations and cable routes of the aircraft including the length and positions are modelled. It is a graph-like representation of a simplified 3-dimensional construction plan.

In the **Mappings layer** assignment objects can be created, which assign task to devices, devices to installation locations, as well as signals and cables. The basic constraint determining if an assignment is valid or not, is a linear **resource provisioning and resource consumption model**. Each task or device K_i requires a set of certain resources r^{K_i} in a certain amount $r_j^{K_i} \in \mathbb{R}_+$, i.e.

$$r^{K_i} = (r_1^{K_i}, \dots, r_n^{K_i}) \tag{1}$$

A resource is an abstract unit of what has to be provided to the task or device and is consumable, e.g. computational power, memory or space in installation location. Devices and locations provide resources. An assignment to D_j is valid as long as the available resources r^{D_j} are not exceeded, i.e.

$$\sum_{K_i \subseteq K} r^{K_i} \leq r^{D_j}. \tag{2}$$

This must hold for all assignments of the architecture. In addition, constraints on devise, locations, power sources, areas, symmetries, and co-locationing control what are valid mappings. Multiple mapping variants of the same elements can be created to represent different architecture variants. For each variant the validity and design objectives are individually be calculated.

Technically OAAM is realized with the ECORE **metamodeling** language of the Eclipse Modelling Framework (EMF) [9], which allows to define formal UMLlike meta-models and automatically derive the implementation, persistence layer, and edit tools. Moreover, extensions exist for the verification and evaluation of EMF derived domain-specific models. OAAM models are edited, validated, and evaluated within a specialized Eclipse instance. In addition, an interface to MATLAB was developed.

2 Avionics Architecture Optimization

The IMA systems of current aircraft have more than 4000 tasks and peripheral as well as more than 50 devices and thousands of possible installation locations and cable routings. The pure number of elements prevents that the design engineer is able to derive the optimal dimensioning, installation, and software assignment by hand. Even if he would do, he would not be able to prove it. Moreover, the optimality of an avionics system is not a unique property.

There are multiple design objectives desired in the design process. Simple examples are cost and mass, which shall both be minimal. Objectives that are more complex are installation cost or maintenance effort. There is usually not a single architecture optimizing all objectives, but objectives are partially contradictional, such that the best tradeoff is desired.

In order to automate and prove the optimality of recurring design tasks, a link between the architecture model and combinatorial optimization was developed. Overall, **eight generic optimization routines** for avionics architecture were developed as depicted in Figure 2. Optimization routines are classified in 1-level , 2-level, and 3-level assignments depending on how many different element types are assigned in parallel.

The most basic optimization routine is **function assignment**, which assigns a set of functions to a given set of devices. The optimization objectives are, for instance, the device weight (i.e. use a few devices as possible) or the wire weight (i.e. put tasks as close as possible to the related sensors and actuators). In addition, all defined constraints for the tasks must hold.

A more complex routine is **device type optimization**, which starts from an empty topology and selects the optimal number, installation, and dimensioning of devices, while assigning tasks.

3-level assignment is the highest level of automation. It derives devices and network, as well as task assignment and signal routing in a single step. It has the highest degree of freedom and it was shown in [10] that in general this leads to the highest optimization potential. However, also the complexity of the optimization problem rises, such that this is only feasible to calculate single systems with no more than 50 tasks. 1-level and 2-level assignments are feasible for scenarios with 4000 and more tasks. Calculation times are between several hours and four weeks.

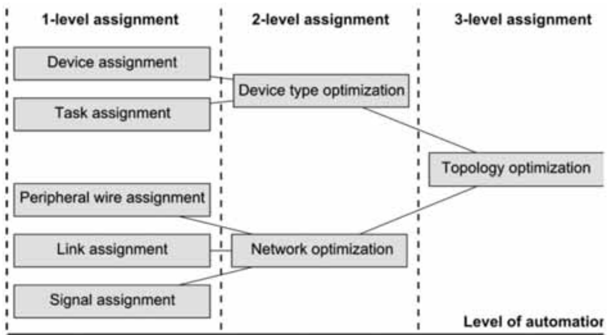


Figure 2. Eight generic optimization routines for avionics

architecture optimization.

Technically, all optimization routines are implemented as multi-objective binary programs (BP). A multi-objective BP searches for a binary solution vector x such that

$$f_1^T x, f_2^T x, \dots, f_n^T x \tag{3}$$

are minimized subject to

$$\begin{aligned} Ax &\leq b \\ A^{eq} &= b^{eq} \\ x &\in \{0,1\}^n. \end{aligned} \tag{4}$$

x encodes - depending on the optimization tine - either task assignments, signal routes, or hardware topologies. The cost vectors $f_1, \dots, f_n$ allow for linear objectives. With the addition of auxiliary variables, also non-linearities can be included. Linear inequalities A and equalities A^{eq} constrain the resource consumption, enforce segregation, and ensure a unique mapping of every object. For more information on the optimization problem formulations please refer to [11, 12, 13, 14, 15].

Information from the domain-specific model are automatically converted to the mathematical optimization problems. The conversion and the invocation of combinatorial optimization are implemented in MATLAB. Commercial-of-the-shelf solvers [16] for Mixed Integer Linear Programming (MILP) are integrated to efficiently solve BPs. The solution is converted back in meaningful model information. In case of multiple-contradicting objectives, a custom iterative multi-objective solver calculates the so-called Pareto optimum [17], i.e. the set of best possible trade-off solutions. See an example of a Pareto optimum in Figure 1 on the lower right.

Several of the optimization routines were used to derive an optimal avionics system for the ARIANE 5 space launcher [18, 19] and for deriving some general scaling laws for avionics architecture design [20]. In all application, up to 30% improvements in single objectives between the manually derived architectures and the optimized architectures were found.

3 AFDX Simulation

The avionics architecture model presented above is static. For design, validation, and evaluation static properties and capacities are assumed that have sufficient safety margins such that the architecture should also be valid during operation. For instance, during architecture design the CPU is modelled as a static resource, i.e. the percentage of CPU load consumed by each task vs. 100% available CPU. For sure, however, within the real system, most tasks are periodic and must be scheduled and the schedule must be valid in terms of the individual deadlines. This can be considered by assuming, for instance, a maximum load of 70% during design. Nevertheless, real schedules have to be determined and their correctness has to be proven.

Another example is the network. During design and optimization, bandwidth is assumed as a single consumable resource. Each signal has a bandwidth portion it consumes. However, in the real systems, network messages have to be scheduled and maximum transfer delays have to be proven. Typically, this happens with network calcus or network simulations.

A common network for IMA systems is **Avionics Full Duplex Switched Ethernet (AFDX)**. AFDX is an asynchronous switched network based on Ethernet. AFDX messages address virtual links (VL) instead of target devices.

VLs are preconfigured and static communication routes in the network. Each VL is assigned a maximum bandwidth (Bandwidth Allocation Gap - BAG). If too large or too many messages are sent, those are dropped by the switches. An AFDX network and the corresponding message routing is assumed to be valid if for each VL a valid BAG can be found and if under worst case conditions the maximum delay and jitter requirements for each signal are met. It is common to validate this in simulations.

Worst case conditions, delay, and jitter measurements cannot be made in the OAAM model. However, a simulation framework for AFDX was developed within MATLAB/SIMULINK. The simulation framework provides basic components for switches, as well as the AFDX send and receive interfaces of avionics devices. In addition, it is able to run on a rapid prototyping system, which is able to output the virtual communication on a real AFDX interface.

All components of the simulation framework can be used to create virtual AFDX networks by hand and validating the timing of all messages up to an accuracy of 1 ms. More common is, however, to derive the AFDX simulation model automatically from the OAAM static architecture, which – if a mapping was calculated – includes all necessary information. Figure 3 shows an example of an Airbus A380-like AFDX network completely derived from an architecture model. It was possible to run the simulation in realtime on a rapid prototyping system.

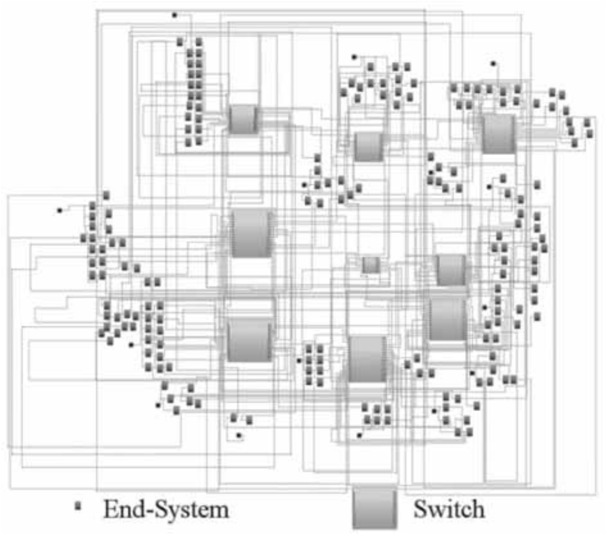


Figure 3. An A380-like AFDX system simulated in MATLAB/SIMULINK.

The detailed description of the AFDX simulation and the results can be found in [21].

4 Conclusion

Current avionics systems are generic resource-sharing distributed computing systems. The most common representatives are Integrated Modular Avionics (IMA). A domain-specific model for avionics architectures and system function requirements enables computer-aided design of avionics systems including automated validation, evaluation, and optimization. Eclipse EMF and MILP solvers are used as technologies. Optimization of real aircraft and the ARIANE 5 space launcher revealed improvements of up to 30% in single objectives as weight, compared to the manual design. The applicability of the model during system development is extended by the possibility to derive dynamic simulations, which was shown for a MATLAB/SIMULINK AFDX simulation.

In future works, it is intended to extend the capabilities of the model-driven avionics architecture tool chain by additional optimization routines. Moreover, it is worked on a self-configuring avionics system, which utilizes OAAM as its online knowledge base.

References

[1] Ramsey JW. “Integrated modular avionics: Less is more - approaches to IMA will save weight, improve reliability of A380 and B787 avionics,” Avionics magazine, 2007. [Online]. Available: http://www.aviation today.com/av/categories/commercial/8420.html

[2] Wilkinson C. “IMA aircraft improvements,” Aerospace and Electronic Systems Magazine, IEEE, vol. 20, no. 9, pp. 11–17, September 2005.

[3] Butz H. “Open integrated modular avionic (IMA): State of the art and future development road map at airbus deutschland,” in Proceedings of the 1st International Workshop on Aircraft System Technologies, March 2007, pp. 211–222.

[4] Watkins C, Walter R. “Transitioning from federated avionics architectures to integrated modular avionics,” in Digital Avionics Systems Conference, 2007. DASC ’07. IEEE/AIAA 26th, oct. 2007, pp. 2.A.1–1 –2.A.1–10.

[5] Wilson A, Preyssler T. “Incremental certification and integrated modular avionics,” in 27th Digital Avionics Systems Conference, 2008.

[6] Annighöfer B , Kleemann E, Thielecke F. “Model-based development of integrated modular avionics architectures on aircraft-level,” in Deutscher Luft- und Raumfahrtkongress, Bremen 27. - 29. Sept. 2011, no. 1395. Bremen: Deutsche Gesellschaft für Luft- und Raumfahrt, September 2011.

[7] Annighöfer B, Riedlinger M, Marquardt O. “How to tell configuration-free integrated modular avionics what to do?!” in 36th Digital Avionics System Conference, Saint Petersburg, FL , USA, September 2017.

[8] www.oaam.de

[9] www.eclipse.org/modeling/emf/

[10] Annighöfer B, Thielecke F. “A systems architecting framework for optimal distributed integrated modular avionics architectures,” CEAS Aeronautical Journal, pp. 1–12, 2015.

[11] Annighöfer B. “Model-based architecting and optimization of distributed integrated modular avionics,” Dissertation, Hamburg University of Technology, March 2015. ISBN: 978-3-8440-3420-2

[12] Annighöfer B, Kleemann E, Thielecke F. “Automated selection, sizing, and mapping of integrated modular avionics modules,” in 32st Digital Avionics System Conference, Syracuse, NY, USA, October 2013.

[13] Annighöfer B, Thielecke F. "Supporting the design of distributed integrated modular avionics systems with binary programming," in Deutscher Luft- und Raumfahrtkongress, Berlin 10. - 12. Sept. 2012. Berlin: Deutsche Gesellschaft für Luft- und Raumfahrt, September 2012.

[14] Annighoefer B, Thielecke F. "Multi-objective mapping optimization for distributed modular integrated avionics", in 31st Digital Avionics System Conference, Williamsburg, VA, USA, October 2012.

[15] IBM CPLEX or GUROBI

[16] Annighoefer B, Reif C, Thielecke F. "Network topology optimization for distributed integrated modular avionics", in 33rd Digital Avionics System Conference, Colorado Springs, CO, USA, October 2014.

[17] Ehrgott M. Multicritia Optimization, 2, Ed. Springer Berlin Heidelberg, 2005.

[18] Annighoefer B, Nil C, Sebald J, Thielecke F. "Structured and symmetric ima architecture optimization: Use case Ariane launcher," in 34th DASC, 2015.

[19] Annighöfer B, Çelen Nil, Sebald J, Thielecke F. "Ariane-5-based studies on optimal integrated modular avionics architectures for future launchers," in 6th EUCASS, 2015.

[20] Annighöfer B, Posternak V, Thielecke F. "Empirical investigations on avionics scaling laws," in 35th Digital Avionics System Conference, Sacramento,CA , USA, September 2016.

[21] Annighöfer B, Ihle H, Thielecke F. "An easy-to-use realtime AFDX simulation framework," in 35th Digital Avionics System Conference, Sacramento,CA , USA, September 2016.

Performance Modelling of a Computer Integrated Manufacturing and Management System

Jozef B. Lewoc[1*], Antoni Izworski[2], Slawomir Skowronski[2]

[1]Design, Research and Translation Agency Leader (Leading Designer)
53-138 Wroclaw, ul. Powst. Sl. 193/28, Poland; **leader@provider.pl*

[2]Wroclaw University of Technology, 50-370 Wroclaw, Wybrzeze Wyspianskiego 27, Poland

SNE 28(2), 2018, 67 - 74, DOI: 10.11128/sne.28.tn.10416
Received: December 2012 (Draft); Revised 2014 (Corrected);
Accepted: October 20, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper presents a predicted target architecture of an integrated manufacturing and management system, based on a metropolitan-type infranet and an industrial process control and monitoring network. Such systems are severely required by prospective users, especially large-scale manufacturing enterprises. Nevertheless, neither big manufacturers of computer integrated manufacturing systems nor big manufacturers of computer integrated systems are apt to develop combined computer integrated manufacturing and management systems. Considering that, a team of volunteer design and research workers initiated some work oriented toward facilitating future development of such combined systems. Since the combined systems will be novel ones, prior results of research work conducted for the manufacturing systems or the management systems separately will not be valid, since the actual requirements will have to cover the needs of both manufacturing and managerial applications. This refers to, among other things, to performance evaluation. To evaluate performance, a method developed for packet switching networks with end-to-end acknowledgement was applied. The network is modelled as a set of closed routes consisting of a user and a series of service stations (communication links, switches, host processes). The paper describes the investigations carried out for the case study. Some consideration is given to the performance evaluation accuracy, basing on the validation work results obtained from analytical work, simulation and measurements on the Polish pilot wide area network.

Introduction

The Computer-Integrated Manufacturing and Management (CIMM) systems are severely needed by prospective users, especially big and medium manufacturing enterprises. In spite of that, big manufacturers and vendors of Integrated Management (IM) systems, having at their disposal financial resources sufficient to develop and implement at least pilot CIMM systems, are no apt to enter the manufacturing domain [1,2]. The basic reason for that seems to be the fact that the big IM system manufacturers and vendors do not possess at their disposal the designers and implementers educated and experienced enough to cope with both manufacturing and management issues. Such persons are available, paradoxically many of them are available in poorly developed countries where the designers must have possess a wide scope of experience in order that they are successful [3], but they are dispersed in various industrial and/or research organisations and they do not have at their disposal the financial resources needed to develop novel IT systems of the CIMM type.

In such circumstances, a reasonable duty of research and development workers seems to be to carry out the initial work on the prospective project of CIMM systems, even if no financing is available for such work. This work may be considered as a volunteer work or a work for the society that have financed their earlier research and development work.

A team of IT experts and process technologists was established and worked out an approach to development of CIMM systems that seems to be feasible, rather fast and economic [4,5,6,7]. The next step that can and should be made by the research and development workers even before the CIMM project is established is that of performance evaluation of probable hardware structures.

The task of performance evaluation is important in the case of CIMM since the project concerns a novel IT system for which nobody possesses the sense of performance and no rules of thumb concerning performance could have been developed earlier. The performance evaluation issues connected with development of CIMM systems is the subject matter of the present paper.

The authors believe that a specific in-depth example is more educational than a general study, in particular in the context described above, and, therefore, the form of a case study has been adopted for the present paper.

The following section presents the architecture of the CIMM system under investigation.

1 Exemplary CIMM Architecture

The pilot Polish CIMM system has been designed several years ago for the biggest then Polish manufacturer of household appliances, Polar, Wroclaw (the Enterprise), employing several thousand people. The CIMM issues were analysed for the Enterprise already in the early nineties, but the project was abandoned because of the severe down economy period in Poland in the nineties. In conformity with the early design work, it was assumed that the target organisational structure of the Enterprise would be that depicted in Figure 1.

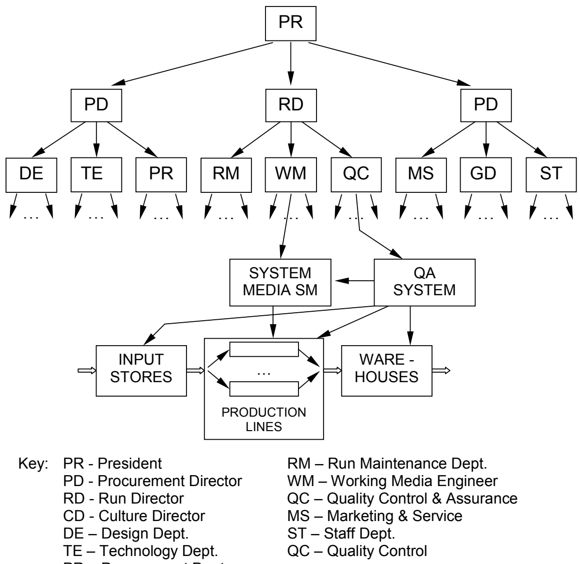


Figure 1. General organisational diagram of the enterprise.

For the general Enterprise organisation presented in Figure 1, the hardware architecture of the CIMM system, depicted in Figure 2, has been proposed.

The architecture is based on the infranet metropolitan network operating under the TCP/IP protocol suite [8] and on the process control and monitoring network of the LonWorks type [9].

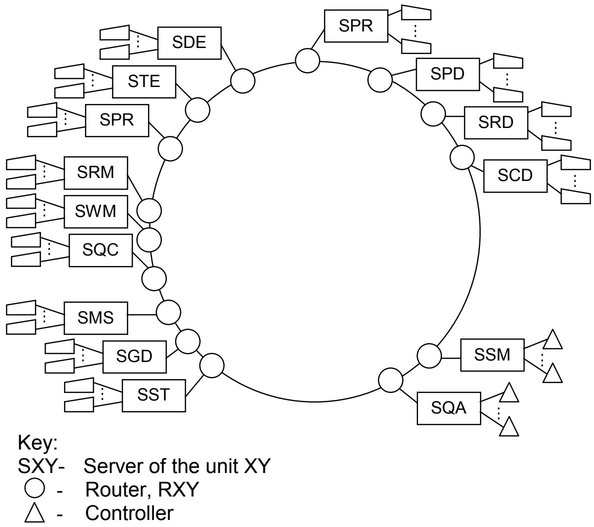


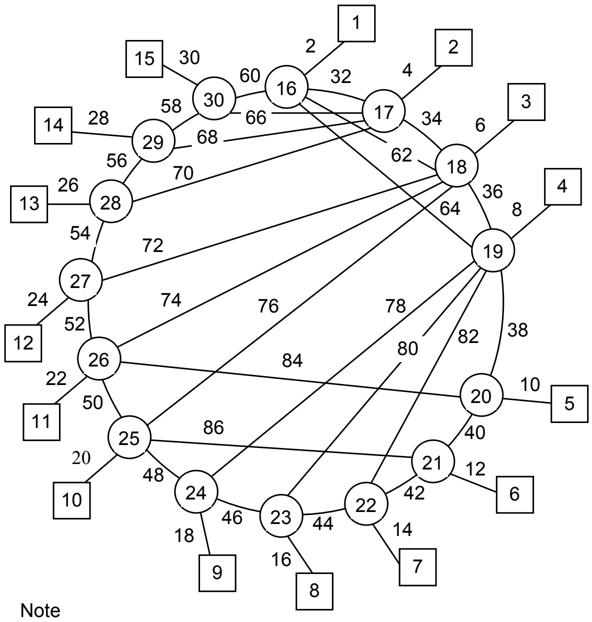
Figure 2. Hardware architecture of the CIMM system

2 The Network under Investigation

It is assumed that the time needed for transfer of data via any switch (server, gateway, node or router) is insignificant in comparison with the time needed to transfer the data via a co-operating data link. This is in conformity with the specifications of actual hardware proposed for the CIMM system since data is transferred via network switches as 8-bit byte blocks transferred via fast direct memory access (DMA) channels and internal processing of the data is usually connected only with the message headers.

However, the method proposed enables to consider the delays in network switches though the calculations may be somewhat more complicated.

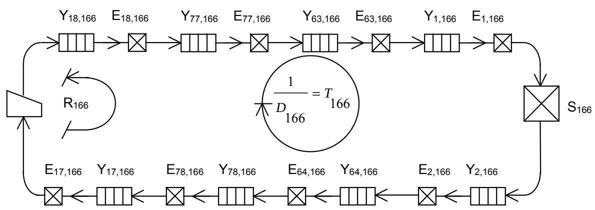
With the assumption that the delays in transmission links are insignificant, the hardware structure of the CIMM system may be presented as in Figure 2. The network depicted in Figure 2 is composed of 30 switches (15 routers and 15 servers). Let the individual switches and data links be numbered as shown in Figure 3 (server Nos: 1- 15, router Nos: 16-30, link pairs (for both transmission directions): (1,2) – (85,86).



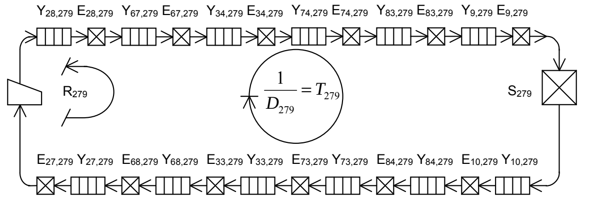
Note
Link number, i =2k, k=1,2,..,43 is, in fact, a pair of numbers, 2k and 2k-1; the first is the number of the link directed from the node of the lower number to that of the higher one and the other in the opposite direction.

Figure 3. Numbering of links and nodes.

Upon the network of Figure 3, there is stretched a set of closed routes. Exemplary closed routes in the CIMM network are the connections between a Marketing & Service Department worker and the general enterprise database (the President's database) (closed route No. s = 166; ref. Figure 4a) and between a technologist and the system Media database (closed route No. $s = 279$; ref. Figure 4b).



a) Exemplary closed route of a Marketing & Service employee using general enterprise database



b) Exemplary closed route of a technologist using the System Media

Key:

$$s \in S = 1, \dots, 299;\ i \in I = 1, \dots, 86$$

Rs = round route (trip) delay for s-th closed route
T_s = s-th closed route cycle time
D_s = s-th closed route throughput
$Y_{i,s}$ = waiting time of s-th closed route to i-th link
$E_{i,s}$ = service time of s-th closed route to i-th link
M_s = s-th closed route thinking time
S_s = s-th closed loop ultimate service time

Note: Upper case letters denote random variables while lower case letters – relevant mean values

Figure 4. Exemplary closed routes.

Note: it is assumed that any p-th server, $p \in 1, \dots, 15$ (ref. Figure 3) is able to support 20 closed routes of numbers $s = (p-1)\,20, \dots, (p-1)\,20 + 19$.

Note also that the network described here as a set of closed routes may be considered as a case of the Kelly networks [5] where the entity (packet) sojourn time in any switch is a function of the number of packets in all routes in the network. However, the classic queuing theory does not provide solutions for the Kelly networks [10].

In theory, the problem may be solved by the stochastic queuing system methods. Unfortunately, the stochastic queuing system methods are not suitable for actual network designers and, even more, closed networks always produce severe problems [11].

Another class of performance evaluation methods are Petri nets, either deterministic or stochastic [12]. However, Petri nets are only formalisms and do not provide anything of merits, not covered by the basic classic or stochastic performance evaluation methods.

Therefore, it was decided to adapt the approximate performance evaluation method [13], developed and validated earlier for computer and communication networks with end-to-end acknowledgement, to the cases of the CIMM system and to apply the method for performance evaluation for the case study under discussion.

3 The Approximations

The reference [13] is hardly available now. Therefore, some basic assumptions and results will be repeated in the present paper.

In addition to the denotations defined in Figure 3, the following will be used in the present paper:

A_i is the set of closed routes beginning at the i-th link, such that the $s(i)$-th user is connected directly to the i-th link:

$$A_i = \{s_{i,1}, s_{i,2}, ..., s_{i,a_i}\} \tag{1}$$

B_i is the set of closed routes passing via the i-th link but not beginning at that link:

$$B_i = \{z_{i,1}, z_{i,2}, ..., z_{i,b_i}\} \tag{2}$$

The link preceding the i-th link in the s-th closed route is denoted by $h = hs(i)$, provided that $s \in B_i$.

The set of closed routes parallel to the s-th closed route in the i-th link is defined by (3).

$$C_i = B_i \cap (A_h \cup B_h) \tag{3}$$

The power of C_i is given by (4).

$$\overline{C_i} = c_i \tag{4}$$

The mean cycle time of closed routes encountered by the s-th route at the i-th link is approximated by (5).

$$\bar{t}_{i,s} = \frac{a_i + b_i - 1}{\sum\limits_{l \in \{A_i \cup B_i\}/\{s\}} \frac{1}{t_i}}; \frac{0}{0} = 0; \tag{5}$$

The mean waiting time of the closed routes encountered by the s-th route at the i-th link is given by (6).

$$\bar{y}_{i,s} = \frac{\bar{t}_{i,s}}{(a_i + b_i - 1)} \left(\sum_{l \in \{A_i \cup B_i\} \setminus \{s\}} \frac{y_{i,l}}{t_i} \right); \tag{6}$$

The mean service time for the closed routes encountered by the s-th route at the i-th link is given by (7).

$$\bar{e}_{i,s} = \frac{\bar{t}_{i,s}}{(a_i + b_i - 1)} \left(\sum_{l \in \{A_i \cup B_i\} \setminus \{s\}} \frac{e_{i,l}}{t_l} \right); \tag{7}$$

The mean thinking time for the closed routes encountered by the s-th route at the i-th link is given by (8).

$$\bar{m}_{i,s} = \frac{\bar{t}_{i,s}}{(a_i + b_i - 1)} \left(\sum_{l \in \{A_i \cup B_i\} \setminus \{s\}} \frac{e_{i,l}}{t_l} \right); \tag{8}$$

The definitions of the mean values defined above for the set of closed routes that are not parallel to the s-th route are as follows:

$$\bar{t}'_{i,s} = \frac{a_i + b_i - c_i}{\sum\limits_{l \in \{A_i \cup B_i\} \setminus C_i} \frac{1}{t_i}}; \tag{9}$$

$$\bar{y}'_{i,s} = \frac{\bar{t}'_{i,s}}{a_i + b_i - c_i} \sum_{l \in \{A_i \cup B_i\} \setminus C_i} \frac{y_{i,l}}{t_i}; \tag{10}$$

$$\bar{e}'_{i,s} = \frac{\bar{t}'_{i,s}}{a_i + b_i - c_i} \sum_{l \in \{A_i \cup B_i\} \setminus C_i} \frac{e_{i,l}}{t_l}; \tag{11}$$

$$\bar{m}'_{i,s} = \frac{\bar{t}'_{i,s}}{a_i + b_i - c_i} \sum_{l \in \{A_i \cup B_i\} \setminus C_i} \frac{m_{i,l}}{t_l}; \tag{12}$$

For all closed routes, the balance equations (13) have been defined. The basic reasons for the balance equations are as follows.

The first set of equations in (13) is obvious: the cycle time, t_s, for any closed loop is a sum of the thinking time, m_s, all waiting times, $y_{i,s}$, and all transmission times, $e_{i,s}$, for the closed loop involved (ref. the examples in Figure 4).

For the two other equation sets in (13), the unknown mean values are approximated by the mean values of uniformly distributed variables and the probabilities that any entity (packet) is in any specific state (thinking, waiting for transmission, transmission) is approximated by the mean duration for that state divided by the mean cycle time for the variable under consideration. Another assumption is that there may exist one and only one entity (packet) in any closed loop of the network.

The second equation set in (13) refers to the case of an entity (packet) of the s-th closed route beginning at the i-th link. This entity may find there an entity of any closed route passing via the i-link with the probability $(a_i+b_i-1)\dfrac{\bar{e}_{i,s}}{\bar{t}_{i,s}(1-\frac{e_{i,s}}{t_s})}$ (the latter dividend describes the condition that the entity (packet) under consideration is not in the state of transmission).

The expression in the parenthesis in the second equation set of (13) is a sum of three terms, denoted here by $e_{i,s}(A + B + C)$. After multiplying, the first term is the mean value of the time from the instant that the entity of the s-th closed loop finds the entity in transfer till the instant that the latter transmission is completed, i.e. $\frac{\bar{e}_{i,s}}{2}$ in accordance with the assumptions accepted.

The second term approximates the queue that may have been gathered during the time interval that the entity (packet) found in transfer via the i-th link (there are $(a_i + b_i - 2)$ eligible candidates), provided that the possible candidate is not in the thinking state (the quotient (1- $\frac{\overline{m}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}}$)).

The third term approximates the queue that may have gathered during the time interval from the instant that the entity encountered in the i-th link has been in transfer till the instant that the entity of the s-th closed loop appears at the i-th link (there are $((a_i + b_i) - 1)$ eligible candidates to be multiplied by the mean probability the candidate is in the waiting state ($\frac{\overline{y}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}}$) provided that it is not in the state of thinking $(1 - \frac{\overline{m}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}})$.

The third equation set in (13) describes the routes passing via the i-th link but not beginning there. It is approximated that any loop parallel to the s-th one is of no impact on the waiting time spent by any entity of the s-th closed loop at the i-th link (the complete share of the parallel loops in the waiting time of the loop under consideration is included at the beginning link). Therefore, the averaging in this equation set is done only for the closed loops that are non-parallel to the one under consideration (formulae (9)-(12)). Except of that, this set of equations is analogous to the second one (note that in this case, the minimum value of c_i equals to 1 (formulae (3), (4)).

The set of equations (5) – (13) enables to compute iteratively the basic unknown mean values of the network performance, i.e. the closed route cycle time, t_s, the mean throughput in the s-th closed route, $d_s = \frac{1}{t_s}$, and the round-trip delay, $r_s = t_s - m_s$.

$$\mathrm{I}_{s \in S} (t_s = \sum_{i \in I_s} (y_{i,s} + e_{i,s})); \qquad (13)$$

where S is the set of all closed loops, $\overline{\overline{S}} = v$.

$$\mathrm{I}_{s \in A_i} (y_{i,s} = \frac{(a_i + b_i - 1)\overline{e}_{i,s}^2}{\overline{t}_{i,s}(1 - \frac{e_{i,s}}{t_s})} (\frac{1}{2} + $$

$$+ (a_i + b_i - 2)(1 - \frac{\overline{m}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}}) \frac{\overline{y}_{i,s}}{\overline{t}_s - \overline{e}_{i,s}} +$$

$$(a_i + b_i - 1)(1 - \frac{\overline{m}_{i,s}}{\overline{t}_{i,s} - \overline{e}_{i,s}}) \frac{\frac{\overline{e}_{i,s}}{2}}{\overline{t}_{i,s} - \overline{e}_{i,s}});$$

$$\mathrm{I}_{s \in B_i} (y_{i,s} = \frac{(a_i + b_i - c_i)\overline{e}'^{\,2}_{i,s}}{\overline{t}'_{i,s}(1 - \frac{e_{i,s}}{t_s})} (\frac{1}{2} +$$

$$(a_i + b_i - c_i - 1)(1 - \frac{\overline{m}'_{i,s}}{\overline{t}'_{i,s} - \overline{e}'_{i,s}}) \frac{\overline{y}'_{i,s}}{\overline{t}'_{i,s} - \overline{e}'_{i,s}} +$$

$$+ (a_i + b_i - c_i)(1 - \frac{\overline{m}'_{i,s}}{\overline{t}'_{i,s} - \overline{e}'_{i,s}}) \frac{\frac{\overline{e}'_{i,s}}{2}}{\overline{t}'_{i,s} - \overline{e}'_{i,s}});$$

4 Validation of Approximations

Validation with accurate results for cyclic queuing systems

The problem of performance evaluation of homogeneous star-topology networks is the same as that investigated for cyclic queuing networks. The accurate solutions in the form of limit probabilities of the entity (packet, request) numbers have been known since early seventies. These results, for exponential thinking time, were employed to validate the approximations presented above.

Several hundred comparisons were done for the number of the customers (closed routes), v, changing between 2 and several dozen. The relative error of the mean cycle time, t_s, calculated for the set of comparison reached its maximum less than 0.03 at $v \approx 14$ and decreased for v tending to zero or to infinity.

Validation with simulation

In order that the approximations may be validated, a fast WAN simulator was developed on the basis of earlier simulation studies for real-time computer control systems. This simulator was used, primarily, for the Polish pilot wide area network MSK. However, both actual and planned network configurations (including those to operate at much higher transmission rates) were investigated. In addition, available foreign simulation results for local and wide area networks were also used for validation.

The number of validation experiments was higher that 500, with the number of closed loops and links equal up to 500 and 100, respectively. The maximum relative error found was lower than 0.08. However, in the case of results obtained for MSK, where the fair conditions of comparison could be ensured (this, obviously, was not true for foreign simulation investigation), the maximum relative error of the cycle time (and throughput) was lower than 3.5.

Validation with measurements

To investigate MSK and the approximations presented above, an internal communication network measuring tool Sitwa was developed and implemented.

It was used, primarily, to validate the simulation results on the existing possible configurations of MSK. The investigations showed that the simplifying assumptions in simulation (e.g. omission of the flow control packets and/or frames) did not result in significant simulation errors. The maximum relative error of the approximations under discussion did not exceed 0.05.

5 Exemplary Results

The set of equations (5) ÷ (13) was solved iteratively. The basic user characteristics, r_s and d_s, for the exemplary closed routes, $s = 166$ and $s = 279$, are presented in Figure 5 ÷ 8.

The closed route, $s = 166$, passes via the most severely loaded links, $i = 1$ and $i = 2$. This results in that r_{166} reaches more than 4 s at m_s values close to 0 (ref. Figure 5). If it is assumed that an annoying value (i.e. the value that can not be accepted by the Enterprise employees) of rs is that higher than 2 s, then the m_s values below some 0.5 s should be avoided.

s	I_s	s	I_s
1÷18	{Ø}	**160**	16,79,35,76,83,11.12,86,75,36,80,15
19	2,62,74,83, 9,10,84,73,61,1	**161÷178**	18,77,63,1,2,64,78,17
20	2,62,76,85,11,12,86,75,61,1	**179**	18,77,35,74,83,9,10,84,73,36,78,17
21÷38	4,31,1,2,32,3	**180**	18,77,35,76,85,11,12,86,75,36,78,17
39	4,34,74,83,9,10,84,73,33,3	**181÷198**	20,75,71,2,1,62,76,19
40	4,34,76,85,11,12,86,75,33,3	**199**	20,75,74,83,9,10,84,73,76,19
41-58	6,61,1,2,62,5	**200**	20,85,11,12,86,19
59	6,74,83,9,10,84,73,5	**201-218**	22,73,61,2,1,62,72,21
60	6,76,85,11,12,86,75,5	**219**	22,83,9,10,84,21
61÷78	8,63,1,2,64,7	**220**	22,73,75,85,11,12,86,75,72,21
79	8,35,73,83,9,10,84,74,36,7	**221-238**	24,71,61,2,1,62,72,23
80	{Ø}	**239**	24,71,74,83,9,10,84,73,72,23
81÷98	{Ø}	**240**	24,71,76,85,11,12,86,75,72,23
100	{Ø}	**241÷258**	28,69,31,1,2,32,70,25
101÷118	{Ø}	**259**	26,69,34,74,83,9,10,84,73,33,70,25
119	{Ø}	**260**	26,69,34,76,85,11,12,86,75,33,70,25
120	{Ø}	**261÷278**	28,67,31,1,2,32,68,27
121÷138	14,81,63,1,2,64,82,13	**279**	28,67,34,74,83,9,10,84,73,33,68,27
139	14,81,35,74,83,9,10,84,73,36,82,13	**280**	28,67,34,76,85,11,12,86,75,33,68,27
140	14,81,35,76,85,11,12,86,75,36,82,13	**281÷298**	30,65,31,1,2,32,66,29
141÷158	16,79,63,1,2,64,80,15	**299**	30,65,34,74,83,9,10,84,73,33,66,29
159	16,79,35,74,83,9,10,84,73,36,80,15	**300**	30,65,34,76,85,11,12,86,75,33,66,89

Table 1. I_s versus s

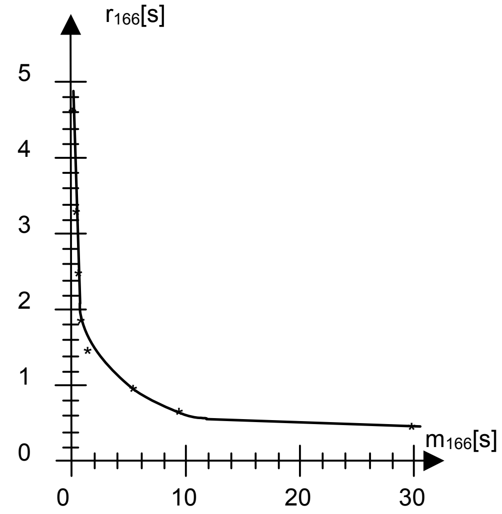


Figure 5. r_{166} versus m_{166}.

The d_{166} values (the mean throughput values depicted in Figure 6) show a definite saturation (congestion) at m_s values below ca. 0.5 s. If the mean offered load for the 166-th closed route is defined by (14).

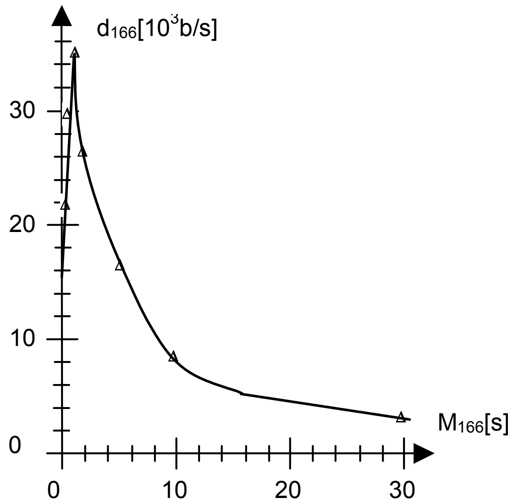


Figure 6. d_{166} versus m_{166}.

$$t_{166}^{off} = \sum_{i \in 166} (e_{i,s} + m_s) \quad (14)$$

Then the increased offered load results in the decreased mean throughput, d_{166}, for the given scenario and for $m_s < 0.5\, s, s = 1, \ldots, 300$.

The closed route, $s = 279$ (Figure 7), shows only the acceptable r_s values (below 1 s while the values not greater than 2 s are acceptable as not annoying ones) and the ds value is decreasing monotonically with the offered load increasing ($m_s \rightarrow 0, s = 1, \ldots, 300$) (Figure 8). The reason is that the 279-th closed route does not pass via any link under heavy traffic. Therefore, even when the mean thinking time is decreased towards zero, no saturation (congestion) occurs in the links passed by the 279-th closed route.

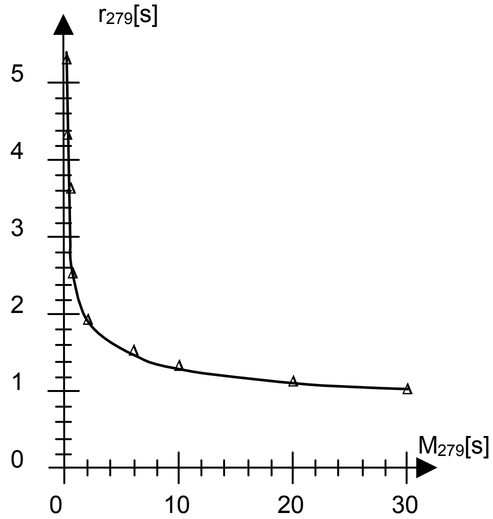


Figure 7. r_{279} versus m_{279}

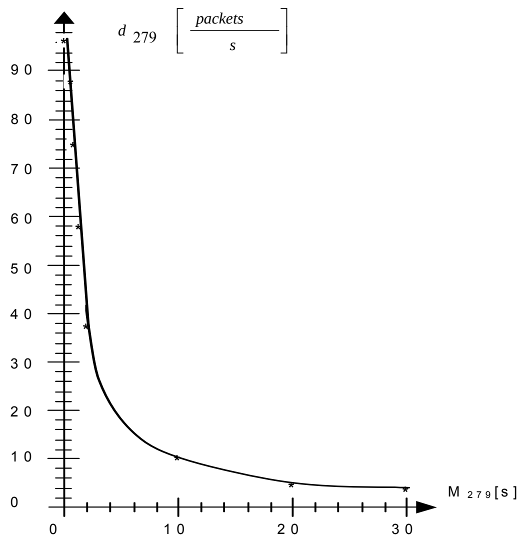


Figure 8. d_{279} versus m_{279}

6 Final Remarks

The authors do not claim that the method devised and verified some dozen years ago needs no further work. On the contrary, they do realise that the method validated primarily on the pilot Polish Interuniversity Computer Network MSK, operating at rather low transmission rates and in a limited configuration, needs further validation and possible tuning. In addition, the method may and should be upgraded: thanks to the rapid growth of the computing power that has happened during the last decade, more accurate approximations may be done for the balance equations (e.g., the simplifying assumption that an entity (packets) may find only one entity of one closed loop in the queue to some link) may be discarded. Then the balance equations become more complicated but may be still solved by the iterative method. And the networks with sliding windows or credit-based flow control may be modelled directly, instead of modelling several loops for credits bigger than 1, as it has been done earlier.

However, the authors believe that the method may and should be published in its original form now. Any protest against it, raised by the queuing theory experts pointing at some possible formal deficiencies of the method, would sound rather false: in a general case, the classic queuing theory methods have failed and have not produced any useful tools for performance evaluation of actual computer and communication networks, severely needed by network designers, developers and operators. The authors are designers and/or consultants for actual computer systems and networks. To do their jobs in a proper way, they had to get involved in some research work in the domain of performance evaluation. Any further work is, however, within the duty range of the queuing theory people.

References

[1] Leader, Technical correspondence, Wroclaw, 2006.
[2] Ridler R. Proposal for CIM, Technical correspondence of Leader, Wroclaw, 2000.
[3] Blach LK, et al. In West-East approaches of automation: A case study, Cultural Aspects of Automation ’91 (IFAC), Krems 1991.
[4] Franasik L, et al. System Media – a Step toward Computer Integrated Manufacturing and Management systems, *Modelling, Measurement and Control (AMSE Press)*, Vol. 22, No. 3: 37-46, 2001.
[5] Izworski A, Lewoc JB, Skowronski S. Development of Computer Integrated Manufacturing and Management Systems, In: *Information Control Problems in Manufacturing*, Saint Etienne, 2006.
[6] J.B. Lewoc, A. Izworski and S. Skowronski, ALSIS case study: computer integrated manufacturing and management systems, In *Applications of Large Scale Industrial Systems*, (IFAC), Helsinki-Stockholm, 2006.
[7] J.B. Lewoc, Performance Evaluation Problems for Actual Computer Networks, AMSE Monographs, Tassin, 1990.
[8] http://www.acm.org/crossroads/xrds1-1/tcpjmy. ht ml
[9] http://www.echelon.com
[10] Robertazzi TG. Computer Networks and Systems. Queueing Theory and Performance Evaluation, Springer-Verlag, 2000.
[11] Surfozo R. Introduction to Stochastic Networks, Springer-Verlag, 1999.
[12] German R., Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets, Wiley, 2000.
[13] Lewoc JB. Performance Evaluation Problems for Actual Computer Networks, AMSE Monographs, Tassin, 1990.

MbedTarget - A Simulink Target for Cortex-M Microcontrollers

Olaf Hagendorf

HS Wismar, Research Group CEA; *olaf.hagendorf@hs-wismar.de*

SNE 28(2), 2018, 75 - 80, DOI: 10.11128/sne.28.sw.10417
Received: April 15, 2018 (Selected ASIM GMMS/STS 2018 Postconf. Publ.), Accepted: May 15, 2018
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The MATLAB/Simulink add-ons for C and C++ code generation, MATLAB Coder, Simulink Coder and Embedded Coder, are a widely used and well established technologies for rapid prototyping, model based design, real-time simulation and similar technologies. Mathworks provides the prerequisites for code generation for many Simulink blocks and MATLAB functions. Missing is the direct support of peripheral functions like digital and analog input/outputs, communication functionalities and other microcontroller features when the goal is to execute a Simulink model within a small embedded system. To extend the above coders to support these functions, several add-on toolboxes exists, delivered by Mathworks itself or by third-party suppliers. In the following, the programming of microcontrollers is shortly introduced and their resource utilization is compared, starting with assembler up to code generation by a Mathworks coder products. After introducing and comparing different Cortex-M coder toolboxes, the MbedTarget toolbox as an open source alternative is presented. With examples and code snippets, the principle of the code generation process, the work flow of MbedTarget together with the principle to define a Simulink block is shown.

Introduction

MATLAB/Simulink is widely used in the engineering education, among others in control theories. Other topics of the student education are microcontroller (MCU) programming and the integration of these into physical processes. Main tools for implementing the algorithms are C and C++ compiler.

The interaction with physical systems makes knowledge about control theory often necessary. But the media break between Simulink at one side and C and C++ programming at the other side can make this complicated.

With MATLAB, Simulink and Embedded Coder, toolboxes to create C and C++ code out of Simulink models or MATLAB programs are provided.

To combine both topics, the direct support of peripheral functions like digital and analog input/outputs and other MCU features to interact with a physical system within a Simulink model is necessary. For this, additional packages to extend the coder toolboxes exist, as MATLAB or third-party Add-Ons.

After a short introduction of MCU programming variants in Chapter 1 and their resource comparison, the next chapter describes a few coder toolboxes. In chapter 3, the new MbedTarget is introduced as an alternative coder toolbox.

1 Cortex Microcontroller

Cortex MCUs are 32- and 64bit microprocessors, divided into three subfamilies Cortex-M, -R and -A. They are licensed by ARM Holding. The licensees are producing MCUs with the Cortex microprocessor core but company specific peripheral components.

1.1 Cortex-M family overview

Members of the Cortex-M family are 32bit MCUs with a broad range of processing power, memory sizes, peripheral components, etc. Since the introduction of the first variant, the Cortex-M3 core was released in 2005 and first silicon products were sold in 2006 [1], other variants with more or less powerful cores, e.g. Cortex-M7 and -M0, were introduced. The Cortex-M MCUs have been become a widely used technology, a huge amount of manufacturers are delivering variants. The distributor mouser [2] has ca. 6800 variants of Cortex-M MCUs in its catalog. At all, it lists ca. 40000 MCUs.

1.2 Programming

The following subchapters are shortly introducing and comparing typical embedded programming principles.

Assembler

Since the invention of microprocessors, assembly language has been used [3]. To have the full control over them and their periphery, it is still common to use assembly language. The importance of it decreased over the time in comparison to other languages, but it is still under the 10 most used [4].

C/C++ with HAL library

Many MCU producing companies not only provide the silicon chip but also deliver software libraries. Experiences show that unlike with previous 8 and 16bit processors, the programming of current 32bit MCUs without these libraries considerably increases the overhead. As an example, STM32Cube by STMicroelectronics (STM) [5] is introduced. The software consists of two parts:

- **STM32CubeMX** A tool for C code generation for initialization of hardware functions, adding and configuring of middleware libraries like TCP/IP stack, RTOS, USB, file systems etc. and finally the generation of project files for several integrated development environments (IDEs).
- **STM32 MCU** packages These packages implement a hardware abstraction layer (HAL) to provide standardized API calls for all STM MCU families. An additional part of the packages are middleware libraries like TCP/IP stack, RTOS, USB, file systems and graphical libraries.

Mbed library

The Mbed project was started in 2005 by two ARM employers, Simon Ford and Chris Styles, who are helping out in undergraduate and after-school projects. Both were not satisfied with the current situation and developed the idea to ease the MCU development [6]. Among others, main ideas of this are [7]:

- open source MCU debugging and programming
- hardware
- online toolchain object oriented HAL with adaption layer to different company specific HAL libraries

The object oriented HAL is a C/C++ MCU software platform containing object oriented peripheral and library APIs, C adaption layers and startup code for 232 MCUs [7].

Additionally, several Python tools are part of Mbed, providing functions for compiling, testing library management, exporting project files to several IDEs etc.

Mbed significantly reduces the effort of getting started with ARM based MCUs. A drawback is the amount of additional software layers which leads to an increase of binary sizes. The possibility to compile the same code for different MCUs is advantage and a disadvantage at the same time.

Data flow oriented programming

This programming paradigm, i.e. visual programming, has some advantages in comparison to the above mentioned imperative, text based languages [8, 9]:

- the depiction of a real world object succeeds particularly well
- challenging concepts of imperative languages like variables and dynamic data structures are not relevant
- it provides parallel execution without the need of explicit instruction
- the clarity eases prototypical implementation

Conclusion

As a comparison of the different programming principles, the hello world of MCU programming, a LED blinking application, is compared in its resource usage (sizes of read-only and read/write memory) [10]. The effort to program the application, can only be qualitatively estimated. The effort decreases from the usage of assembler language, over a general C and HAL programming and Mbed. The data flow oriented programming will produce the lowest cost in terms of time to create the application. But it also has the highest resource utilization as can be seen in Table 1.

2 MATLAB/Simulink Targets for Cortex-M Controller

Simulink is a widely used data flow oriented development tool. To program MCUs with it, beside the MbedTarget, introduced in this paper, several other targets exist. They are provided by the Add-Ons manager of MATLAB, or are available as third-party tools. Three of these: Embedded Coder Support Package for STMicroelectronics Discovery Boards [11], Simulink Coder Support Package for STMicroelectronics Nucleo Boards [12] and STM32-MAT/TARGET [13] are introduced in the following.

Programming principle	Flash size (byte)	RAM size (byte)
assembler	88	0
C without library	716	1632
MCU specific HAL	1392[1] 2852[2]	1032[1] 1032[2]
Mbed HAL	22576[3] 37716[4]	1432[3] 8484[4]
Data flow oriented[5]	5893[2]	8060

[1] low layer library used [5]
[2] high layer library used [5]
[3] without RTOS
[4] with RTOS (default config.)
[5] MbedTarget v1

Table 1. Resource comparison of programming principles using the blinking example [10].

2.1 Embedded Coder Support Package for STMicroelectronics Discovery Boards

STM32 Discovery kits are low cost solutions for the evaluation of STM32 Cortex-M MCUs by STM. Beside the MCU and the necessary infrastructure to use it, the kits contains additional peripheral items like displays and sensors. Mouser [2] lists 20 kits in its catalog, the support package can handle three of them: STM32F746G-DISCO, STM32F769I-DISCO and STM32F4-Discovery.

For the STM32F4-Discovery the old, no longer supported standard peripheral library is used, the other two boards are using the Mbed library.

The only supported compiler is GNU GCC.

For the STM32F4-Discovery board, Simulink blocks for analog in, digital in/out, audio in/out and Interrupt handling are available. All MCU pins are usable.

For the other boards, blocks for analog in, digital in/out, serial communication, timer, TCP, UDP and audio in/out are available. The MCU pin usage is restricted to 22 pins.

The supported MCU hardware functionality for all three variants is very basic, the pin usage is partly restricted.

To generate, compile and link the C Code of a Simulink model a single step – Deploy to Hardware – is necessary.

The package requires the Embedded Coder toolbox.

Figure 1 shows an example, an alternating digital pin.

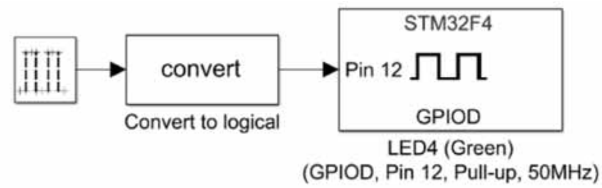


Figure 1. Example with Embedded Coder Support Package for STM Discovery Boards.

2.2 Simulink Coder Support Package for STMicroelectronics Nucleo Boards

STM32 Nucleo kits are the lowest cost solutions for the evaluation of STM32 Cortex-M MCUs by STM. Beside the MCU and the necessary infrastructure to use it, the kits does not contain additional peripheral items except a few kits with Ethernet connectivity.

Mouser [2] lists 38 kits in its catalog, the support package supports 9 of them: Nucleo-F401RE, Nucleo-F103RB, Nucleo-F302R8, Nucleo-F031K6, Nucleo-L476RG, Nucleo-L053R8, Nucleo-F746ZG, Nucleo-F411RE and Nucleo-F767ZI.

For all boards the Mbed library is used.

The only supported compiler is GNU GCC.

For all boards, Simulink blocks for analog in, digital in/out, serial communication and timer are available. The MCU pin usage is restricted to 22 pins. The supported MCU hardware functionality is very basic.

To generate, compile and link the C Code of a Simulink model a single step – Deploy to Hardware – is necessary.

The package requires the Simulink Coder toolbox. Figure 2 shows an example, an alternating digital pin.

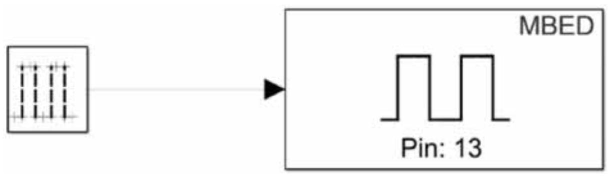


Figure 2. Example with Simulink Coder Support Package for STM Nucleo Boards.

2.3 STM32-MAT/TARGET

This packages is provided by STM. It is based on-STM32Cube. The Simulink package supports allSTM32 MCUs.

All compiler supported by STM32CubeMX can beused: EWARM, Keil MDK V4 and V5, TrueSTUDIO,SW4STM32 and GNU GCC.

Blocks for analog in/out, digital in/out, serial communication, timers, watchdogs are provided. The block set uses the code generation utility STM32CubeMX. Because of this, the supported MCU hardware functionality is considerably large.

To generate, compile and link the C Code of a Simulink model, two steps are necessary: executing the Simulink function: Deploy to Hardware and project building in the chosen IDE.

The package requires the Embedded Coder toolbox. Figure 3 shows an example, an alternating digital pin.

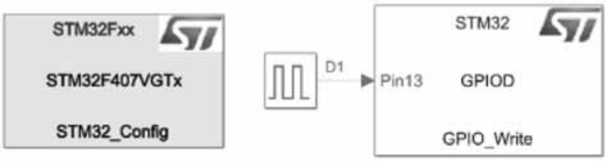


Figure 3. Example with STM32-MAT/TARGET.

2.4 Conclusion

Whereas the provided functionality of the support packages introduced in Chapter 2.1 and Chapter 2.2 is very basic, the other package offers nearly as much functions as the MCU hardware provides.

All packages are free of charge. But the main disadvantage of all is: they are closed source. Own extensions are difficult to realize or even not possible. The supported MCU depends on the manufacturer, mainly STM32 products are supported. Mathworks and third parties offers some more targets supporting e.g. NXP KL25Z and K64F, Infineon XMC, Nordic Semiconductor NRF51 and BBC micro:bit. Although the number of Cortex-M MCUs is huge, the selection of Simulink targets and supported MCUs is restricted.

3 MbedTarget

MbedTarget is completely open from both sides: all code and configuration files i.e. MATLAB code, templates, configurations etc. as well as all MCU libraries are available free of charge and as source codes. It is based on MbedOS 5, therefore all MbedOS 5 compatible microcontrollers can be used to run Simulink models. MbedTarget uses internally GNU GCC to compile the generated source code. Additionally, project files for all Mbed supported IDEs can be generated. With these project files, the generated code can be manually compiled and/or debugged when the Simulink model does not run or own blocks are developed.

The Simulink block library contains two groups of blocks:

- for Mbed functionality
- for sensors, actors, .. based on additional libraries

The first group contains blocks for analog in/out, digital in/out, user LEDs, user buttons, serial communication, timers, Ethernet, RTOS etc.

The second group contains blocks for external analog to digital converter, digital to analog converter, external digital in/out chips, several sensors and actors like temperature, pressure, motion, magnetometer, range, motor control, display, etc. Additionally, MCU specific blocks for encoder, input capture, random number generator and counter are provided.

MbedTarget supports both available coder toolboxes with Simulink model code generation capabilities, Simulink Coder and Embedded Coder.

To generate, compile, link and flash the C Code of a Simulink model a single step – Deploy to Hardware – is necessary.

Figure 4 shows an example, a blinking LED.

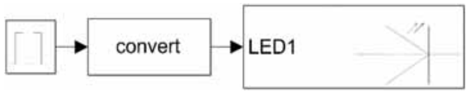


Figure 4. Example with MbedTarget.

3.1 Principle of MbedTarget code generation

The principle workflow when processing a Simulink model to an executable binary is shown in Figure 5.

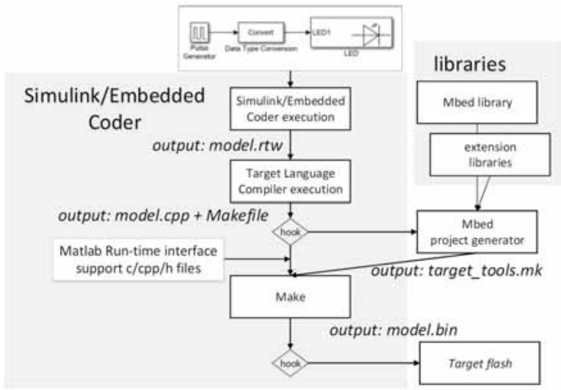


Figure 5. Principle of the MbedTarget code generation.

Simulink or Embedded Coder are generating an rtw file from the model. That is a specific textual representation of the model. Together with tlc files (target language compiler files) which are part of MATLAB and of the specific Simulink target, the target language compiler (TLC) generates several c/cpp/h files and a Makefile meanwhile the process can be influenced by hook calls.

MbedTarget uses the code generation principle based on tlc files: mbed.tlc for the Embedded Coder or mbed_grt.tlc for Simulink Coder as the starting points.

The TLC offers several hooks to customize the code generation process i.e. the transformation process from rtw to c/cpp files. Mainly two hooks are used by MbedTarget: *before_calling_make* for code handling processes and *after_calling_make* for the compiling and flashing process.

The whole procedure can be summarized by the following steps:

1. The TLC uses the current MATLAB working path to store all generated files in a folder with a name constructed using the Simulink model name and 'slprj', e.g. blinky_slprj. During this process Simulink tlc files where used for each standard Simulink block, for each non-Simulink block another tlc file is provided by MbedTarget. Beside the c/cpp/h files generated out of tlc files, also a makefile is created based on template makefiles mbed.tmf or mbed_grt.tmf. This makefile needs an additional make include file, generated within the hook function, described in step 2.
2. After step 1, during the hook call *before_calling_make*, a target specific folder is created. Into this folder, all generated files are copied and a Mbed specific make include file is generated. This include file is very similar to a standard gcc makefile used by Mbed.
3. After executing the hook *before_calling_make* make is called and a bin file is created.
4. During the hook call *after_calling_make,* the bin file is flashed to the target MCU.

3.2 MbedTarget Simulink block usage

This chapter describes shortly the components of an MbedTarget Simulink block. The Digital Output block is chosen because it is one of the simplest blocks. It has only a single input port to write a digital value to MCU pin and is depicted in Figure 6.

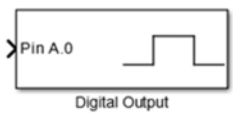


Figure 6. MbedTarget Digital Output block.

With a double click onto the block, a configuration dialog opens as shown in Figure 7. The block has 4 parameters which has to be configured in the dialog:

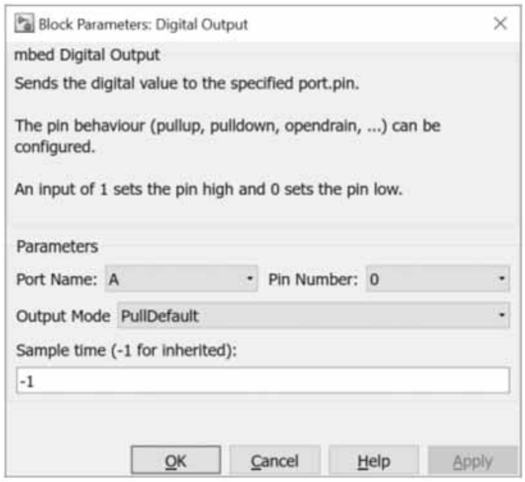


Figure 7. Parameter dialog box.

- Port Name and Pin Number to choose a digital port, e.g. PA0
- Output Mode for pull up, pull down, open drain, … - options are corresponding to Mbed options of the DigitalInOut C++ class for digital in/out pins.
- Sample time defines the time period, how often the digital value is written to the MCU pin. The value has to be a multiple of the global step size.

3.3 MbedTarget Simulink block creation

To implement custom Simulink blocks, the following elements are necessary:

- a block mask
- a S-function
- a tlc file consisting of a mixture of target language and C/C++ code

The *block* mask defines the outlook of the block as shown in the Simulink model editor, defines the input items available in the block mask dialog. It also defines block title, block help text and the help menu entry.

The block parameters connect the block with a *S-function*, i.e. a binary mexw64 file implemented in C, and a tlc file witch has the identical name. In MbedTarget, the S-function mainly describes the behavior of the block: number and type of input and output ports, checks parameters and prepares the transfer of these to the TLC.

A *tlc file* is a mixture of tlc code, a script like language, and C/C++ code in a form of snippets. The script code controls the usage of the C/C++ snippets and how and where they are put into a generated C/C++ file. The tlc file contains mainly three functions: *Setup*, *Start* and *Output*.

Setup controls the inclusion of additional headers and source files, where *Start* is executed once to generate initialization code and *Output* is called once in every simulation loop. Even if the name is output, the function has to handle also input values, send by Simulink to input ports.

In the following, snippets of the *Start* function are described in detail as an example:

```
1. %assign nPort-
   Name=LibBlockParameterValue(PortName,0)
2. %assign nPin-
   Num=LibBlockParameterValue(PinNumber,0)
3. %assign
   pname="P"+FEVAL("char",nPortName+64)
4. %assign
   pname=pname+"_"+FEVAL("int2str",nPinNum-1)
```

Lines 1 and 2 fetches the variables entered in the block mask dialog. Lines 3 and 4 creates from these a pin name, e.g. PA_0 as used for STM32 MCUs.

```
5. %assign name = FEVAL("strrep",
   LibGetFormattedBlockPath(block),"/","_")
6. %assign name = FEVAL("strrep",name," ","_")
7. %assign name = FEVAL("strrep",name,"-","_")
```

Lines 5 to 8 create a unique name based on the complete block name, e.g. *blinky_Digital_Output* for a block *Digital Output* in a model *blinky*. The Simulink path contains characters, e.g. '/', spaces and '-', which are replaced by '_' to create a valid C identifier.

```
8.  %openfile declbuf
9.  DigitalInOut %<name>(%<pname>);
10. %closefile declbuf
11. %assign srcFile = LibGetModelDotCFile()
12. %<LibSetSourceFileSection(srcFile, "Decla-
    rations", declbuf)>
```

Finally, lines 8 to 12 create a single line in the declaration section of the generated source file, shown in line 13:

```
13. DigitalOut blinky_Digital_Output(PA_0);
```

The line 13 is the necessary Mbed code to create a digital output.

A few more lines in the tlc file, using the same principle, are creating the remaining, necessary C code.

3.4 MbedTarget main function

The MbedTarget supports the single task model of Simulink. To create the main function, the target contains template file: mbed_srmain.tlc and mbed_grt_main.cpp. The creation of multithreaded application by Simulink is not yet supported, but can be done manually with MbedTarget RTOS blocks.

4 Conclusion

MbedTarget was developed to improve and ease the MCU usage in several modules like robotics, sensor/ actor systems and embedded control systems of student education. A common characteristic of all these modules is the intensive usage of MATLAB/Simulink in the theoretical part. The practical implementation of the theoretical knowledge is time consuming and errorprone when using typical programming languages like C/C++. By using MbedTarget, the media break between Simulink and C programming could be removed. The disadvantage of an increased resource usage does not play a role in the prototypical implementations in this application field, enough MCU resources are available without problems.

MbedTarget is published at GitHub [14] and will be continuously developed there.

References

[1] Yiu J. Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors. Newnes, UK, 2013.

[2] https://www.mouser.de (accessed 01.2018)

[3] Kusswurm D. Modern X86 Assembly Language Programming. Springer, USA, 2014.

[4] https://www.tiobe.com/tiobe-index/ (accessed 12.2017)

[5] STM32Cube: http://www.st.com/en/embedded - software/stm32cube-mcu-packages.html (accessed 12.2017)

[6] https://os.mbed.com/handbook/Foundersinterview (accessed 12.2017)

[7] https://www.mbed.com/en/ (accessed 12.2017)

[8] Keltsch C. Ein visuelles Programmiersystem zur Modellierung deskriptiver Untersuchungen in der Datenanalyse. Diplomarbeit, Diplomica Verlag, 1998.

[9] Maydl W. Komponentenbasierte Softwareentwicklung für datenflußorientierte eingebettete Systeme. Dissertation, University Passau, Germany, 2005.

[10] https://github.com/ATM-HSW/ASIMFachtagung2018 (accessed 01.2018)

[11] https://de.mathworks.com/hardwaresupport/st-discovery-board.html?s_tid=AO_HS_info (accessed 01.2018)

[12] https://de.mathworks.com/hardwaresupport/st-nucleo.html?s_tid=AO_HS_info (accessed 01.2018)

[13] http://www.st.com/content/st_com/en/products/develop ment-tools/software-developmenttools/stm32-software-developmenttools/stm32-utilities/stm32-mat-target.html (accessed 01.2018)

[14] https://github.com/ATM-HSW/mbed_target (accessed 01.2018) 130 Proc.

SNE Simulation News

EUROSIM Data and Quick Info

EUROSIM 2019

10th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 1 – 5, 2019

Contents

Simulation Notes Europe SNE is the official membership journal of EUROSIM and distributed / available to members of the EUROSIM Societies as part of the membership benefits.

If you have any information, announcement, etc. you want to see published, please contact a member of the editorial board in your country or the editorial office. For scientific publications, please contact the EiC.

This *EUROSIM Data & Quick Info* compiles data from **EUROSIM** societies and groups: addresses, weblinks, and officers of societies with function and email, to be published regularly in **SNE** issues. This information is also published at EUROSIM's website *www.eurosim.info*.

SNE Reports Editorial Board

EUROSIM Emilio Jiménez, *emilio.jimenez@unirioja.es*
Andreas Körner, *andreas.koerner@tuwien.ac.at*
Miguel Mujica Mota, *m.mujica.mota@hva.nl*
ASIM A. Körner, *andreas.koerner@tuwien.ac.at*
CEA-SMSG Emilio Jiménez, *emilio.jimenez@unirioja.es*
CROSSIM Vesna Dušak, *vdusak@foi.hr*
CSSS Mikuláš Alexík, *alexik@frtk.utc.sk*
DBSS M. Mujica Mota, *m.mujica.mota@hva.nl*
FRANCOSIM Karim Djouani, *djouani@u-pec.fr*
HSS András Jávor, *javor@eik.bme.hu*
ISCS M. Savastano, *mario.savastano@unina.it*
LIOPHANT F. Longo, *f.longo@unical.it*
LSS Yuri Merkuryev, *merkur@itl.rtu.lv*
PSCS Zenon Sosnowski, *zenon@ii.pb.bialystok.pl*
RNSS Y. Senichenkov, *senyb@dcn.icc.spbstu.ru*
SIMS Esko Juuso, *esko.juuso@oulu.fi*
SLOSIM Vito Logar, *vito.logar@fe.uni-lj.si*
UKSIM A. Orsoni, *A.Orsoni@kingston.ac.uk*
KA-SIM Edmond Hajrizi, *info@ka-sim.com*
MIMOS Paolo Proietti, *roma@mimos.it*
ROMSIM Marius Radulescu, *mradulescu@ici.ro*
Albanian Society Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*

SNE Editorial Office /ARGESIM

→ *www.sne-journal.org*, *www.eurosim.info*
office@sne-journal.org, Andreas Körner, (info, news)
eic@sne-journal.org, Felix Breitenecker (publications)
✉ SNE Editorial Office, Andreas Körner c/o ARGESIM / Mathematical Modelling & Simulation Group, TU Wien /101, Wiedner Hauptstrasse 8-10, 1040 Vienna , Austria

EUROSIM Federation of European Simulation Societies

General Information. **EUROSIM**, the Federation of European Simulation Societies, was set up in 1989. The purpose of **EUROSIM** is to provide a European forum for simulation societies and groups to promote advancement of modelling and simulation in industry, research, and development. → *www.eurosim.info*

Member Societies. **EUROSIM** members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation. At present **EUROSIM** has 16 *Full Members* and 2 *Observer Members*, and one member candidate.

ASIM	Arbeitsgemeinschaft Simulation *Austria, Germany, Switzerland*
CEA-SMSG	Spanish Modelling and Simulation Group *Spain*
CROSSIM	Croatian Society for Simulation Modeling *Croatia*
CSSS	Czech and Slovak Simulation Society *Czech Republic, Slovak Republic*
DBSS	Dutch Benelux Simulation Society *Belgium, Netherlands*
FRANCOSIM	Société Francophone de Simulation *Belgium, France*
HSS	Hungarian Simulation Society; *Hungary*
ISCS	Italian Society for Computer Simulation *Italy*
KA-SIM	Kosovo Simulation Society, *Kosovo*
LIOPHANT	LIOPHANT Simulation Club *Italy & International*
LSS	Latvian Simulation Society; *Latvia*
PSCS	Polish Society for Computer Simulation *Poland*
MIMOS	Italian Modelling and Simulation Association, *Italy, Observer Member*
RNSS	Russian National Simulation Society *Russian Federation*
ROMSIM	Romanian Society for Modelling and Simulation, *Romania, Observer Member*
SIMS	Simulation Society of Scandinavia *Denmark, Finland, Norway, Sweden*
SLOSIM	Slovenian Simulation Society *Slovenia*
UKSIM	United Kingdom Simulation Society *UK, Ireland*

EUROSIM Board / Officers. EUROSIM is governed by a board consisting of one representative of each member society, president and past president, and representatives for **SNE** Simulation Notes Europe. The President is nominated by the society organising the next **EUROSIM** Congress. Secretary, Secretary to the Board, and Treasurer are elected out of members of the board.

President	Emilio Jiménez (CAE-SMSG), *emilio.jimenez@unirioja.es*
Past President	Esko Juuso (SIMS) *esko.juuso@oulu.fi*
Secretary	M. Mujica Mota (DBSS), *m.mujica.mota@hva.nl*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
Secretary to the Board	Andreas Körner *andreas.koerner@tuwien.ac.at*
Webmaster	I. Husinsky, *irmgard.husinsky*@tuwien.ac.at
SNE Representative	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

SNE – Simulation Notes Europe. **SNE** is a scientific journal with reviewed contributions as well as a membership newsletter for **EUROSIM** with information from the societies in the *News Section*. **EUROSIM** societies are offered to distribute to their members the journal **SNE** as official membership journal. **SNE** Publishers are **EUROSIM**, ARGESIM and ASIM.

SNE Editor-in-Chief	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

→ *www.sne-journal.org*,

office@sne-journal.org

EUROSIM Congress. **EUROSIM** is running the triennial conference series **EUROSIM** Congress. The congress is organised by one of the **EUROSIM** societies.

EUROSIM 2019, the 10th EUROSIM Congress, will be organised by CAE-SMSG, the Spanish simulation society, in La Rioja, Logroño, Spain, July 1 – 5, 2019.

Chairs / Team EUROSIM 2019
Emilio Jiménez, EUROSIM President, *emilio.jimenez@unirioja.es* Juan Ignacio Latorre, *juanignacio.latorre@unavarra.es*

→ *www.eurosim.info*

EUROSIM Member Societies

ASIM
German Simulation Society
Arbeitsgemeinschaft Simulation

ASIM (Arbeitsgemeinschaft Simulation) is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria. ASIM was founded in 1981 and has now about 600 individual members, and 90 institutional or industrial members.

→ *www.asim-gi.org* with members' area

info@asim-gi.org, admin@asim-gi.org

ASIM – Inst. f. Analysis and Scientific Computing
Vienna University of Technology (TU Wien)
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

ASIM Officers	
President	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*
Vice presidents	Sigrid Wenzel, *s.wenzel@uni-kassel.de*
	T. Pawletta, *thorsten.pawletta@hs-wismar.de*
	A. Körner, *andreas.koerner@tuwien.ac.at*
Secretary	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	A. Körner, *andreas.koerner@tuwien.ac.at*
Treasurer	Anna Mathe, *anna.mathe@tuwien.ac.at*
Membership Affairs	S. Wenzel, *s.wenzel@uni-kassel.de*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
Repr. EUROSIM	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
	A. Körner, *andreas.koerner@tuwien.ac.at*
Internat. Affairs – GI Contact	O. Rose, *Oliver.Rose@tu-dresden.de*
	N. Popper, *niki.popper@dwh.at*
Webmaster	I. Husinsky, *irmgard.husinsky@tuwien.ac.at*
Editorial Board SNE	T. Pawletta, *thorsten.pawletta@hs-wismar.de*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
Web EUROSIM	A. Körner, *andreas.koerner@tuwien.ac.at*

Last data update June 2018

ASIM is organising / co-organising the following international conferences:

- ASIM Int. Conference 'Simulation in Production and Logistics' – bi-annual
- ASIM 'Symposium Simulation Technique' – biannual
- MATHMOD Int. Vienna Conference on Mathmatical Modelling – triennial

Furthermore, ASIM is co-sponsor of *WSC* – Winter Simulation Conference, of SCS conferences *SpringSim* and *SummerSim*, and of *I3M* and *Simutech* conference series.

ASIM Working Committee	
GMMS	Methods in Modelling and Simulation Th. Pawletta, *thorsten.pawletta@hs-wismar.de*
SUG	Simulation in Environmental Systems Jochen Wittmann, *wittmann@informatik.uni-hamburg.de*
STS	Simulation of Technical Systems Walter Commerell, *commerell@hs-ulm.de*
SPL	Simulation in Production and Logistics Sigrid Wenzel, *s.wenzel@uni-kassel.de*
EDU	Simulation in Education/Education in Simulation A. Körner, *andreas.koerner@tuwien.ac.at*
BIG DATA	Working Group Data-driven Simulation in Life Sciences; *niki.popper@dwh.at*
WORKING GROUPS	Simulation in Business Administration, in Traffic Systems, for Standardisation, etc.

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control and it is the national member of IFAC (International Federation of Automatic Control) in Spain. Since 1968 CEA-IFAC looks after the development of the Automation in Spain, in its different issues: automatic control, robotics, SIMULATION, etc. The association is divided into national thematic groups, one of which is centered on Modeling, Simulation and Optimization, constituting the CEA Spanish Modeling and Simulation Group (CEA-SMSG). It looks after the development of the Modelling and Simulation (M&S) in Spain, working basically on all the issues concerning the use of M&S techniques as essential engineering tools for decision-making and optimization.

→ *http://www.ceautomatica.es/grupos/*

→ *emilio.jimenez@unirioja.es*
simulacion@cea-ifac.es

CEA-SMSG / Emilio Jiménez, Department of Electrical Engineering, University of La Rioja, San José de Calasanz 31, 26004 Logroño (La Rioja), SPAIN

CEA - SMSG Officers	
President	Emilio Jiménez, *emilio.jimenez@unirioja.es*
Vice president	Juan Ignacio Latorre, *juanignacio.latorre@unavarra.es*
Repr. EUROSIM	Emilio Jiménez, *emilio.jimenez@unirioja.es*
Edit. Board SNE	Juan Ignacio Latorre, *juanignacio.latorre@unavarra.es*
Web EUROSIM	Mercedes Perez *mercedes.perez@unirioja.es*

Last data update February 2018

CROSSIM – Croatian Society for Simulation Modelling

CROSSIM-*Croatian Society for Simulation Modelling* was founded in 1992 as a non-profit society with the goal to promote knowledge and use of simulation methods and techniques and development of education. CROSSIM is a full member of **EUROSIM** since 1997.

→ *www.eurosim.info*

vdusak@foi.hr

CROSSIM / Vesna Dušak
Faculty of Organization and
Informatics Varaždin, University of Zagreb
Pavlinska 2, HR-42000 Varaždin, Croatia

CROSSIM Officers	
President	Vesna Dušak, *vdusak@foi.hr*
Vice president	Jadranka Božikov, *jbozikov@snz.hr*
Secretary	Vesna Bosilj-Vukšić, *vbosilj@efzg.hr*
Executive board members	Vlatko Čerić, *vceric@efzg.hr*
	Tarzan Legović, *legovic@irb.hr*
Repr. EUROSIM	Jadranka Božikov, *jbozikov@snz.hr*
Edit. Board SNE	Vesna Dušak, *vdusak@foi.hr*
Web EUROSIM	Jadranka Bozikov, *jbozikov@snz.hr*

Last data update December 2012

CSSS – Czech and Slovak Simulation Society

CSSS -The *Czech and Slovak Simulation Society* has about 150 members working in Czech and Slovak national scientific and technical societies (*Czech Society for Applied Cybernetics and Informatics*, *Slovak Society for Applied Cybernetics and Informatics*). The main objectives of the society are: development of education and training in the field of modelling and simulation, organising professional workshops and conferences, disseminating information about modelling and simulation activities in Europe. Since 1992, CSSS is full member of **EUROSIM**.

→ *www.fit.vutbr.cz/CSSS*

snorek@fel.cvut.cz

CSSS / Miroslav Šnorek, CTU Prague
FEE, Dept. Computer Science and Engineering,
Karlovo nam. 13, 121 35 Praha 2, Czech Republic

CSSS Officers	
President	Miroslav Šnorek, *snorek@fel.cvut.cz*
Vice president	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Scientific Secr.	A. Kavička, *Antonin.Kavicka@upce.cz*
Repr. EUROSIM	Miroslav Šnorek, *snorek@fel.cvut.cz*
Edit. Board SNE	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Web EUROSIM	Petr Peringer, peringer@fit.vutbr.cz

Last data update December 2012

DBSS – Dutch Benelux Simulation Society

The Dutch Benelux Simulation Society (DBSS) was founded in July 1986 in order to create an organisation of simulation professionals within the Dutch language area. DBSS has actively promoted creation of similar organisations in other language areas. DBSS is a member of **EUROSIM** and works in close cooperation with its members and with affiliated societies.

→ *www.DutchBSS.org*

a.w.heemink@its.tudelft.nl

DBSS / A. W. Heemink
Delft University of Technology, ITS - twi,
Mekelweg 4, 2628 CD Delft, The Netherlands

DBSS Officers	
President	M. Mujica Mota, *m.mujica.mota@hva.nl*
Vice president	A. Heemink, *a.w.heemink@its.tudelft.nl*
Treasurer	A. Heemink, *a.w.heemink@its.tudelft.nl*
Secretary	P. M. Scala, p.m.scala@hva.nl
Repr. EUROSIM	M. Mujica Mota, *m.mujica.mota@hva.nl*
Edit. SNE/Web	M. Mujica Mota, *m.mujica.mota@hva.nl*

Last data update February 2018

FRANCOSIM - Société Francophone de Simulation

FRANCOSIM was founded in 1991 and aims to the promotion of simulation and research, in industry and academic fields.

djouani@u-pec.fr

FRANCOSIM / Yskandar Hamam
Groupe ESIEE, Cité Descartes,
BP 99, 2 Bd. Blaise Pascal,
93162 Noisy le Grand CEDEX, France

FRANCOSIM Officers	
President	Karim Djouani, *djouani@u-pec.fr*
Treasurer	François Rocaries, *f.rocaries@esiee.fr*
Repr. EUROSIM	Karim Djouani, *djouani@u-pec.fr*
Edit. Board SNE	Karim Djouani, *djouani@u-pec.fr*

Last data update December 2012

HSS – Hungarian Simulation Society

The Hungarian Member Society of EUROSIM was established in 1981 as an association promoting the exchange of information within the community of people involved in research, development, application and education of simulation in Hungary and also contributing to the enhancement of exchanging information between the Hungarian simulation community and the simulation communities abroad. HSS deals with the organization of lectures, exhibitions, demonstrations, and conferences.

→ *www.eurosim.info*
javor@eik.bme.hu
✉ HSS / András Jávor,
Budapest Univ. of Technology and Economics,
Sztoczek u. 4, 1111 Budapest, Hungary

HSS Officers	
President	András Jávor, *javor@eik.bme.hu*
Vice president	Gábor Szűcs, *szucs@itm.bme.hu*
Secretary	Ágnes Vigh, *vigh@itm.bme.hu*
Repr. EUROSIM	András Jávor, *javor@eik.bme.hu*
Deputy	Gábor Szűcs, *szucs@itm.bme.hu*
Edit. Board SNE	András Jávor, *javor@eik.bme.hu*
Web EUROSIM	Gábor Szűcs, *szucs@itm.bme.hu*

Last data update March 2008

ISCS – Italian Society for Computer Simulation

The Italian Society for Computer Simulation (ISCS) is a scientific non-profit association of members from industry, university, education and several public and research institutions with common interest in all fields of computer simulation.

→ *www.eurosim.info*
Mario.savastano@uniina.it
✉ ISCS / Mario Savastano,
c/o CNR - IRSIP,
Via Claudio 21, 80125 Napoli, Italy

ISCS Officers	
President	M. Savastano, *mario.savastano@unina.it*
Vice president	F. Maceri, *Franco.Maceri@uniroma2.it*
Repr. EUROSIM	F. Maceri, *Franco.Maceri@uniroma2.it*
Secretary	Paola Provenzano, *paola.provenzano@uniroma2.it*
Edit. Board SNE	M. Savastano, *mario.savastano@unina.it*

Last data update December 2010

LIOPHANT Simulation

Liophant Simulation is a non-profit association born in order to be a trait-d'union among simulation developers and users; Liophant is devoted to promote and diffuse the simulation techniques and methodologies; the Association promotes exchange of students, sabbatical years, organization of International Conferences, courses and internships focused on M&S applications.

→ *www.liophant.org*
info@liophant.org
✉ LIOPHANT Simulation, c/o Agostino G. Bruzzone,
DIME, University of Genoa, Savona Campus
via Molinero 1, 17100 Savona (SV), Italy

LIOPHANT Officers	
President	A.G. Bruzzone, *agostino@itim.unige.it*
Director	E. Bocca, *enrico.bocca@liophant.org*
Secretary	A. Devoti, *devoti.a@iveco.com*
Treasurer	Marina Massei, *massei@itim.unige.it*
Repr. EUROSIM	A.G. Bruzzone, *agostino@itim.unige.it*
Deputy	F. Longo, *f.longo@unical.it*
Edit. Board SNE	F. Longo, *f.longo@unical.it*
Web EUROSIM	F. Longo, *f.longo@unical.it*

Last data update June 2016

LSS – Latvian Simulation Society

The Latvian Simulation Society (LSS) has been founded in 1990 as the first professional simulation organisation in the field of Modelling and simulation in the post-Soviet area. Its members represent the main simulation centres in Latvia, including both academic and industrial sectors.

→ *briedis.itl.rtu.lv/imb/*
merkur@itl.rtu.lv
✉ LSS / Yuri Merkuryev, Dept. of Modelling
and Simulation Riga Technical University
Kalku street 1, Riga, LV-1658, LATVIA

LSS Officers	
President	Yuri Merkuryev, *merkur@itl.rtu.lv*
Secretary	Artis Teilans, *Artis.Teilans@exigenservices.com*
Repr. EUROSIM	Yuri Merkuryev, *merkur@itl.rtu.lv*
Deputy	Artis Teilans, *Artis.Teilans@exigenservices.com*
Edit. Board SNE	Yuri Merkuryev, *merkur@itl.rtu.lv*
Web EUROSIM	Vitaly Bolshakov, *vitalijs.bolsakovs@rtu.lv*

Last data update June 2016

KA-SIM Kosovo Simulation Society

Kosova Association for Modeling and Simulation (KA-SIM, founded in 2009), is part of Kosova Association of Control, Automation and Systems Engineering (KA-CASE). KA-CASE was registered in 2006 as non Profit Organization and since 2009 is National Member of IFAC – International Federation of Automatic Control. KA-SIM joined EUROSIM as Observer Member in 2011. In 2016, KA-SIM became full member.
KA-SIM has about 50 members, and is organizing the international conference series International Conference in Business, Technology and Innovation, in November, in Durrhes, Albania, and IFAC Simulation Workshops in Pristina.

→ *www.ubt-uni.net/ka-case*
ehajrizi@ubt-uni.net
✉ MOD&SIM KA-CASE; Att. Dr. Edmond Hajrizi
Univ. for Business and Technology (UBT)
Lagjja Kalabria p.n., 10000 Prishtina, Kosovo

KA-SIM Officers	
President	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Vice president	Muzafer Shala, *info@ka-sim.com*
Secretary	Lulzim Beqiri, *info@ka-sim.com*
Treasurer	Selman Berisha, *info@ka-sim.com*
Repr. EUROSIM	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Deputy	Muzafer Shala, *info@ka-sim.com*
Edit. Board SNE	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Web EUROSIM	Betim Gashi, *info@ka-sim.com*

Last data update December 2016

PSCS – Polish Society for Computer Simulation

PSCS was founded in 1993 in Warsaw. PSCS is a scientific, non-profit association of members from universities, research institutes and industry in Poland with common interests in variety of methods of computer simulations and its applications. At present PSCS counts 257 members.

→ *www.eurosim.info*, www.ptsk.pl/
leon@ibib.waw.pl
✉ PSCS / Leon Bobrowski, c/o IBIB PAN,
ul. Trojdena 4 (p.416), 02-109 Warszawa, Poland

PSCS Officers	
President	Leon Bobrowski, *leon@ibib.waw.pl*
Vice president	Tadeusz Nowicki, *Tadeusz.Nowicki@wat.edu.pl*
Treasurer	Z. Sosnowski, *zenon@ii.pb.bialystok.pl*
Secretary	Zdzislaw Galkowski, *Zdzislaw.Galkowski@simr.pw.edu.pl*
Repr. EUROSIM	Leon Bobrowski, *leon@ibib.waw.pl*
Deputy	Tadeusz Nowicki, *tadeusz.nowicki@wat.edu.pl*
Edit. Board SNE	Zenon Sosnowski, z.sosnowski@pb.ed.pl
Web EUROSIM	Magdalena Topczewska *m.topczewska@pb.edu.pl*

Last data update December2013

SIMS – Scandinavian Simulation Society

SIMS is the *Scandinavian Simulation Society* with members from the five Nordic countries Denmark, Finland, Iceland, Norway and Sweden. The SIMS history goes back to 1959. SIMS practical matters are taken care of by the SIMS board consisting of two representatives from each Nordic country (Iceland one board member).

SIMS Structure. SIMS is organised as federation of regional societies. There are **FinSim** (Finnish Simulation Forum), **MoSis** (Society for Modelling and Simulation in Sweden), **DKSIM** (Dansk Simuleringsforening) and **NFA** (Norsk Forening for Automatisering).

→ *www.scansims.org*
erik.dahlquist@mdh.se
✉ SIMS / Erik Dahlquist, School of Business, Society and Engineering, Department of Energy, Building and Environment, Mälardalen University, P.O.Box 883, 72123 Västerås, Sweden

SIMS Officers	
President	Erik Dahlquist, *erik.dahlquist@mdh.se*
Vice president	Bernt Lie, *Bernt.Lie@usn.no*
Treasurer	Vadim Engelson, *vadim.engelson@mathcore.com*
Repr. EUROSIM	Erik Dahlquist, *erik.dahlquist@mdh.se*
Edit. Board SNE	Esko Juuso, *esko.juuso@oulu.fi*
Web EUROSIM	Vadim Engelson, *vadim.engelson@mathcore.com*

Last data update February 2018

SLOSIM – Slovenian Society for Simulation and Modelling

SLOSIM - Slovenian Society for Simulation and Modelling was established in 1994 and became the full member of **EUROSIM** in 1996. Currently it has 90 members from both Slovenian universities, institutes, and industry. It promotes modelling and simulation approaches to problem solving in industrial as well as in academic environments by establishing communication and cooperation among corresponding teams.

→ www.slosim.si

slosim@fe.uni-lj.si

SLOSIM / Vito Logar, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

SLOSIM Officers	
President	Vito Logar, *vito.logar@fe.uni-lj.si*
Vice president	Božidar Šarler, *bozidar.sarler@ung.si*
Secretary	Aleš Belič, *ales.belic@sandoz.com*
Treasurer	Milan Simčič, *milan.simcic@fe.uni-lj.si*
Repr. EUROSIM	B. Zupančič, *borut.zupancic@fe.uni-lj.si*
Deputy	Vito Logar, *vito.logar@fe.uni-lj.si*
Edit. Board SNE	R. Karba, *rihard.karba@fe.uni-lj.si*
Web EUROSIM	Vito Logar, *vito.logar@fe.uni-lj.si*

Last data update February 2018

UKSIM - United Kingdom Simulation Society

The UK Simulation Society is very active in organizing conferences, meetings and workshops. UKSim holds its annual conference in the March-April period. In recent years the conference has always been held at Emmanuel College, Cambridge. The Asia Modelling and Simulation Section (AMSS) of UKSim holds 4-5 conferences per year including the EMS (European Modelling Symposium), an event mainly aimed at young researchers, organized each year by UKSim in different European cities.

Membership of the UK Simulation Society is free to participants of any of our conferences and their co-authors.

→*uksim.info*

david.al-dabass@ntu.ac.uk

UKSIM / Prof. David Al-Dabass
Computing & Informatics,
Nottingham Trent University
Clifton lane, Nottingham, NG11 8NS
United Kingdom

UKSIM Officers	
President	David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Secretary	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Treasurer	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Membership chair	G. Jenkins, *glenn.l.jenkins@smu.ac.uk*
Local/Venue chair	Richard Cant, *richard.cant@ntu.ac.uk*
Repr. EUROSIM	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Deputy	G. Jenkins, *glenn.l.jenkins@smu.ac.uk*
Edit. Board SNE	A. Orsoni, *A.Orsoni@kingston.ac.uk*

Last data update March 2016

RNSS – Russian Simulation Society

NSS - The Russian National Simulation Society (Национальное Общество Имитационного Моделирования – НОИМ) was officially registered in Russian Federation on February 11, 2011. In February 2012 NSS has been accepted as an observer member of **EUROSIM**, and in 2015 RNSS has become full member.

→ *www.simulation.su*

yusupov@iias.spb.su

RNSS / R. M. Yusupov,
St. Petersburg Institute of Informatics and Automation RAS, 199178, St. Petersburg, 14th lin. V.O, 39

RNSS Officers	
President	R. M. Yusupov, *yusupov@iias.spb.su*
Chair Man. Board	A. Plotnikov, *plotnikov@sstc.spb.ru*
Secretary	M. Dolmatov, *dolmatov@simulation.su*
Repr. EUROSIM	R.M. Yusupov, *yusupov@iias.spb.su* Y. Senichenkov, *senyb@dcn.icc.spbstu.ru*
Deputy	B. Sokolov, *sokol@iias.spb.su*
Edit. Board SNE	Y. Senichenkov, *senyb@dcn.icc.spbstu.ru*, *senyb@mail.ru*

Last data update February 2018

EUROSIM OBSERVER MEMBERS

ROMSIM – Romanian Modelling and Simulation Society

ROMSIM has been founded in 1990 as a non-profit society, devoted to theoretical and applied aspects of modelling and simulation of systems. ROMSIM currently has about 100 members from Romania and Moldavia.

→ *www.eurosim.info* (*www.ici.ro/romsim*)

sflorin@ici.ro

ROMSIM / Florin Hartescu,
National Institute for Research in Informatics, Averescu Av. 8 – 10, 71316 Bucharest, Romania

ROMSIM Officers	
President	
Vice president	Florin Hartescu, *flory@ici.ro* Marius Radulescu, *mradulescu.csmro@yahoo.com*
Repr. EUROSIM	Marius Radulescu, *mradulescu.csmro@yahoo.com*
Deputy	Florin Hartescu, *flory@ici.ro*
Edit. Board SNE	Constanta Zoe Radulescu, *zoe@ici.ro*
Web EUROSIM	Florin Hartescu, *flory@ici.ro*

Last data update June 2017

MIMOS – Italian Modelling and Simulation Association

MIMOS (Movimento Italiano Modellazione e Simulazione – Italian Modelling and Simulation Association) is the Italian association grouping companies, professionals, universities, and research institutions working in the field of modelling, simulation, virtual reality and 3D, with the aim of enhancing the culture of 'virtuality' in Italy, in every application area.
MIMOS became **EUROSIM** Observer Member in 2016 and is preparing application for full membership.

→ *www.mimos.it*

roma@mimos.it – info@mimos.it

MIMOS – Movimento Italiano Modellazione e Simulazione; via Ugo Foscolo 4, 10126 Torino – via Laurentina 760, 00143 Roma

MIMOS Officers	
President	Paolo Proietti, *roma@mimos.it*
Secretary	Davide Borra, *segreteria@mimos.it*
Treasurer	Davide Borra, *segreteria@mimos.it*
Repr. EUROSIM	Paolo Proietti, *roma@mimos.it*
Deputy	Agostino Bruzzone, *agostino@itim.unige.it*
Edit. Board SNE	Paolo Proietti, *roma@mimos.it*

Last data update December 2016

CANDIDATES

Albanian Simulation Society

At the Department of Statistics and Applied Informatics, Faculty of Economy, University of Tirana, Prof. Dr. Kozeta Sevrani at present is setting up an Albanian Simulation Society. Kozeta Sevrani, professor of Computer Science and Management Information Systems, and head of the Department of Mathematics, Statistics and Applied Informatic, has attended a EUROSIM board meeting in Vienna and has presented simulation activities in Albania and the new simulation society.
The society – constitution and bylaws are being worked out – will be involved in different international and local simulation projects, and will be engaged in the organisation of the conference series ISTI – Information Systems and Technology. The society intends to become a EUROSIM Observer Member.

kozeta.sevrani@unitir.edu.al

Albanian Simulation Goup, attn. Kozeta Sevrani
University of Tirana, Faculty of Economy
rr. Elbasanit, Tirana 355 Albania

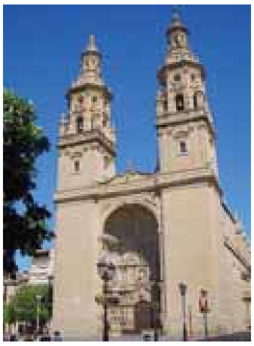
Albanian Simulation Society- Officers (Planned)	
President	Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Secretary	
Treasurer	
Repr. EUROSIM	Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Edit. Board SNE	Albana Gorishti, *albana.gorishti@unitir.edu.al* Majlinda Godolja, *majlinda.godolja@fshn.edu.al*

Last data update December 2016

EUROSIM 2019

10th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 1 – 5, 2019



UNIVERSIDAD DE LA RIOJA

EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM 2019, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics The EUROSIM 2019 Congress will include invited talks, parallel, special and poster sessions, exhibition and versatile technical and social tours. The Congress topics of interest include, but are not limited to:

- Intelligent Systems and Applications
- Hybrid and Soft Computing
- Data & Semantic Mining
- Neural Networks, Fuzzy Systems & Evolutionary Computation
- Image, Speech & Signal Processing
- Systems Intelligence and Intelligence Systems
- Autonomous Systems
- Energy and Power Systems
- Mining and Metal Industry
- Forest Industry
- Buildings and Construction
- Communication Systems
- Circuits, Sensors and Devices
- Security Modelling and Simulation
- Bioinformatics, Medicine, Pharmacy and Bioengineering
- Water and Wastewater Treatment, Sludge Management and Biogas Production
- Condition monitoring, Mechatronics and maintenance
- Automotive applications
- e-Science and e-Systems
- Industry, Business, Management, Human Factors and Social Issues
- Virtual Reality, Visualization, Computer Art and Games
- Internet Modelling, Semantic Web and Ontologies
- Computational Finance & Economics
- Simulation Methodologies and Tools
- Parallel and Distributed Architectures and Systems
- Operations Research
- Discrete Event Systems
- Manufacturing and Workflows
- Adaptive Dynamic Programming and Reinforcement Learning
- Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
- Control of Intelligent Systems
- Robotics, Cybernetics, Control Engineering, & Manufacturing
- Transport, Logistics, Harbour, Shipping and Marine Simulation

Congress Venue / Social Events The Congress will be held in the City of Logroño, Capital of La Rioja, Northern Spain. The main venue and the exhibition site is the University of La Rioja (UR), located on a modern campus in Logroño, capital of La Rioja, where 7500 students are registered. The UR is the only University in this small, quiet region in Northern Spain. La Rioja is where the Monasteries of San Millán de la Cogolla, cradle of the first words written in the Spanish language, are situated, sites included in UNESCO's World Heritage List in 1996. Of course, social events will reflect this heritage – and the famous wines in la Rioja.

Congress Team: The Congress is organised by CAE CAE-SMSG, the Spanish simulation society, and Universidad de la Rioja.

Info: Emilio Jiménez, EUROSIM President, emilio.jimenez@unirioja.es
Juan Ignacio Latorre, juanignacio.latorre@unavarra.es

www.eurosim.info

www.tuverlag.at
ISBN 978-3-903024-70-0