Comparisons of Different Modelica-Based Simulators Using Benchmark Tasks

Olaf Enge-Rosenblatt1), **Christoph Clauß**1), **Peter Schwarz**1),
Felix Breitenecker2), **Christoph Nytsch-Geusen**3)
1) Fraunhofer Institute for Integrated Circuits, Branch Lab Design Automation,
Zeunerstraße 38, 01069 Dresden, Germany
olaf.enge@eas.iis.fraunhofer.de
2) Vienna University of Technology, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria
3) Fraunhofer Institute for Computer Architecture and Software Technology,
Kekuléstraße 7, 12489 Berlin, Germany

Abstract

A benchmark library is presented which collects models for testing and comparing different analog and hybrid simulators as well as their numerical simulation algorithms. Many of these models are described with Modelica and simulated with Dymola and the Modelica-related simulator Mosilab. But VHDL-AMS descriptions are also used to compare simulation results of Modelica simulators with those of other types of simulators. The motivation of the selection of benchmark problems, the modeling and documentation "style guide", and some small examples from electronics and mechanics are described.

1 Motivation

The development of new simulators and model libraries has to be accompanied by intensive simulations of test examples and their comparison. The first reason for collecting a new benchmark library was the development of a Modelica-based simulator Mosilab [1] and accompanying test examples to ensure the Mosilab functionality. But, there are some other objectives:

- comparison of Mosilab with commercial Modelica simulators: Dymola, SimulationX;
- potential extension to comparisons with other analog simulators (e.g. VHDL-AMS, Verilog-AMS, SystemC-AMS);
- getting experiences with the numerical properties of the implemented solvers and their robustness (e.g., influenced by simulator control parameters);
- testing extreme cases (e.g., depending on the number of variables and equations as well as numerical parameter values);
- collecting models with a special focus on systems with variable structure;
- preparation of regression tests;
- and, last but not least, pedagogical aspects: for use in lectures and tutorials.

Therefore, the construction or selection of benchmark models has to fulfill many criteria. The ARGESIM comparisons ([2], [3]), published in the journal Simulation News Europe (SNE) and via http://www.argesim.org/ , have a similar goal. They are considered here from a common point of view.

Further suggestions are expected from benchmarks in other disciplines ([7], [8]) or with a general methodological background ([9]).

2 Types of simulation problems

The benchmark models are selected with respect to the following tasks:

- simple tests of keywords and other language constructs (especially for compiler tests and version checking in the new Mosilab simulator),
- simple but non-trivial electric circuits (from RLC circuits up to transformers and rectifiers),
- testing typical numerical simulation problems (e.g. stiff differential equations, discontinuities, simulation of ideal oscillators)
- more complicated transistor models which lead in many cases to numerical simulation problems in simulators which are not specialized for electronic applications,
- test of advantageous description means (e.g. object oriented approaches)

- erroneous models (e.g. parallel ideal voltage sources) to check the simulator's behavior in error cases
- inclusion of some "classical", mostly non-electrical ARGESIM comparisons in new or updated form (until now: C1, C3, C5, C7; in preparation: C11, C12),
- testing the capability of simulating systems with variable structures (also called "structural variability" or "model structure dynamics", see [3], [4], [5], [6]): rectifiers with ideal diodes, voltage duplexers with two ideal diodes, constrained pendulum C7, string pendulum,
- modeling with embedded statecharts (as a potential extension of the Modelica language), especially for the Mosilab capabilities of handling variable structures.

1	**Introduction**	**5**
2	**Small examples, taken from Modelica**	**6**
2.1	**Damped RLC oscillator (rlc_osci)**	**6**
2.2	**Transformer (trafo)**	**8**
2.3	**Ideal diode (ideal_diode)**	**12**
2.4	**MOS inverter (mos_inv)**	**15**
2.5	**MOS oscillator (mos_osci)**	**18**
2.6	**Switched capacitor integrator (sc_integrator)**	**23**
2.7	**Test of integrator's error (integr_error)**	**27**
2.8	**Voltage duplexer (Vduplexer)**	**30**
2.9	**Graetz rectifier with resistive load (ideal_graetz)**	**34**
2.10	**Graetz rectifier with RC load (ideal_graetz2)**	**38**
3	**ARGESIM examples**	**43**
3.1	**Comparison 1 (comparison_1)**	**43**
3.2	**Comparison 3 (comparison_3)**	**48**
3.3	**Comparison 5 (comparison_5)**	**55**
3.4	**Comparison 7 (comparison_7)**	**61**
4	**Examples with structural variability**	**63**
4.1	**Ideal diode using events (ideal_diode_event):**	**63**
4.2	**Ideal diode using statecharts**	**64**
4.2.1	**1st realisation (ideal_diodeSC)**	**64**
4.2.2	**2nd realisation (ideal_diodeSCcon)**	**67**
4.2.3	**3rd realisation (ideal_diodeSCdyn)**	**69**
4.2.4	**4th realisation (ideal_diodeSCvN)**	**71**
4.3	**Voltage duplexer using statecharts**	**75**
4.3.1	**1st realisation (VduplexerSC)**	**75**
4.3.2	**2nd realisation (VduplexerSCcon)**	**78**
4.3.3	**3rd realisation (VduplexerSCdyn)**	**81**
4.4	**Comparison 7 using statecharts**	**84**
4.4.1	**1st realisation (comparison_7SC)**	**84**
4.4.2	**2nd realisation (comparison_7SCdyn)**	**86**
4.5	**String pendulum with free motion (fadenpendel)**	**88**
4.6	**String pendulum with free motion using statecharts**	**96**
4.6.1	**Realisation fadenpendelSC**	**96**
4.6.2	**Realisation fadenpendelSCaF**	**102**
4.6.3	**Realisation fadenpendelSC1**	**103**
4.6.4	**Realisation fadenpendelSC3a**	**103**
4.6.5	**Realisation fadenpendelSCdyn**	**106**
4.7	**Switched capacitor integrator in QV formulation**	**112**
4.7.1	**Realisation without statecharts (sc_integrator_QV)**	**112**
4.7.2	**Realisation with statecharts (sc_integrator_QVSC)**	**116**

Table 1: Benchmark library

ARGESIM continues the comparisons by benchmarks with extended information and prepares special benchmarks with emphasis on various modeling approaches. In 2008, benchmarks for *hybrid* modeling and simulation will be published, addressing different modeling techniques for four or five systems (constrained pendulum, rotating pendulum, heat diffusion with different regimes, rotor dynamics).

The content of the actually implemented benchmark library is summarized in Table 1. It consists of three main sections. In the first section, some electrical examples are collected. The second section deals with a selection of the ARGESIM benchmarks, which are mainly published in the journal Simulation News Europe (SNE). The third section collects examples which are characterized by a variation of the model structure. Such systems lead to different sets of differential-algebraic equations and the need of exchanges between them from time to time during the simulation process ([1], [4], [5], [10]).

3 Documentation

Each test example is documented in the same manner:

- short description of the problem and the reason for selecting this model,
- graphical description (schematic/sketch),
- definition of relevant physical quantities and dimensions,
- interface description (e.g., type of signals and quantities),
- textual input description in the Modelica language,
- applied simulator control parameters,
- graphical simulation results and some additional textual information,
- discussion of results (e.g., accuracy, runtime behavior) and detected problems.

If the models should be used for regression tests, further regimentations are necessary.

4 Examples

In this section, some interesting benchmark tasks are collected and discussed shortly. All examples are characterized by variable structure because serious numerical problems consist yet in very small systems.

4.1 Electric example

The electric example shall illustrate the application of different models of a diode component. For this purpose, the diode is used within two different well-known set-ups: a one-way rectifier with an ohmic load (shown in Fig. 1) and a Graetz rectifier with an ohmic-capacitive load (depicted in Fig. 2).

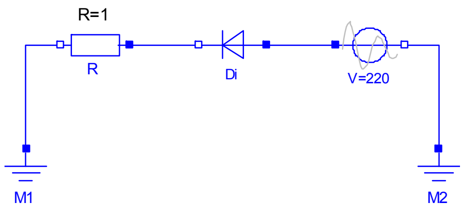

Figure 1: One-way rectifier with ohmic load

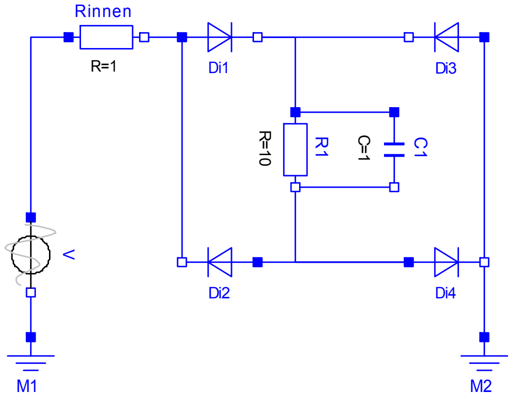

Figure 2: Graetz rectifier with ohmic-capacitive load

First, the piecewise-linear (PWL) diode model of the Modelica Standard Library is used. The relevant source code is shown in Table 2. This model implements the behavior of an idealized switching diode consisting of a piecewise-linear voltage-current characteristic. A so-called auxiliary variable is used which implements a parametric representation of the length of both straight lines [11], [12], [14].

```
model IdealDiode
  extends OnePort;
  parameter Real Ron= 1.E-5,
  parameter Real Goff= 1.E-5;
  Boolean off(start=true);
  Real s;
equation
  off = s < 0;
  v = s*(if off then 1 else Ron);
  i = s*(if off then Goff else 1);
end IdealDiode;
```

Table 2: Source code of diode using auxiliary variable

Second, an ideal diode model was implemented:

- The voltage in flow direction is zero (conducting state).
- The current in the blocking direction is zero (cut-off state).

Conditional equations are used for voltage and current always forcing at least one of them to zero. The source code is shown in Table 3. This implementation requires an event handling by the simulator.

```
model IdealDiodeEvent
  extends OnePort;
  Boolean blocking(start=true);
equation
  blocking = if pre(blocking)
               then v<0 else i<0;
  if blocking then
    i = 0;
  else
    v = 0;
  end if;
end IdealDiodeEvent;
```

Table 3: Source code of diode using conditional equation

With all simulators under test, very similar simulation results were received for the one way rectifier. Exemplarily, Fig. 3 shows simulation results for some voltages calculated by Mosilab using the PWL diode model. The results of the other simulators are the same. This statement also holds for the ideal diode model no matter which simulator is tested. Of course, the current of the blocking diode is now exactly equal to zero or, vice versa, the voltage of the conducting diode now vanishes completely.

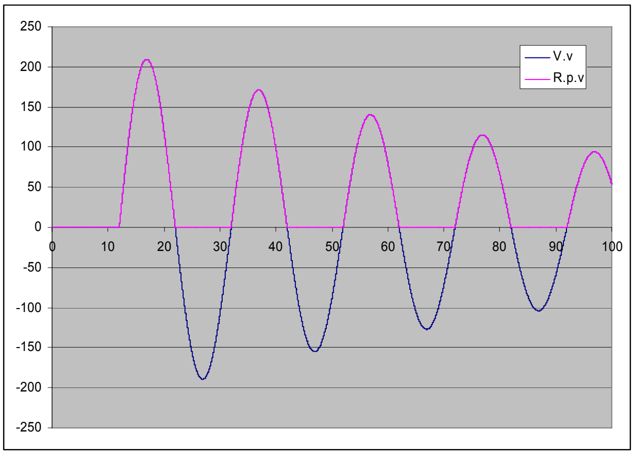

Figure 3: Simulation result from Mosilab for a one-way rectifier

In contrast, the electric circuit of the Graetz rectifier can only be simulated using the PWL diode model (Table 2). The property of such a circuit, that two diodes of the four must unconditionally be closed (or opened) at the same time, is the reason for this fact. This conclusion is valid for Dymola as well as for Mosilab. To handle a circuit with a Graetz rectifier using ideal diodes, it is necessary to qualify a simulator with the feature of finding a valid new model structure from the complete set of structures at each switching point in time.

Some simulation results for the Graetz rectifier using the PWL diode model are shown in Fig. 4 and Fig. 5. Fig. 4 depicts some voltages while the corresponding currents are shown in Fig. 5.

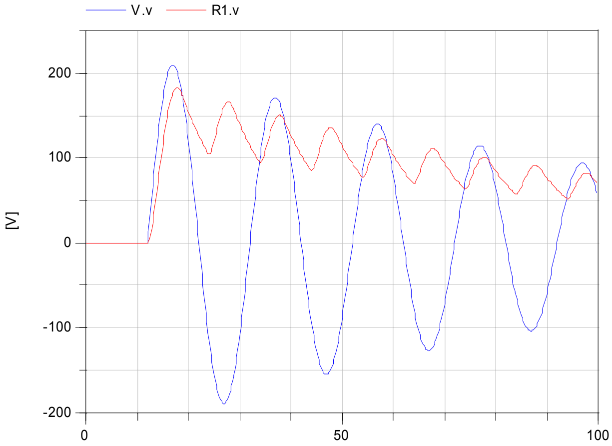

Figure 4: Voltages of the Graetz rectifier circuit

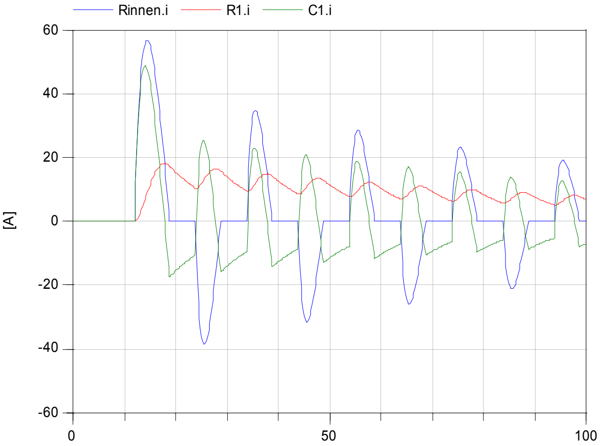

Figure 5: Currents of the Graetz rectifier circuit

4.2 Two-state model

The two-state model considered here is the ARGESIM comparison C5 which is of high interest regarding to the numerical behavior of each simulator. The problem consists of the two simple differential equations:

$$\begin{aligned}\dot{y}_1 &= c_1\left(y_2 + c_2 - y_1\right),\\ \dot{y}_2 &= c_3\left(c_4 - y_2\right).\end{aligned} \tag{1}$$

In Equ. (1), the parameters c1 and c3 are fixed while c2 and c4 have different values depending on the actual state of the system. State 1 is valid as long as $y_1 < 5.8$. Reaching this value, the system state is changed to state 2 which, then, is valid until y_1 goes below 2.5. All parameters and initial conditions were chosen in a very sophisticated manner. This way, the numerical accuracy of the simulators under test can be investigated by looking at the switching points in time, especially at the last one (denoted with `t5`) which appears generally at about 5 Seconds.

The Dymola result computed by the DASSL solver using the highest possible numerical accuracy (tolerance is set to `1E-12`) shall be taken as reference for other simulators. The last switching point in time appears at `t5=4.999999646`. Other solvers, like Runge-Kutta methods, are less suitable for such kind of a simulation task.

With Mosilab, the switching point in time is found very well if using the IDA solver which is very similar to the DASSL method. With an absolute tolerance of `1E-14` and a relative tolerance of `1E-10`, the switching point in time can be determined to `t5=4.999999645`. Surely, this is a very good result. But using lower tolerances or using one of the other numerical solvers of the Mosilab simulator leads to more inexact results.

Exemplarily, Fig. 6 shows the time behavior of y_1 using Dymola with the DASSL method as mentioned above.

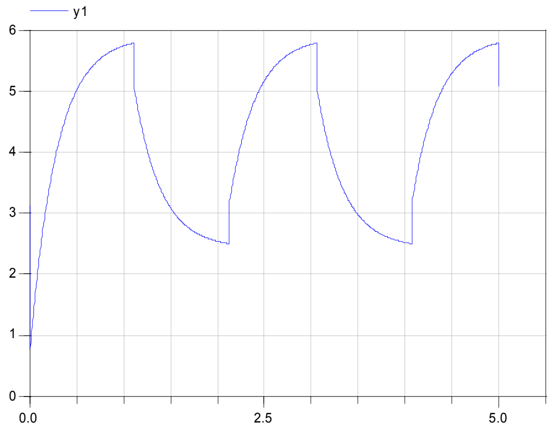

Figure 6: Time behavior of state variable y_1

4.3 String pendulum

A string pendulum is shown in Fig. 7. A point mass is able to perform circular or free (downfall) movements – so-called phases (see Fig. 7A). The circular movement is characterized by a stretched (but non-widening) thread, i.e. the mass has the maximal possible distance to the fixing point. In contrast, the mass has a smaller distance and the thread is folded during the free movement. This is an extension of the well-known mathematical pendulum with small elongations and without the downfall phase.

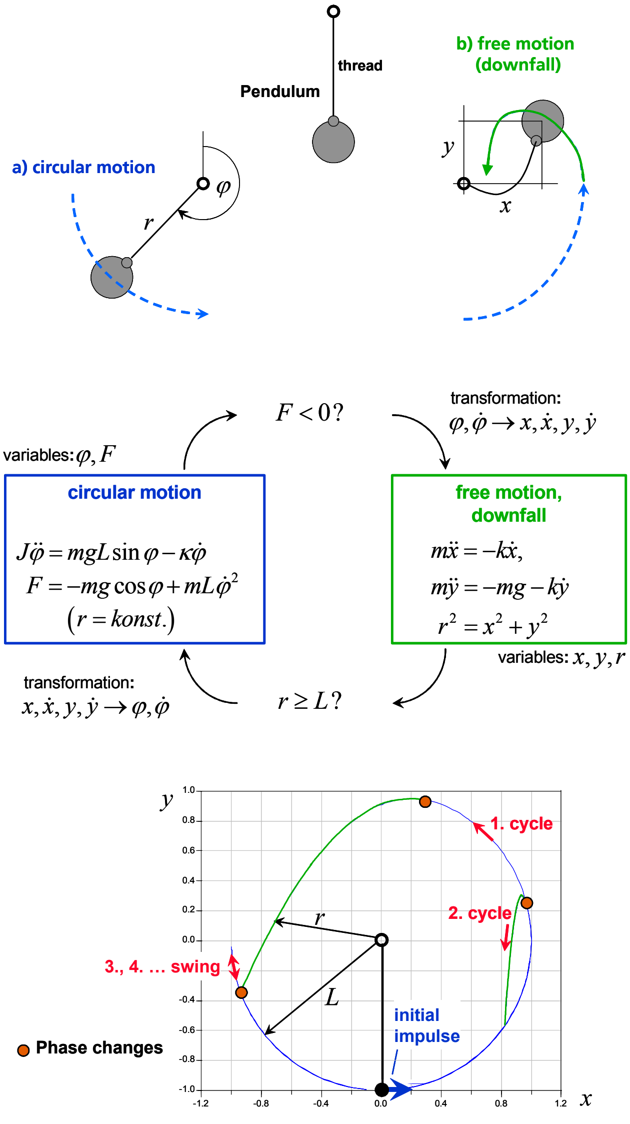

Fig. 7: String pendulum

A) Geometrical configuration

B) Mathematical problem formulation of both phases

C) Simulation results

The simplest description of the circular motion uses polar coordinates; the downfall motion may be described with Cartesian coordinates. There are two differential-algebraic equation systems with two and three variables, respectively, describing both phases (see Fig. 7B). In phase 1 (circular movement), the stretching force F in the thread is greater zero. In phase 2 (free movement), the distance r between point mass and fixing point is less than the length L of the thread. The "indicator functions" ($F < 0$ and $r \geq L$) are used to detect the points in time of a necessary switching between the phases.

A large initial impulse results in a sequence of circular and free movements. This is illustrated in Fig. 7C. The point mass performs two "circles" followed by some swinging movements. The time behavior of the mass position and the corresponding force F are shown in Fig. 8 and Fig. 9, respectively.

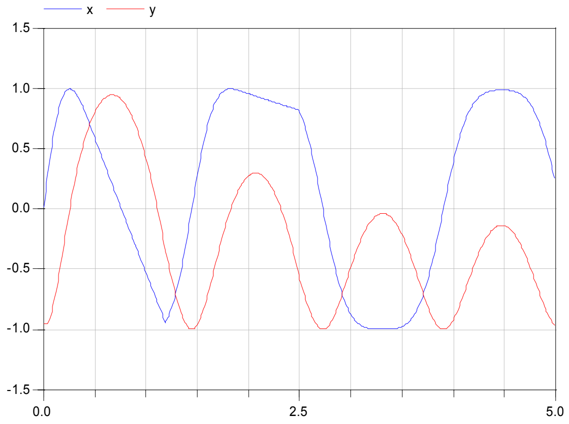

Figure 8: Pendulum's mass position (x and y)

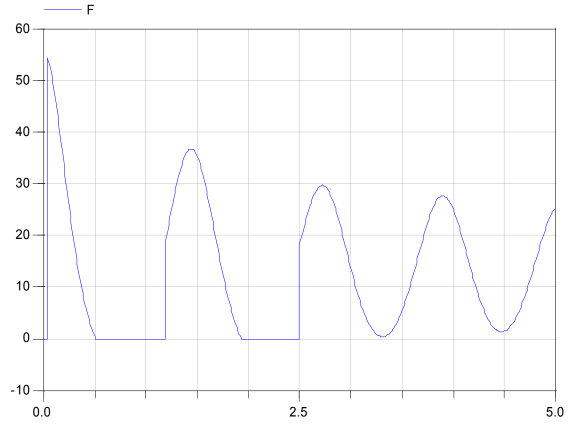

Figure 9: Force F in the thread

This description is closely related to a statechart description, which can be formulated with the State Graph Library and simulated using Dymola or with an extended Modelica description for the Mosilab simulator.

The model implementation depends strongly on the applied simulator. In Dymola, it is necessary to use the same number of equations in both phases. Therefore, some dummy equations have to be introduced. In simulators like AnyLogic or Mosilab, different numbers of equations are allowed in various model states.

5 Summary and outlook

This collection of benchmark problems is under development in connection with the Mosilab development [1] and has its roots in a Fraunhofer-internal research project GENSIM. Some parts of these examples will be published in connection with new Modelica-oriented projects. Problems of more general interest will be prepared for the widely-distributed ARGESIM comparisons.

The collection of benchmarks presented here has proved as a powerful tool for testing the numerical behavior and the modeling limits of different simulators. In this paper, only some examples of general interest are described.

The collection is under continuous development. The pool of tasks as well as the tested simulators and the different modeling languages have to be extended.

It is intended to include parts of the benchmark examples into the regression test library ModelicaTest, which is used by the Modelica Design Group for developing the Modelica Standard Library.

References

[1] Nytsch-Geusen C et al. Mosilab: Development of a Modelica based generic simulation tool supporting model structural dynamics. 4th Int. Modelica Conf., Hamburg, Germany, March 2005.

[2] Breitenecker F et al. Education in modeling and simulation using ARGESIM comparisons/benchmarks with physical modeling. MATHMOD (5th IMACS Symposium on Mathematical Modelling), Vienna, Austria, February 2006.

[3] Pawlik S et al. A classification of modeling and simulation approaches based on the ARGESIM benchmarks. EUROSIM, Ljubljana, Slovenia, September 2007.

[4] Schwarz P. Simulation of systems with dynamically varying model structure. MATHMOD (5th IMACS Symposium on Mathematical Modelling), Vienna, Austria, February 2006.

[5] Breitenecker F et al. Structure of simulators for hybrid systems – general development and introduction of a concept of external and internal state events. EUROSIM, Ljubljana, Slovenia, September 2007.

[6] Nytsch-Geusen C. The use of the UML within the modelling process of Modelica-models. Proc. 1st Int. Workshop EOOLT, Berlin, Germany, August 2007.

[7] Benchmarks for water supply modelling, simulation and optimisation. http://www.eng.dmu.ac.uk/~wss/

[8] Iwnicki S (Ed.). The Manchester Benchmarks for Rail Vehicle Simulation (Supplement Vehicle System Dynamics (SVD)). Swets & Zeitlinger, July 1999.

[9] Gelfert A. Simulating many-body models in physics: rigorous results, ‘benchmarks’, and cross-model justification. LSE/CPNSS Conf. on Models and Simulations, Paris, France, June 2006.

[10] Enge-Rosenblatt O et al. Numerical Simulation of Continuous Systems with Structural Dynamics. EUROSIM, Ljubljana, Slovenia, September 2007.

[11] Mattsson SE, Otter M, Elmqvist H. Modelica Hybrid Modeling and Efficient Simulation. 38th IEEE Conference on Decision and Control, CDC’99, Phoenix, Arizona, USA, December 1999.

[12] Otter M, Elmqvist H, Mattsson SE. Objektorientierte Modellierung physikalischer Systeme, Teil 8. at – Automatisierungstechnik 47(1999)9, A29-A32.

[13] Fritzson P. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Wiley, New York, 2004.

[14] Clauß C, Haase J, Kurth G, Schwarz P. Extended admittance description of nonlinear n-poles. Archiv für Elektronik und Übertragungstechnik / Int. J. Electronics and Communications, Vol. 40, pp. 91-97, 1995.