Improvement of MSL Electrical Analog Library

Kristin Majetta   Christoph Clauß   Matthias Franke   Peter Schneider

Fraunhofer-Institute for Integrated Circuits, Design Automation Division

Zeunerstraße 38, 01069 Dresden, Germany

{Kristin.Majetta, Christoph.Clauss, Matthias.Franke, Peter.Schneider}@eas.iis.fraunhofer.de

Abstract

The set of device models of the Modelica electrical analog library needs to be extended to support improved applications. Within the EUROSYSLIB project a multiple line model, a zener diode, a thyristor, an opamp and some switches with arc models, and a conditional heat port were added. The performance was improved generally. Furthermore, IGBT, TRIAC and converter models are planned.

Keywords: Modelica standard library, electrical analog library, EUROSYSLIB

1   Introduction

The electrical analog library of the Modelica Standard Library (MSL) has been developed for more than 10 years [1], [2]. It was widely used which caused suggestions for improving the library. The intention of the library development is to

- meet the user requirements for functionality
- give examples for device modeling
- meet a textbook like level of simplicity

Therefore, the library development is a long-lasting user-driven process. In this paper some single improvements are collected which arose on partner demand in the EUROSYSLIB project.

2   Improved performance due to smooth-operator

The smooth-operator is an event-related built-in intrinsic operator of Modelica [3]. It can be used to state the smootheness property of a Real expression. If an expression is expected to be discontinuous (e.g. due to an if-statement) usually an event has to be generated which interrupts the analog simulation algorithm. This interruption can be avoided if the smoothness property was stated using the smooth operator. Furthermore, the level of smoothness (how often the expression is continuously differentiable)

can be used by the simulation tool to choose the appropriate simulation order.

The MSL Electrical Analog Library was checked with regard to smoothness. Mostly at semiconductor devices the smooth-operator could be used. This improved the performance in large circuit models considerably. As an example an adder chain was simulated using Dymola. It consists of 10 full adders built by MOSFET transistors which are modeled using the MSL Electrical.Analog.Semiconductor models. All in all 620 transistors are used. The improvement is shown in the following table 1.

Advanced. SmoothEvent	Smooth operator	F-Evaluations	steps	CPU Seconds
default (1)	not used	1994791	8575	1230
default (1)	used	1487274	6767	962
0	used	28284	1541	4.8

Tab. 1: Full adder chain statistics

The second line addresses the usage of the smooth-operator whereas all simulator parameters of Dymola are default.

To fully exploit the smoothness information given by the smooth-operator in Dymola the variable

```
Advanced.SmoothEvent = 0
```

is recommended to be set. This causes a drastic improvement shown in the third line of table 1. The SmoothEvent variable controls the event handling if the smooth operator is applied. By setting SmoothEvent to zero no events are generated which causes a high performance, but it must be taken into consideration that precision can be lost. Consequently, the smooth operator should be applied if further models will be added to the MSL. Otherwise, the smooth-operator should be improved to avoid the necessity of setting SmoothEvent.

3   Conditional heat port

In usual electrical circuit simulation a heat port is not necessary. Each device is simulated using its nomi-

nal temperature which sometimes can be set via parameter. If otherwise a coupled electrical and thermal simulation is asked for a thermal heat port for the connection to a thermal network is desired. Therefore, a conditional heat port was introduced that can be switched on if needed. Moreover, the heat port is useful for a clear handling of energy conservation. The loss power which normally disappears in a pure electrical simulation is dissipated to the heat port, and the energy balance becomes correct.

As an example the new resistor model with conditional heat port is described. The Boolean parameter useHeatPort controls the presence of the heat port. Fig. 1 shows the resistor icon which is sensitive regarding to the useHeatPort parameter.

Fig. 1: Resistor icon with and without heat port

If the useHeatPort parameter is set to true, a heat port of the thermal heat transfer library exists the temperature of which is equal to an additional variable T_heatPort. The heat flow rate of the heat port is set to the variable LossPower, the value of which has to be calculated basing on electrical loss power. In this way the electrical loss power is transformed into an accompanying thermal network. Otherwise via T_heatPort the actual temperature can be used in the electrical component.

If the heat port is not active (useHeatPort false, default case) the actual temperature is constant by setting to a parameter T. Though the LossPower variable is calculated it is not used.

Since this functionality of the heat port is the same in all electrical components it was extracted to a partial model ConditionalHeatPort which can be extended from electrical components if a conditional heat port is necessary. The source code of the ConditionalHeatPort model is:

```
partial model ConditionalHeatPort
  import SI=Modelica.SIunits;
  import H=Modelica.Thermal.HeatTransfer;
  parameter Boolean useHeatPort = false
    "=true, if HeatPort is enabled";
  parameter SI.Temperature T=293.15
    "Fixed device temperature
       if HeatPort is disabled";
  H.Interfaces.HeatPort_a
    heatPort(T(start=T)=T_heatPort,
    Q_flow=-LossPower) if useHeatPort;
```

```
  SI.Power LossPower "Loss power leaving
    component via HeatPort";
  SI.Temperature T_heatPort
    "Temperature of HeatPort";
equation
  if not useHeatPort then
    T_heatPort = T;
  end if;
end ConditionalHeatPort;
```

Via inheritance the conditional heat port is used in the resistor component. To make the component temperature sensitive the parameters T_ref (refenrence temperature at which e.g. the resistance R was measured) and alpha (linear temperature coefficient) were introduced. The temperature T_heatPort which is either coming from the heat port or is set to T_ref if the heat port is not enabled is used to calculate the temperature dependent resistance R_actual:

```
  R_actual =R*(1+alpha*(T_heatPort-T_ref));
```

Using alpha = 0 (default) R_actual is identical to the resistance parameter R, and the resistor is not temperature dependent.

The electrical dissipation is simply calculated by

```
  LossPower = v*i;
```

The partial model ConditionalHeatPort connects the variable LossPower to the heat flow rate Q_flow of the heat port.

The source code of the improved resistor model is:

```
model Resistor
  import SI=Modelica.SIunits;
  import A=Modelica.Electrical.Analog;
  parameter SI.Resistance R(start=1)
    "Resistance at temperature T_ref";
  parameter SI.Temperature T_ref=300.15
    "Reference temperature";
  parameter SI.LinearTemperatureCoefficient
    alpha=0;
  extends A.Interfaces.OnePort;
  extends A.Interfaces.ConditionalHeatPort
    (T = T_ref);
  SI.Resistance R_actual;
equation
  R_actual =R*(1+alpha*(T_heatPort-T_ref));
  R_actual*i = v;
  LossPower = v*i;
end Resistor;
```

In this way a conditional heat port was added to many devices with electrical dissipation. The constant and the variable resistor and conductor as well as some semiconductor devices are temperature sensitive. Most of the ideal devices are not temperature sensitive, but the loss power is calculated. In many cases the temperature sensitive devices are still separated from not sensitive devices.

As an example a simple NOR circuit is regarded (fig. 2) with temperature sensitive bipolar transistors. Fig. 3 shows both the electrical and the thermal behavior. Since the thermal network is nonrealistic because of missing cooling devices, the temperature rises until the electrical function crashes (fig. 4).

Fig. 4: Crash due to overheating

4   Multiple line model

For an improved modeling of electrical lines a multiple line model was added. It consists of a series of lumped RLGC segments according to fig. 5. Both the number of lines and the number of segments can be specified.

Further parameters are the length of line, and the matrices of the linear electric constants (r, l, g, c) which are used to calculate the segment parameters. E.g. the segment parameter R1 is calculates by

$$R1 = r1 * \text{length} / \text{number of segments}$$

The segment parameters do not vary along the line. Only the R and L parameters of the first segment are cut into halves. They are complemented to full values by a special segment at the other end of the line which consists of R and L devices only. This special arrangement is used for reasons of symmetry. Frequency dependent loss parameters [4] are not modeled.

Since the linear electric constants matrices are symmetrical only the upper triangular matrix has to be specified.

Fig. 2: Thermal electrical NOR circuit
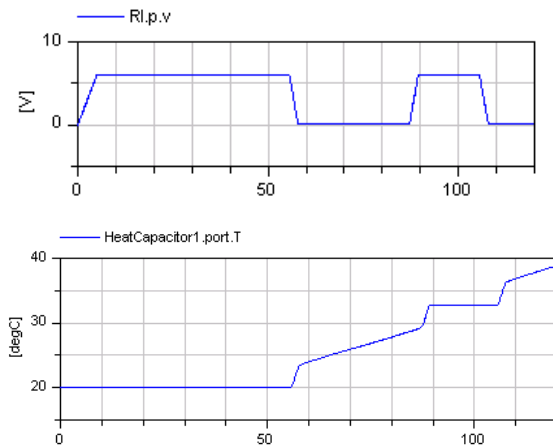
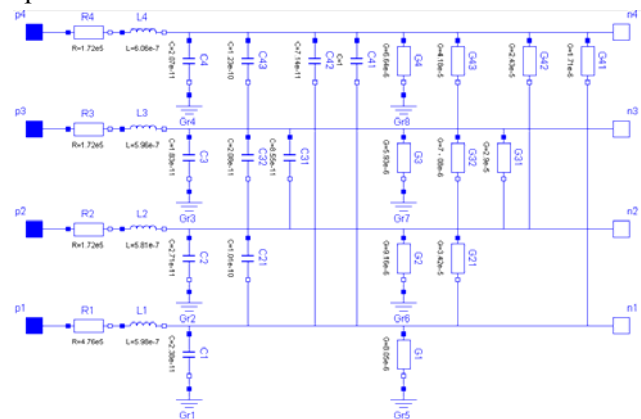Fig. 3: Thermal and electrical behavior

Fig. 5: RLGC segment

For the coupled inductors of the RLGC segment a special model **M_Transformer** was introduced (fig. 6). Added to the Basic subpackage it allows the modeling of multiple winding transformers. Its parameters are the number of inductors and the inductance matrix.
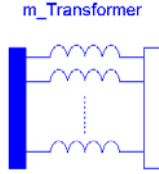

Fig. 6: Icon of the M_Transformer model

To demonstrate the multi line model behavior a four line model is connected with resistors and a voltage source according to fig. 7. Fig. 8 shows the expected behavior. The voltage rising of the first single line is temporarily coupled into the grounded single lines.
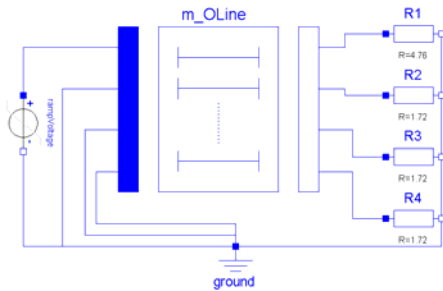
Fig. 7: Circuit containing a four line model
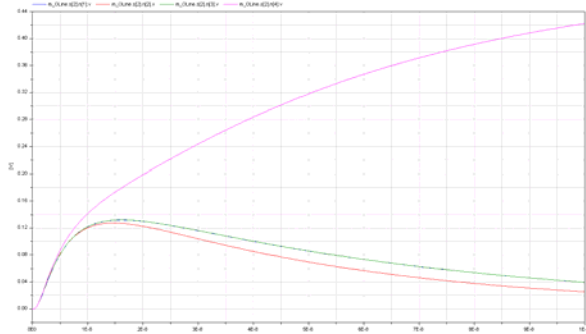
Fig. 8: Voltage coupling due to multiple line model

5   Zener diode

A simple Zener diode model [5] which is widely used for voltage stabilization was added to the semiconductor subpackage. Its equation combines the usual diode behavior with a breakthrough characteristic:

$$i = Ids(e^{v/Vt} - 1) - Ibv(e^{-(v+Bv)/(Nbv*Vt)})$$

Parameters are Ids (saturation current), Vt (voltage equivalent of temperature), Bv (Zener voltage), Ibv

(breakthrough knee voltage), and Nbv (breakthrough emission coefficient). A parallel ohmic resistance is added. If an exponent reaches the limit Maxexp the characteristic is continued linearly to avoid overflow. Fig. 9 shows a voltage stabilization circuit with a sine input. The voltage stabilization reached due to Zener diodes is shown in fig. 10.
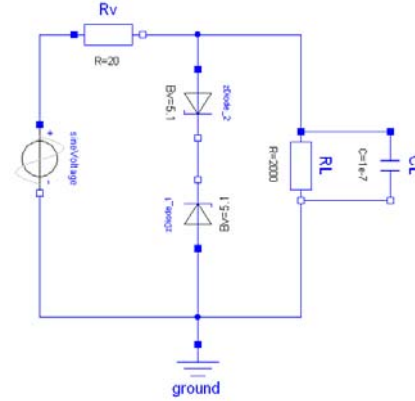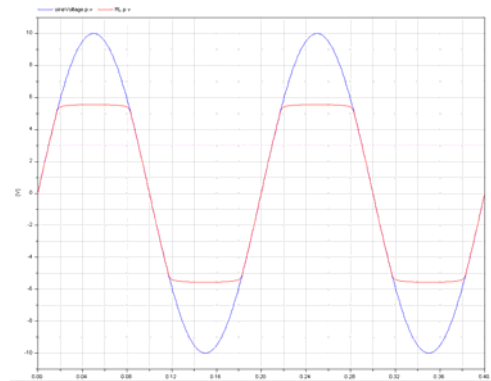

Fig. 9: Voltage stabilization circuit

Fig. 10: Stabilized voltage due to Zener diodes

6   Thyristor

Since the ideal thyristor models are very simple ones a more realistic, but still simple thyristor model was added. It has the three pins Anode, Cathode and Gate.

If the thyristor is in the blocking mode the behavior is like a linear resistor. It changes into the conduction mode if either the voltage between cathode and anode exceeds a certain value or a positive gate current flows a certain time.

There is no possibility to switch off the thyristor via the gate. It stays in the conducting mode until the anode current falls below the holding current value which is a parameter. If the voltage between anode and cathode is negative the model represents a diode with reverse breakthrough voltage.

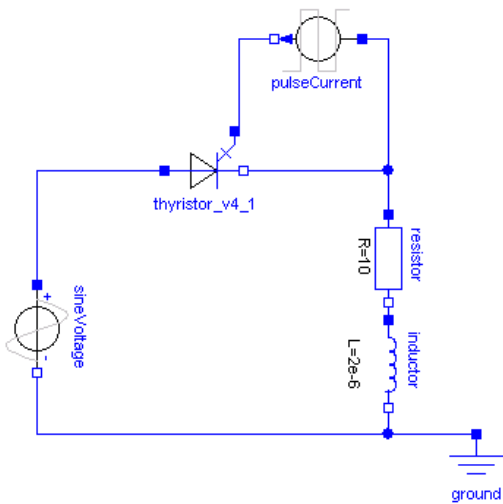The following simple test circuit (fig. 11) shows the behavior of the thyristor model.

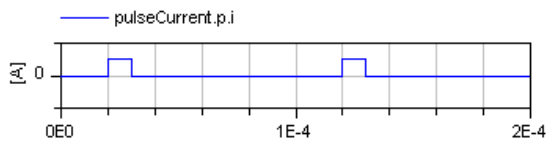Fig. 11: Thyristor behavior test circuit

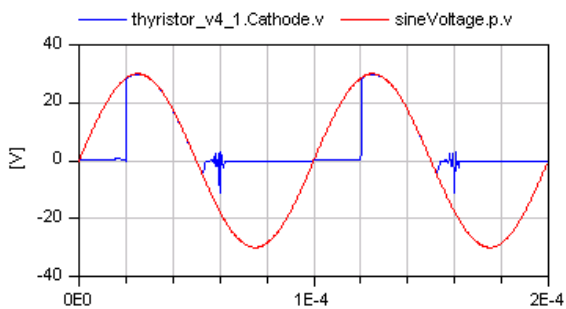

Fig. 12: Gate current of the thyristor test circuit

Fig. 13: Resulting behavior of the thyristor test

In fig. 13 the anode and cathode voltages are shown. If the gate current (fig. 11) rises, the thyristor becomes conducting and the voltage drop between anode and cathode is zero.

7 Operational Amplifier

In many cases the operational amplifier models of the MSL (IdealOpAmp, IdealOpAmpLimitted, OpAmp) are not sufficient. Therefore, a detailed model OpAmpDetailed was added, which is divided into five functional stages (input, frequency response, gain, slew rate, and output). Each stage contains a set of typical data sheet parameters, which are independent from parameters of the other stages. The model is described in detail in [6].

Fig. 14 shows an amplifier circuit using the OpAmpDetailed model. The results in fig. 15 show an amplification with voltage limitation.
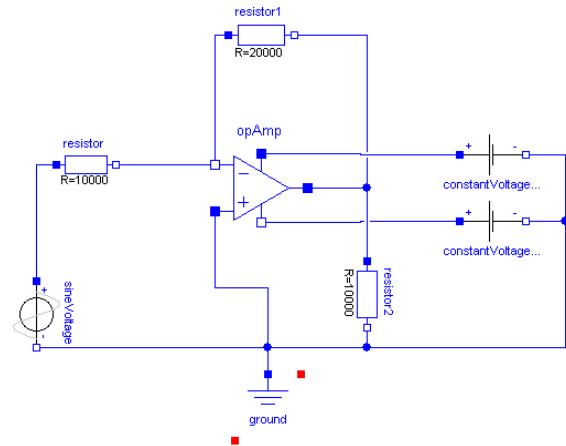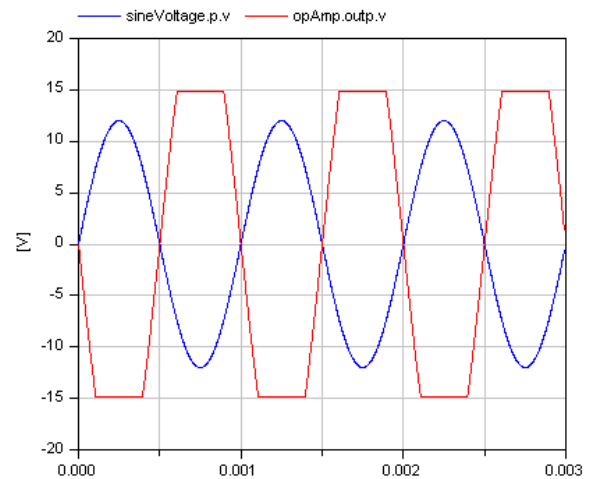

Fig. 14: Voltage amplifier circuit

Fig. 15: Amplification with voltage limitation

8 Switches with Electric Arc

The ideal switch models of the Ideal subpackage interrupt the current flowing through the switch within an infinitesimal time span. This causes numerical difficulties if an inductive circuit is connected since that current has to be differentiated. The voltage across the switch is only limited by the numerical solution methods. To improve this switches a simple electric arc model was introduced.

When the switch opens a voltage across the opened switch is impressed. This voltage starts with the initial arc voltage (V0) and rises with the arc voltage slope (dVdt) until the maximum arc voltage (Vmax) is reached. V0, dVdt and Vmax are parameters. The arc quenches if the current once reaches that current that would flow in the off-state of the switch. Then the off-state is activated.

Both Boolean controlled and voltage controlled openers and closers with arc model are added to the standard library.

Fig. 16 shows a comparison circuit between switches without and with the arc model. The arc model avoids the sudden switching (fig. 17).
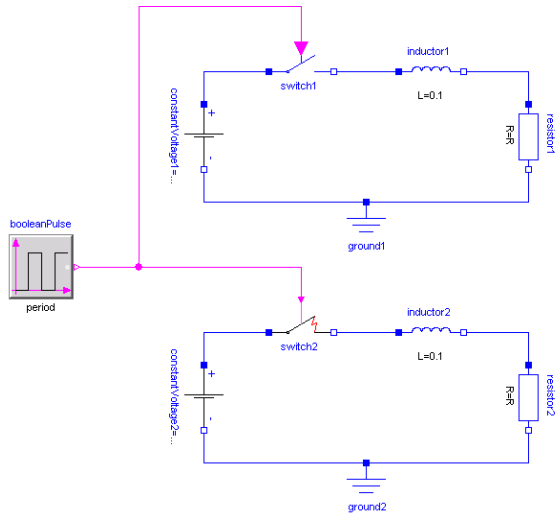

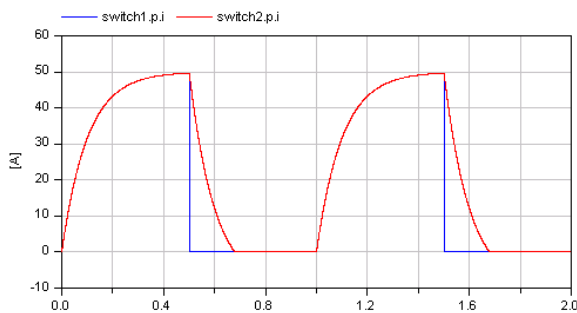Fig. 16: Comparison circuit of switches without and with electrical arc

Fig. 17: Switch currents of the comparison circuit

9 Translational EMF

Similar to the EMF model of the MSL Electrical.Analog.Basic library which transforms electrical energy ideally into rotational mechanical energy a TranslationalEMF model was introduced that transforms the electrical energy into translational mechanical energy. Fig. 18 shows the icons:

Fig. 18: Icons of the translational EMF model

If the parameter useSupport is set to false (default), the model is internally grounded. Therefore it has only one mechanical flange (left icon in fig. 18). Setting useSupport to true the model has two flanges without being internally grounded (right icon in fig. 18). With v being the voltage, i the current, vel the velocity and f the force the essential equations are:

```
k * vel = v;
f = - k * i;
```

k is the transformation coefficient.

10 Conclusions

The modelica electrical analog standard library was improved by a general usage of the smooth operator which increased the performance drastically. A conditional heat port allows thermal-electric simulation. A multiple line model, a multi inductor transformer model, a Zener diode model as well as thyristor and a detailed operational amplifier model, switches with an electric arc model and a translational EMF model were added.

Further models are planned to be added in future: IGBT, TRIAC and converter models. The MSL electrical analog library development is accompanied by the development of a library according to the Berkeley SPICE3 simulator devices.

Acknowledgement

References

[1] Clauß, C.; Leitner, T.; Schneider, A.; Schwarz, P.: *Modelling of electronic circuits with Modelica.* Proc. Modelica Workshop, Lund, Sweden, Oct. 2000, 3-11.

[2] http://www.modelica.org/libraries/Modelica

[3] Modelica – A unified object-oriented language for physical systems modeling. Language Specification 3.0, via http://www.modelica.org

[4] Smolyansky, D.; Klein, W.: Angewandte Simulation von verlustbehafteten digitalen Übertragungsleitungen. Test Kompendium 2004, 109-112

[5] Tietze, U.; Schenk, C.: Halbleiter-Schaltungstechnik. 12. Aufl., Springer-Verlag Berlin Heidelberg, 2002

[6] Conelly, J.A.; Choi, P.: Macromodelling with SPICE. Englewood Cliffs: Prentice Hall, 1992