C. Clauß, A. Schneider, T. Leitner, P. Schwarz:
Modelling of Electrical Circuits with Modelica.
Modelica Workshop 2000 Proceedings, pp. 3-12.

Paper presented at the Modelica Workshop 2000, Oct. 23.-24., 2000, Lund, Sweden.

All papers of this workshop can be downloaded from
http://www.Modelica.org/modelica2000/proceedings.html

Modelling of Electrical Circuits with Modelica

C. Clauß, A. Schneider, T. Leitner, P. Schwarz
Fraunhofer Institute for Integrated Circuits
Design Automation Department
Zeunerstraße 38, D-01069 Dresden, Germany
Email: {clauss, schneider, leitner, schwarz}@eas.iis.fhg.de

Abstract

One building block of the Modelica library is the electrical analog part. It contains important but simple models of electrical and electronic devices including very simple semiconductor device models. The models can be composed to any electrical circuit for which the model simplicity is sufficient enough. The electrical analog components are capable of interacting with the components of the other sublibraries of Modelica.

1 Overview

The electrical analog Modelica [14] library is a collection of electrical components, which are easy to use, easy to understand, and of a wide interest. It follows the simple components of the simulator SPICE [11], some of them are identical. Independent sources are suited to those of other physical domains. The library is devided into:

- time dependent sources (ramp, sine, exponentials, trapezoid et. al. for voltages and currents)
- basic models (resistor, capacitor, inductor, transformer, linear controled sources et. al.)
- semiconductor devices (diode, bipolar transistors, metal-oxide semiconductor FETs)
- line models
- ideal elements (switch, diode, opamp, transformer et. al.)

The symbols of most of the components correspond to the standard [10]. Some of the symbols can be seen in **fig. 1**.

2 Formulation Principles

The underlying modelling concept is n-pole approach [1]. The components are taken as n-poles which interact with their surrounding network part via their n pins. At each pin the electrical quantities current and voltage are defined. After the pins are defined the model equa-

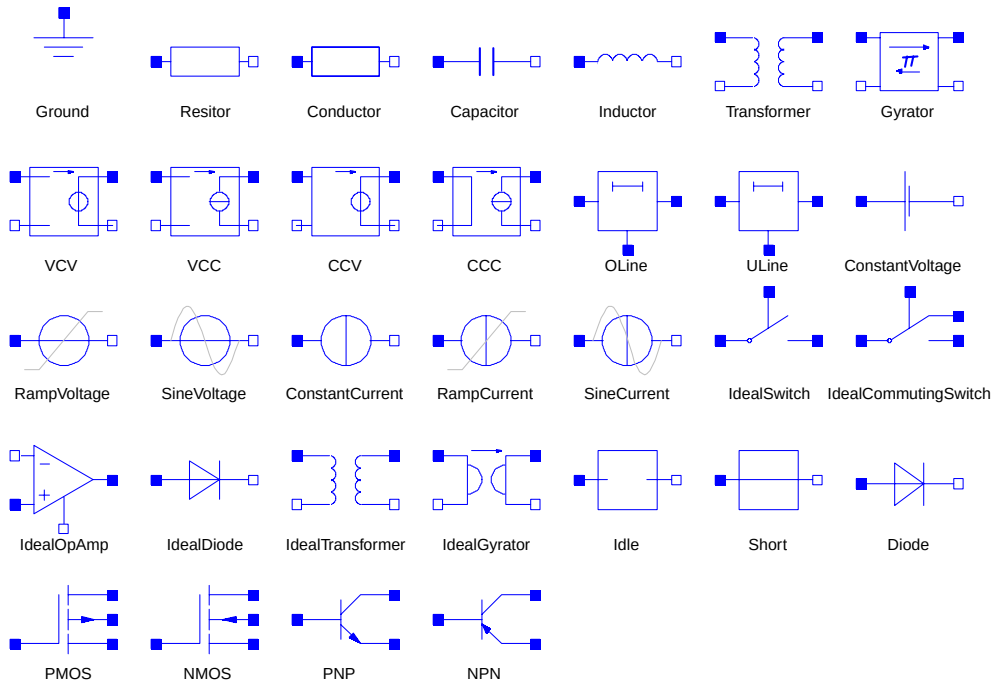

Figure 1: Components of the Modelica package *Modelica.Electrical.Analog*.

tions (terminal behaviour [13]) can be formulated using both the pin quantities and additional variables which are model internal. This is the behavioural description method. Another method is filling the n-pole by a subcircuit which is composed using other electrical components. Both methods can be combined.

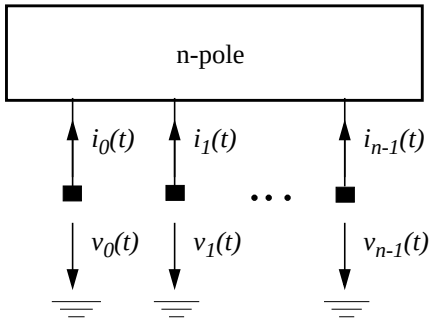

Figure 2: Pins and their quantities at an n-pole

Because of the n-pole concept interfaces are defined. The basic interface definition is the pin, which is a connector. At the pin the pin voltage v (between pin and ground node), and the pin current i are defined. The current is of the flow- type, because the current sum has to be zero if pins are connected.

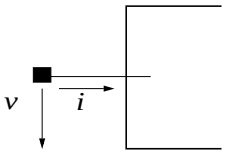

Figure 3: Pin

The Modelica pin definition is*:

```
connector Pin
  SIunits.Voltage v
            "Potential at the pin";
  flow SIunits.Current i
            "Current flowing into the pin";
end Pin;
```

Furthermore, positive and negative pins are defined which differ in their graphical representation only. Throughout the library there are some often used pin patterns, which can be extracted to typical interfaces. These are the TwoPin, the OnePort, and the TwoPort.

* In this paper the Modelica source code examples use an abbreviated notation of components taken from the Modelica packages Modelica.SIunits and Modelica.Electrical.Analog. The complete Modelica source code of all examples can be found at http://www.eas.iis.fhg.de/sim/publications/papers/2000/028/

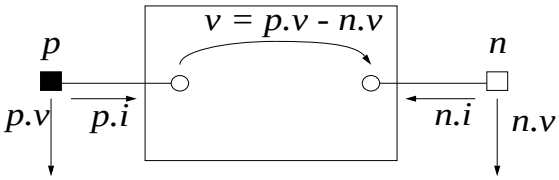

Figure 4: TwoPin

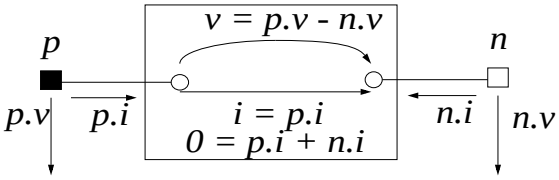

Figure 5: OnePort

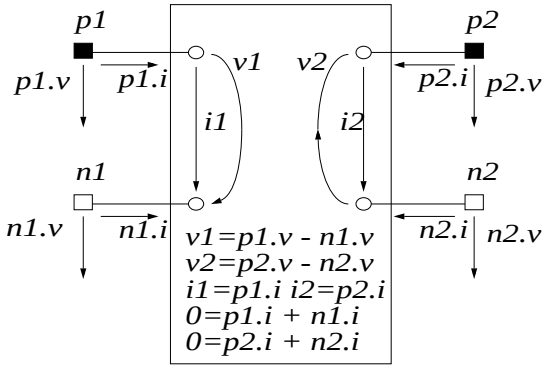

Figure 6: TwoPort

The TwoPin interface does not contain the current sum equation as the OnePort interface does. Therefore, the TwoPin interface is suitable for being filled with other library components, which guarantee the current sum equation. e.g. a subcircuit. The OnePort is suitable for any mathematical v/i characteristic formulation. Another often used Interface is the TwoPort. The interface definitions are part of the electrical analog Modelica library.

Using the predefined interface types most of the library components have the following structure:

- usage of the interface type, sometimes additional pin declarations
- declaration of parameters
- declaration of internal variables
- equation part (connections or directly formulated)
- comments, annotations, and graphical information can be added

The Modelica description of the inductor is an example for a OnePort component definition which does not use internal variables:

```
model Inductor
  extends  OnePort;
  parameter SIunits.Inductance L=1;
  equation
    L*der(i) = v;
end Inductor;
```

The description of an ideal diode is a OnePort which needs an internal variable within the component:

```
model IdealDiode
  extends OnePort;
  parameter SIunits.Resistance Roff
      (final min=0) = 1.E-5;
  parameter SIunits.Conductance Gon
      (final min=0) = 1.E-5;
  Boolean off(start=true)
       "Switching state of diode";
protected
  Real s "Auxiliary variable";
equation
  off = s < 0;
  v = s*(if off then 1 else Roff);
  i = s*(if off then Gon else 1);
end IdealDiode;
```

Independent sources are OnePorts which use timedependent mathematical functions for defining either the quantity v (voltage sources) or the quantity i (current sources).

An example for a TwoPort component is the transformer model. It combines the v1, v2, i1, i2 values defined in the interface model.

```
model Transformer
  extends Analog.Interfaces.TwoPort;
  parameter SIunits.Inductance L1=1;
  parameter SIunits.Inductance L2=1;
  parameter SIunits.Inductance M=1;
equation
  v1 = L1*der(i1) + M*der(i2);
  v2 = M*der(i1) + L2*der(i2);
end Transformer;
```

In this manner most of the components are modeled. Components with other pin patterns are described using the pin connector directly, e.g. in the PMOS model:

```
model PMOS
  Interfaces.Pin D, G, S, B;
  parameter SIunits.Length W=50.0e-6;
  parameter SIunits.Length L=8.0e-6;
  parameter SIunits.Transconductance
      Beta=0.0085;
  parameter SIunits.Voltage Vt=-0.15;
  parameter Real K2=0.41;
  parameter Real K5=0.839;
  parameter SIunits.Length dW=-3.8e-6;
  parameter SIunits.Length dL=-4.0e-6;
protected
  Real v, uds, ubs, ugst, ud,us,id;
equation
  v = Beta*(W + dW)/(L + dL);
  ud = if (D.v > S.v) then S.v else D.v;
  us = if (D.v > S.v) then D.v else S.v;
  uds = ud - us;
  ubs = if (B.v < us) then 0 else B.v - us;
  ugst = (G.v - us - Vt + K2*ubs)*K5;
  id = if (ugst >= 0)
       then v*uds*1.e-7
       else if (ugst < uds)
       then -v*uds*(ugst - uds/2 - 1.e-7)
       else -v*(ugst*ugst/2 - uds*1.e-7);
    G.i = 0;
    D.i = if (D.v > S.v) then -id else id;
    S.i = if (D.v > S.v) then id else -id;
    B.i = 0;
end PMOS;
```

3 Usage

The components of the electrical analog library can be combined to circuits as they are. This is possible on the ASCII text level, or in a graphical way.
Due to the strict objectoriented language design of Modelica an extended library usage is possible:

- Components can be connected to create new models. In this way, a hierarchical description of large circuits is possible.
- The components can be modified by changeing the mathematical description, adding pins, introducing parameters, fixing parameters, ... Especially the possibilities of inheritance [2] allow the creation of own circuit components in a comfortable way.
- Since the description of the library components is totally public they can be useful as examples for components which are to be modelled.

The whole variety of the possibilities of the library usage is not presented. It has to be derived from the language reference manual. Several aspects of the library usage are demonstrated in the examples.

4 Examples

The examples are part of an extensive validation suite. They are chosen to indicate the variety of circuits that can be described using the library. Furthermore, the examples show the complexity which was under test until now. Further tests e.g. with circuits of hundreds of transistors are necessary.

The examples are simulated using the simulator Dymola [12]. Our experience in simulating the examples is:

- Sometimes the tolerance has to be chosen carefully.
- Some examples require the condition of translation ‘Evaluate true’ (e.g. if capacitors are zero).
- most of the simulation problems arose with the usage of NPN/PNP semiconductor devices.

4.1 Resistor-Capacitor-Circuit

The simple resistor-capacitor circuit is usually the beginner model for evaluating electrical analog simulation systems.

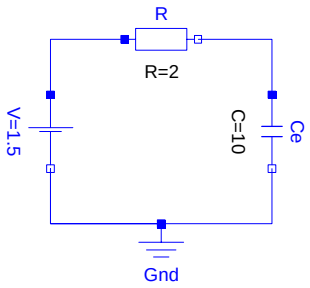

Figure 7: Simple RC circuit.

The following Modelica text shows the circuit description without graphical instructions. The instantiation of library components can be seen as well as the usage of parameters. Once instantiated the circuit elements are connected in the equation part.

```
model SimpleCircuit
  ConstantVoltage V(V=1.5);
  Resistor R(R=2);
  Capacitor C(C=10);
  Ground Gnd;
equation
  connect(V.p, R.p);
  connect(R.n, C.p);
  connect(V.n, C.n);
  connect(V.n, Gnd.p);
end SimpleCircuit;
```

The simulation result is the behaviour of voltage and current of a capacitor which is charged.

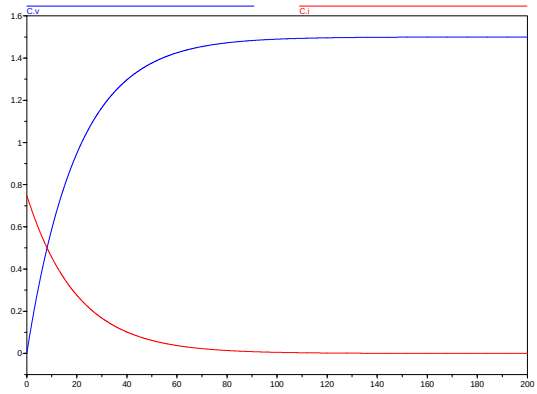

Figure 8: Simulation results

4.2 Diodes

The diode example compares a ideal diode with a real one. Both of the diodes are components of the library. The real diode (Diode) is a semiconductor device. Its current/voltage characteristic is an exponential function which is continuated linearly above a certain exponent which is the parameter Maxexp. The ideal diode (IdealDiode) uses the declarative description method [1], [8], [9]. If the parameters Roff, and Gon are zero the ideal diode is an idle running branch in the open case, and a chort cut in the clodes case. The following simple circuit uses both of the diodes. The parameter are chosen such that differences in the diode behaviour can be seen easily.

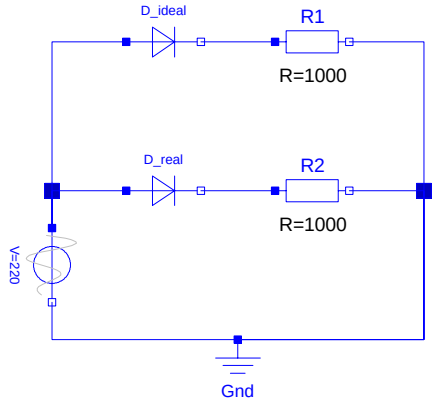

Figure 9: Comparision of real and ideal diode.

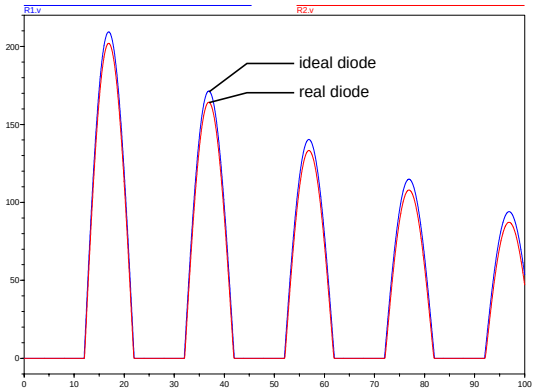

Figure 10: Simulation results of diode example.

4.3 MOS Oscillator

Obviously an odd number of inverters within a chain tends to oscillate. A single inverter is composed by MOS transistors. It has its own voltage source for power supply, and a capacitor. The Modelica description of the inverter is:

```
model Inverter
  Pin  inn  "input pin";
  Pin  outt "output pin";
  RampVoltage VDD
    (V=5, duration=50e-9);
  PMOS TP
    (W=6.5e-6,L=3.1e-6,Beta=1.05e-5,
     Vt=1,K2=0.41,K5=0.8385,dW=-2.5e-6,
     dL=-2.1e-6);
  NMOS TN
    (W=4.5e-6, L=2.5e-6,Beta=4.1e-5,
     Vt=0.8,K2=1.144,K5=0.7311,
     dW=-2.5e-6,dL=-1.5e-6);
  Capacitor Ce(C=0.2e-12);
  Ground Gnd;

equation
  connect(inn,TP.G);
  connect(inn,TN.G);
  connect(VDD.p,TP.D);
  connect(TP.S,TN.D);
  connect(TP.S,Ce.p);
  connect(TN.S, Gnd.p);
  connect(Ce.n,Gnd.p);
  connect(VDD.n, Gnd.p);
  connect(TN.B, Gnd.p);
  connect(TP.B,VDD.p);
  connect(outt, TP.S);
end Inverter;
```

Once the inverter is defined it can be instantiated manifold and connected to a loop. In the following model seven inverters form a loop including one resistor, the eighth inverter is attached out of the loop.

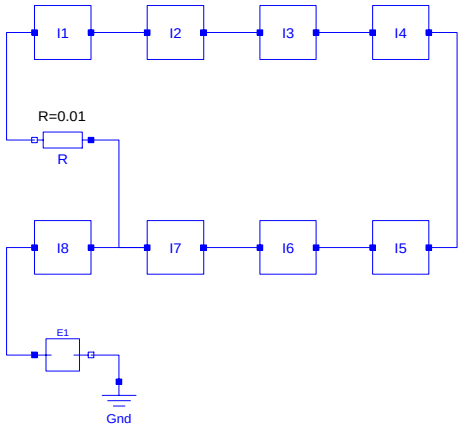

Figure 11: Oscillator.

```
model Oscillator
  Inverter I1, I2, I3, I4, I5, I6, I7, I8;
  Resistor R(R=0.01);
  Idle  E1;
  Ground Gnd;

equation
  connect(R.n,I1.inn);
  connect(I1.outt,I2.inn);
  connect(I2.outt,I3.inn);
  connect(I3.outt,I4.inn);
  connect(I4.outt,I5.inn);
  connect(I5.outt,I6.inn);
  connect(I6.outt,I7.inn);
  connect(I7.outt,I8.inn);
  connect(I8.outt,E1.p);
  connect(E1.n,Gnd.p);
  connect(I7.outt,R.p);
end Oscillator;
```

When the power voltage rises at once the circuit starts oscillating.

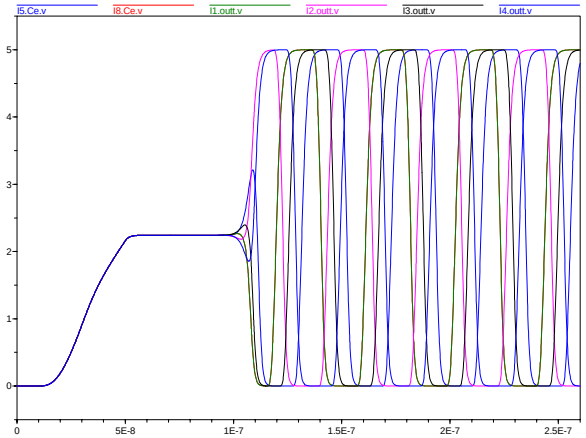

Figure 12: Simulation results of Oscillator.

4.4 Operational Amplifier µA741

A PID device amplifies (**p**roportional), **i**ntegrates, and **d**ifferentiates the input signal. As a weighted sum the three results form the output signal of the device. If the following circuit scheme is used the V_{in} voltage is PID-like treated

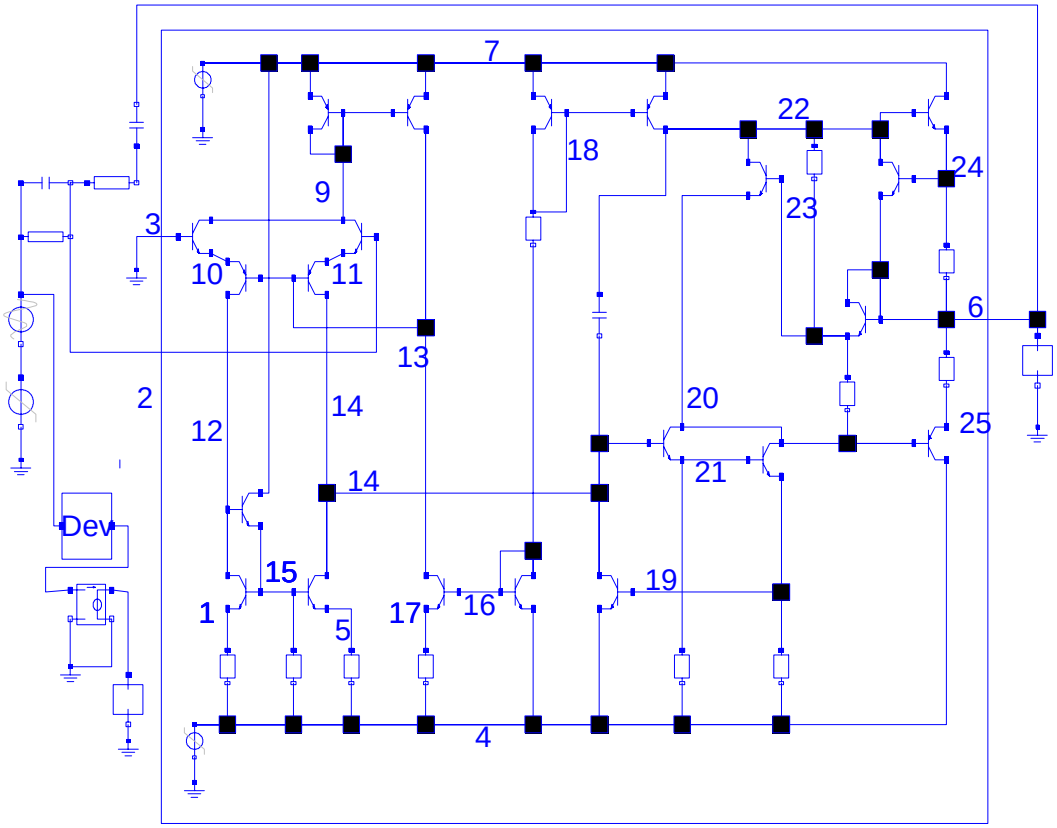

Figure 14: PID circuit with Operational amplifier µA741, and ideal PID device

$$V_{out} = PV_{in} + I\int V_{in}dt + D\frac{dV_{in}}{dt}$$

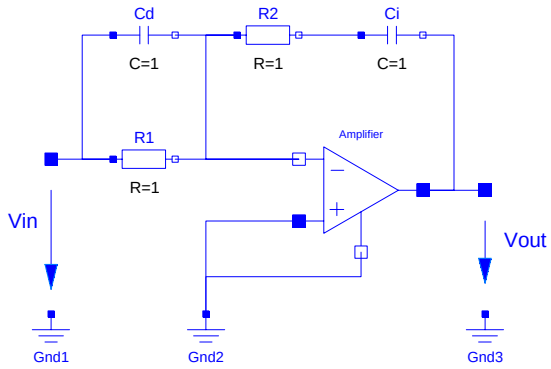

Figure 13: PID

The coefficients P, I, and D can be simply derived to be

$$P = \frac{R_2}{R_1} + \frac{C_D}{C_I}$$

$$I = \frac{1}{C_I R_1}$$

$$D = C_D R_2$$

Unfortunately, this ideal PID behaviour requires an ideal operational amplifier in the circuit. If the uA741 [6] amplifier is used instead of the ideal one the PID behaviour is limited. E.g. if V_{out} runs into saturation the PID behaviour does not longer exist. To demonstrate this a model is simulated which contains both the PID circuit using the uA741 and a pure mathematical device which calculates the ideal PID behaviour (**fig. 14**). In the plot the point can be seen where ideal and real behaviour diverge.

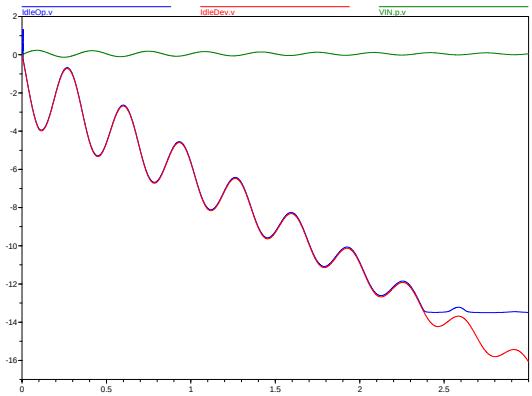

Figure 15: Real, and ideal PID behaviour

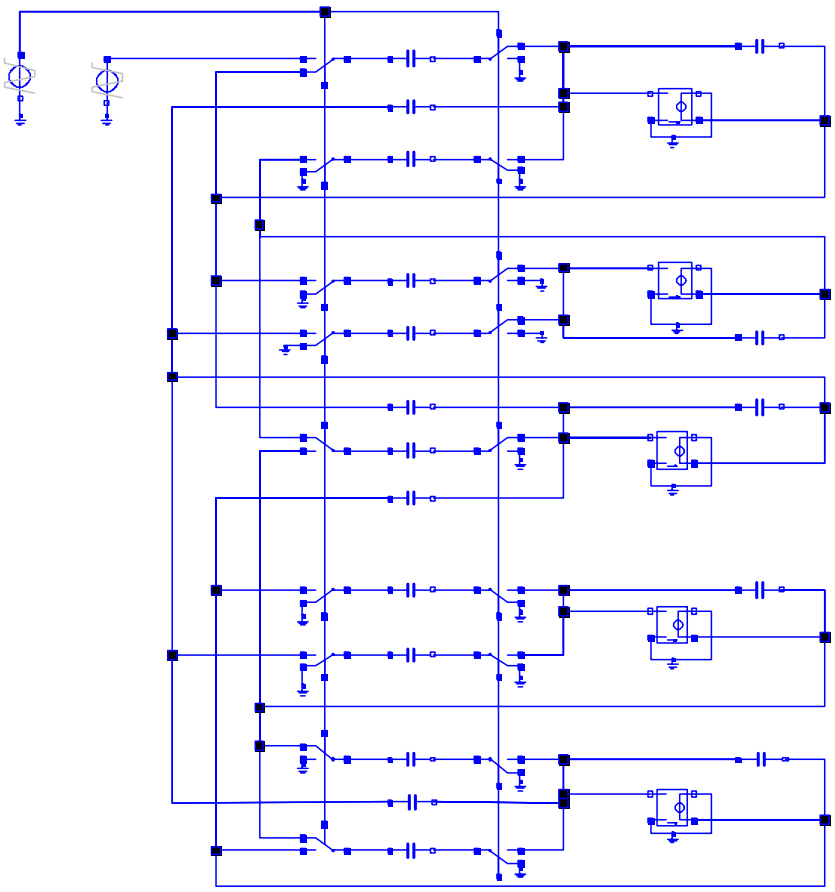

Figure 16: Cauer filter

4.5 Cauer's Filter

The switched capacitor [4] Cauer's filter model [5] is a low-pass-filter of the order five. It uses voltages sources, capacitors, and voltage controlled voltage sources of the electrical analog library. To demonstrate a slightly modification of a library component the commuting switches are extended. At the positive pin a resistor R is added. The extended switch is called RealComSwitch. Like the ideal switch it has the same pins p, n1, n2, and control. The symbol is taken from the ideal switch. Without any graphical information the Modelica description of the extended switch is:

```
model RealSwitch
  Pin p "positive pin";
  Pin n1 "negative pin n1";
  Pin n2 "negative pin n2";
  Pin control "control pin";
  Resistor R(R=0.01)  "additional resistor";
  ControlledIdealCommutingSwitch
      S(level=2.5);
equation
  connect(p, R.p);
  connect(R.n, S.p);
  connect(n1, S.n1);
  connect(n2, S.n2);
  connect(control, S.control);
end RealSwitch;
```

Fig. 16 shows Cauer's filter in a graphical way. As an example a pulse response is simulated. The small oscillations are caused by the switching clock signal.

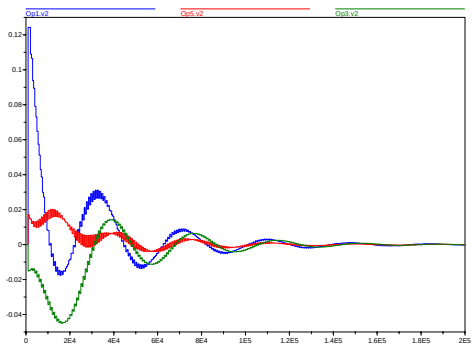

Figure 17: Pulse response

4.6 Chua's circuit

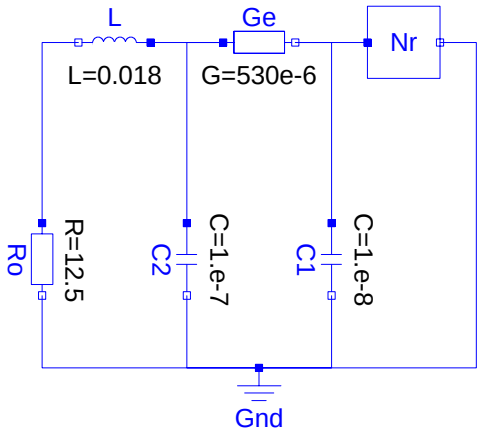

Figure 18: Chua ...

If the initial values of the capacitors of Chua's Circuit [3] are not zero the node voltages at the conductor show chaotic behaviour. The only nonlinear component in the circuit which causes the chaos is Chua's Diode. It is a one-port. Its characteristic can be seen in the following picture:

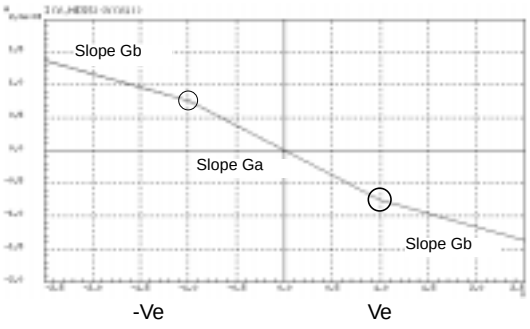

Figure 19: i(v)-characteristic of Chua's Diode

The characteristic depends on the parameters Ga, Gb, and Ve. Because of its originality this component, which is not part of the electrical analog library, is not derived from a library component. But it uses the definition of a OnePort. Therefore, the branch quantities v and i are predefined. The following Modelica text describes Chua's Diode:

```
model ChuasDiode
  extends OnePort;
  parameter SIunits.Conductance Ga;
  parameter SIunits.Conductance Gb;
  parameter SIunits.Voltage Ve;
equation
  i=if(v<-Ve) then Gb*(v+Ve)-Ga*Ve
    else if(v>Ve) then Gb*(v-Ve)+Ga * Ve
    else Ga*v;
end ChuasDiode;
```

Once defined Chua's Diode can be used in the circuit. The Modelica description of the circuit without graphical information is:

```
model ChuaCircuit
   Inductor L(L=0.018);
   Resistor Ro(R=12.5);
   Conductor Ge(G=565e-6);
   Capacitor C1(C=1.e-8, v(start=4.0));
   Capacitor C2(C=1.e-7);
   ChuasDiode Nr(Ga=-757.576e-6,
                 Gb=-409.091e-6, Ve=1);
   Ground Gnd;

  equation
    connect(Ro.n,Gnd.p);
    connect(Ro.p,L.n);
    connect(L.p,Ge.p);
    connect(Ge.n,Nr.p);
    connect(Nr.n,Gnd.p);
    connect(C1.p,Ge.n);
    connect(C1.n,Gnd.p);
    connect(C2.p,Ge.p);
    connect(C2.n,Gnd.p);
end ChuaCircuit;
```

The simulation result (until t=0.05) shows the typical behaviour:

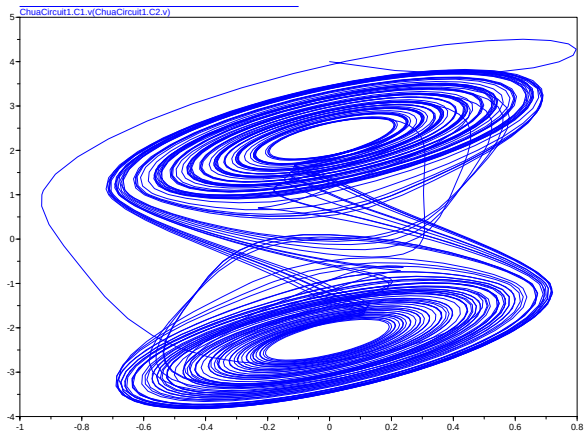

Figure 20: Simulation results of Chua circuit.

5 Conclusion

Once the library exists some improvements become more and more necessary:

- multidomain components like converters (electric-

mechanic, electric-magnetic)
- a library part of precomposed circuits (e.g. integrators, operational amplifiers, gates (AND, NOR) ...)
- the SPICE library, and a compiler which converts spice netlists into a Modelica description

To come to a SPICE library the most desirable way is formulating the SPICE components in Modelica. Because of the complexity of most of the SPICE semiconductor devices this way seems to be very time-consuming. At the moment we would prefer using the foreign function call possibility to provide the SPICE components in the form of C language code.

6 References

[1] Clauß, C.; Haase, J.; Kurth, G.; Schwarz, P.: Extended Amittance Description of Nonlinear n-Poles. Archiv für Elektronik u. Übertragungstechnik 49(1995)2, 91-97

[2] Clauß, Chr.; Leitner, Th.; Schneider, A.; Schwarz, P.: Object-oriented modeling of physical systems with Modelica using design patterns. Workshop on System Design Automation SDA 2000, Rathen, March 13-14, 2000, 209-216

[3] Kennedy, M.P.: Three Steps to Chaos - Part I: Evolution. IEEE Transactions on CAS-I 40(1993) 10, 640-656

[4] de Man, H.; Arnout, G.;Vandevalle, J.: Practical Implementation of a General Computer Aided Design Technique for Switched Capacitor Circuits. IEEE Journal Solid-State Circuits SC15(1980)2, 190-200

[5] Fehlauer, E.; Krauß, M.: Ein effektiver Algorithmus zur Zeitbereichsanalyse von SCOV-Schaltungen. Wiss. Z. Techn. Univers. Dresden 35(1986)H.4, 159-163

[6] Herpy, M.: Analoge integrierte Schaltungen. Akadémiai Kiadó. Budapest 1976

[7] Leitner, Th.: A new approach for semiconductor models basing on SPICE model equations. Proc. ECS'97, Bratislava, Slovakia, 4./5. Sept. 1997, 119-123

[8] Otter, M.; Elmqvist, H.; Mattsson, S.E.: Hybrid modeling in Modelica based on the synchronous data flow principle. CACSD'99, Aug. 22.-26. Aug., Hawaii, 1999

[9] Otter, M.; Elmqvist, H.; Mattsson, S.E.: Objektorientierte Modellierung physikalischer Systeme, Teil 8. at Automatisierungstechnik 47(1999)9

[10] Normen über graphische Symbole für die Elektrotechnik, Schaltzeichen. DIN-Taschenbuch 514, Beuth Berlin, Wien, Zürich, 1994

[11] Johnson, B.; Quarles, T.; Newton, A.R.; Pederson, D.O.; Sangiovanni-Vincentelli, A.:SPICE3 Version 3e, User's Manual., Univ. of California, Berkeley, Ca., 94720, 1991

[12] Dymola: http:/www.Dynasim.se

[13] Reibiger, A.: On the terminal behaviour of networks. Proc. ECCTD '85, Prague, Sept. 1985, 224-227

[14] Elmqvist, H. et al.: Modelica - A Unified Object-Oriented Language for Physical Systems Modeling. Version 1.3, December 1999. http://www.Modelica.org