Proceedings
of the 3rd International Modelica Conference,
Linköping, November 3-4, 2003,
Peter Fritzson (editor)

Peter Beater and Christoph Clauss
University of Applied Sciences Südwestfalen, Soest; Fraunhofer-Institut für Integrierte Schaltungen, Dresden:

Multidomain Systems: Pneumatic, Electronic and Mechanical Subsystems of a Pneumatic Drive Modelled with Modelica

pp. 369-376

Multidomain Systems: Pneumatic, Electronic and Mechanical Subsystems of a Pneumatic Drive Modelled with Modelica

Peter Beater
Beater@fh-swf.de
Department of Mechanical Engineering – Automation
University of Applied Sciences Südwestfalen, Campus Soest
Lübecker Ring 2, D 59494 Soest, Germany

Christoph Clauß
Christoph.Clauss@eas.iis.fhg.de
Fraunhofer-Institut für Integrierte Schaltungen, Außenstelle Entwurfsautomatisierung
Zeunerstraße 38, D 01069 Dresden, Germany

Abstract

The simulation of pneumatic or electronic systems has been state of the art for a long time. For both of these domains there exist highly specialized simulation programs which can be regarded as a kind of industrial standards. However, often problems arise if different domains of technology occur within one system and very detailed models are needed.

As an example a pneumatic drive is presented that is used for teaching mechanical engineering students in Soest. It consists of pneumatic, mechanical, and electronic components. Each component is modelled very detailed using the Modelica language (Modelica, 2002). Without coupling of simulators the complete simulation model can be investigated by *one* tool.

1 Introduction

The engineer of today is used to powerful simulation tools. Within the last forty years these tools mutated from simple solvers of differential equations to computer-aided design software for technical systems. Tools like HSPICE in electronics, ADAMS in mechanics, or HOPSAN in hydraulics are highly specified to meet the needs of the discipline. These tools “know“ the domain-intern peculiarities. Often the models and the simulation algorithms are closely related. Therefore, these tools are very advantageous in simulation, modelling, and postprocessing.

However, often problems arise if technical systems cover more than one established discipline, e. g. in automotive systems or in microsystems engineering. The two fundamental ways out are coupling of simulators, and compact modeling for one simulator.

From the very beginning the Modelica language has been designed for covering several technical disciplines. Complex systems can be modelled with one language to get one model. The further processing within the tool, e. g. the Dymola simulator, results in one mathematical model, typically a system of differential algebraic equations, which is solved by one simulation engine. The challenge of the Modelica approach is to show that its efficiency is not much worse than the efficiency of domain specific tools. To offer evidence of this is surely a long process (Clauss and Beater, 2002). In this paper the multidomain example of an electronically controlled pneumatic drive is presented. It demonstrates that the unified multidiscipline simulation tool Modelica/Dymola meets the challenge quite well.

At first the physical device is presented with emphasising the pneumatic and electronic parts. The Modelica model is shortly described, and simulation results are discussed. It is shown that numerical problems could be solved, and the performance can be accepted.

2 The Pneumatic Drive

Fig. 1 shows the pneumatic drive. It is a typical construction when a part has to be moved for several decimetres, e. g. in material handling. The required forces determine the diameter of the cylinder which is connected to the electrically operated directional control valve. At the ends of the cylinder magnetic switches are installed that signal the end of stroke to the electronic controller. For the controller standard CMOS ICs are used. The “programming” is done by connecting the logical blocks (AND, OR, RS). The task is to begin a repeated extending and retracting of the piston after the start button has been pushed and to stop in the extended position after the stop button has been pushed.

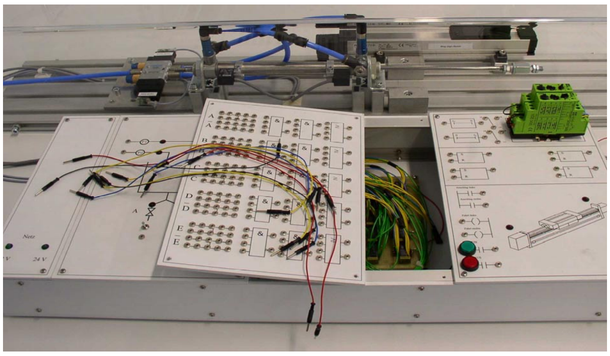

Figure 1 Pneumatic drive as laboratory set-up in Soest

3 The Pneumatic and Mechanic Parts

After preliminary trials using the analogue computer in the fifties the digital simulation of fluid power systems, i. e. hydraulic or pneumatic systems, became important in the eighties. Graphical user interfaces were added in the nineties. Using Modelica and its libraries it is easy to model pneumatic or mechanic systems. The user doesn't need to know all the details of component modeling. If nevertheless details are essential the source code of the models is available. Using models from the Modelica libraries the pneumatic drive according to Fig. 1 could be modelled.

Basically, three physical laws are needed to describe the movement of the piston. The first one is Newton’s second law that models the movement of the mass (piston) as a function of the applied forces. It is described in the block SlidingMass of the Modelica library Mechanics.Translational. The forces can be calculated from the pressures in the cylinder chambers, which are described by:

$$m \cdot \dot{T} + T \cdot \dot{m} = -\frac{1}{c_V} \cdot \left|p_{Chamber}\right| \cdot \dot{x}(t) \cdot A_K - \frac{1}{c_V} \cdot \alpha \cdot A_W \cdot \left(T - T_A\right) + \kappa \cdot T_{surround} \cdot \dot{m} \qquad (1)$$

with:
m gas mass [kg]
T temperature in the chamber [K]
c_v specific heat capacity [J/(kg · K)]
$p_{chamber}$ cylinder pressure [Pa]
A_W heat transfer area [m²]
A_K piston area [m²]
κ ratio of specific heat capacities
$T_{surround}$ temp. of the environment [K]
α coefficient of heat transfer [W/(m² · K)]

These equations are modelled in the library model PneuLib.Chamber. Two Chamber models, the SlidingMass for the piston and a Rod for the housing describe a simple cylinder. A complex model of a double sided cylinder with pneumatic stroke cushioning is shown in Fig. 2.

The mass flow rate to or from the cylinder chambers depends on the pressure upstream and downstream of the valve, p_1 and p_2, and the electrical command signal for the valve. For the opened valve the mass flow rate can be described by the equation of flow through a nozzle:

$$\dot{m} = p_1 \cdot C \cdot \rho_0 \cdot \sqrt{\frac{T_0}{T_1}} \cdot \sqrt{1 - \left(\frac{\frac{p_2}{p_1} - b}{1 - b}\right)^2} \qquad \text{for } \frac{p_2}{p_1} > b$$

$$\dot{m} = p_1 \cdot C \cdot \rho_0 \cdot \sqrt{\frac{T_0}{T_1}} \qquad \text{for } \frac{p_2}{p_1} \leq b \qquad (2)$$

with:
$\dot{m}$ mass flow rate [kg/s]
p_1 upstream pressure [Pa]
C sonic conductance [m³ / s / Pa]
ρ_0 standard density of air [kg/m³]
T_0 standard air temperature [K]
T_1 air temperature upstream [K]
p_2 downstream pressure [Pa]
b critical pressure ratio [1]

This equation is standardized in ISO 6358. Necessary is also a state equation for air, where the ideal gas law is used:

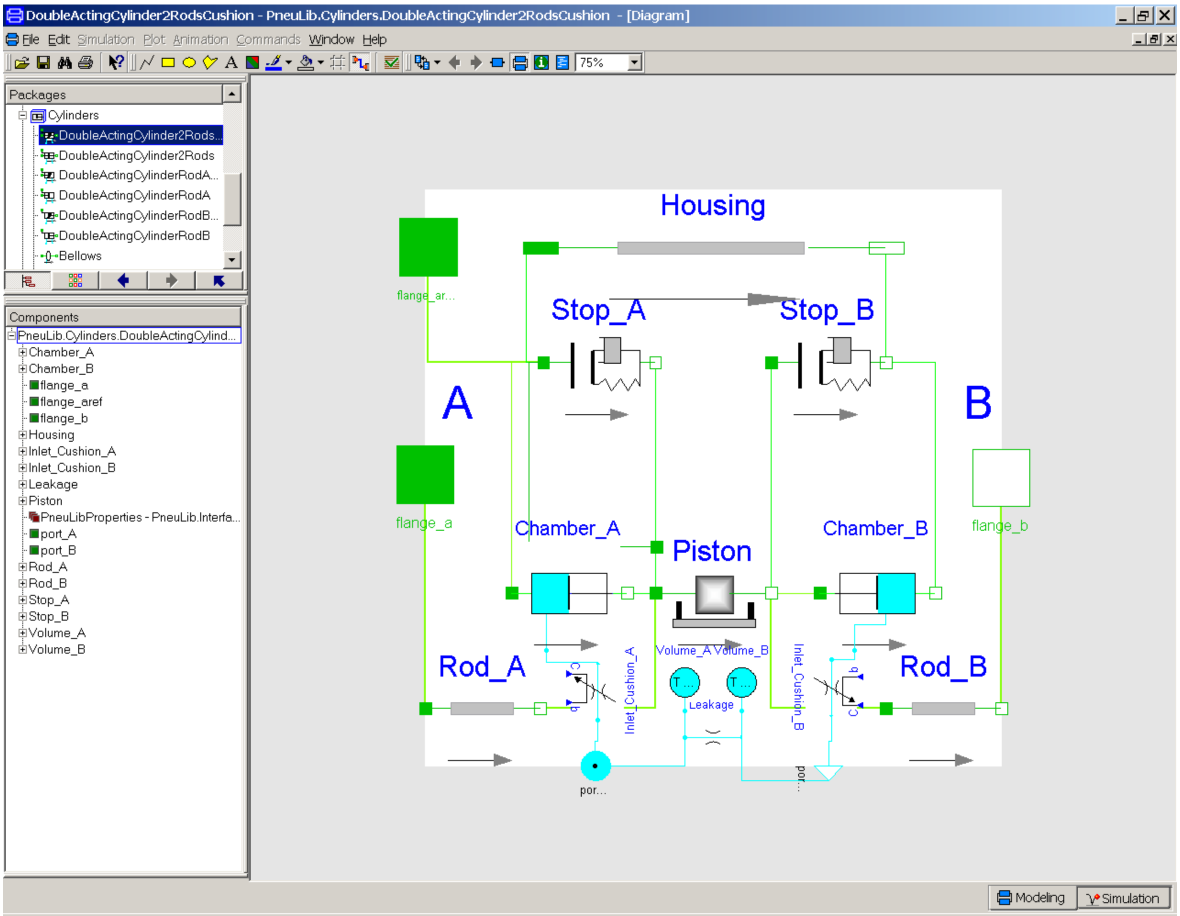

Figure 2 Object diagram of a double-sided two-ended cylinder with stroke cushioning screenshot from Dymola

$$p \cdot V = m \cdot R \cdot T \qquad (3)$$

with: p pressure [Pa]
V volume [m^3]
m air mass [kg]
R specific gas constant [J/(kg $\cdot$ K])
T temperature [K]

For typical operating conditions of pneumatic drives, maximum pressure of about 1 MPa and a temperature around 20 ° C, the differences between air and the modelled ideal gas are negligible. Additional equations are needed that describe e. g. the dynamic behaviour of the directional control valve or the stroke cushioning of the cylinder.

To couple component models of the library PneuLib the through variable mass flow rate $\dot{m}$ and the across variable pressure is used. To have ports for the inflow and outflow of air two connectors are defined:

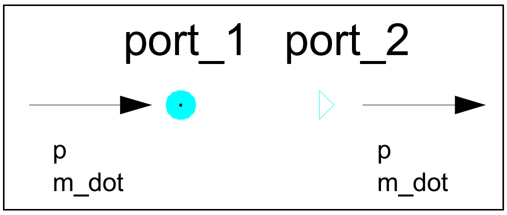

Figure 3 Connectors for port 1 (pressure supply) and port 2 (work)

An example of a simple component is a nozzle that is based on Eq. 2. While this equation is very well suited for measurement purposes it leads to problems when used in the digital simulation of pneumatic systems because the „gain“, i. e. the quotient $d\dot{m} / d\,dp$, goes to infinity as the pressure drop, $dp = p_1 - p_2$, goes to zero. This effect is known from models for incompressible hydraulic oil that use the simple „square root“ dependency

$$q \sim \sqrt{\Delta p} \qquad (4)$$

and has led to the development of more accurate models (Beater 1999). In the pneumatics library the nozzle model according to Eq. 2 is used because it is a generally accepted standard but extended for the region of small pressure differences by a linear relationship between mass flow rate and pressure differential. This is based on the fact that then the turbulent flow becomes laminar and therefore a linear relationship exists between pressure differential and flow rate. This is also an example that simple “textbook” models are not suited for real engineering tasks but have to be extended to avoid numerical problems during integration. Figures 4 and 5 show the icon and the structure of the code. Used is the superclass TwoPortComp that defines all parts that are needed for components with two ports but no mass storage.

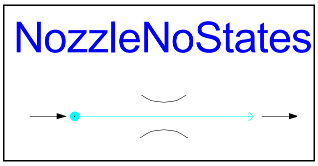

Figure 4 Icon of nozzle model

```
model NozzleNoStates
        "Nozzle model according to ISO 6358."
  extends PneuLib.Interfaces.TwoPortComp;
  parameter SonicConductance C=1e-7
        "sonic conductance";
  parameter CriticalPressureRatio b=b_default
        "critical pressure ratio";
  Real pressure_ratio(start=1.0);
  ...
  equation
    pressure_ratio = port_2.p/port_1.p;
    m_dot = ... ;
end NozzleNoStates;
```

Figure 5 Code of nozzle model

As in the hydraulics library HyLib (HyLib, 2002) there are also components that have lumped volumes directly connected at all pneumatic ports, see e. g. the object diagram of the cylinder in Fig. 2. This modeling concept allows pneumatic components to be connected in an arbitrary way, e. g. in series or in parallel, just by drawing connection lines, no special components for splits or mergers are required.

The advanced features of Modelica 2.1, e. g. the `initial equation` section, are used to make the initilization of states user friendly. In hydraulics atmospheric pressure is used as reference pressure. Therefore a system at rest has pressure states of zero. In pneumatics the gas mass in a volume is used which is non-zero at atmospheric pressure. Therefore a number of calculations is needed to compute the gas mass in all lumped volumes which may include the calculation of the geometric volumes, e. g. of cylinders, before. As engineers usually think of pressure and not gas mass in a volume the input parameters for the initial conditions of the library models are pressures and the gas masses calculated by the library models.

The pneumatics library provides basic model classes for the modeling of cylinders - both standard cylinders with constant piston area and bellows which have a stroke dependent piston area - and motors, valves and restrictions, lumped volumes, lines and sensors. In total there are more than 80 models. Among them there are three models of pneumatic lines. Two describe the resistance by algebraic equations while the third one approximates the partial differential equations from the physical model by a set of ordinary differential equations. Laboratory experiments show an excellent correlation between measurement and simulation for the pressure drop and a good description of the dynamic behaviour, i. e. the frequency response.

For standard applications these classes cover all needed components. If, however, specially designed components are used these can be easily modelled by modifying library components. All relevant effects are available as submodels.

4 The Electronic Part

The control which is necessary for the pneumatic and mechanical parts can be modeled using Boolean algebra with the signal values 'true' and 'false' (Figure 6). A more detailed description is possible if multi-valued logic is used, e. g. with values for 'unknown', 'uninitialized'. Usually VHDL (Lehmann and Wunder, 1994) or Verilog- HDL (Palnitkar, 1996) are behavioral languages for digital logic for which powerful simulators exist, e. g. ModelSim (Modeltech, 2002). The VHDL language was used to verify the control unit design.

The control unit gets the input signals ON and OFF from outside to start and stop the machine. Further input signals are Bl (Br) for reaching the left (right) stop. The output signals both for moving to the left (Xl) and to the right (Xr) are stored within RS flipflops. If an output signal switches to false, the inverse flipflop output allows the other output to be switched to true. These changes are caused by both the Bl and Br signals. To connect the control unit with the pneumatic part a suitable signal conversion is necessary which is done by converter models.

For the unified modeling with Modelica the control unit is described at two levels, the Boolean level and the electrical level. For both of the levels a special small library 'Boole' and 'CMOS' has been developed.

'CMOS'-Library

Basing on the Modelica Standard Library CMOS transistors were combined to form the logic gates on the electronic level. The 'CMOS'-library contains the basic logic models Nand, Nor, Not, And, and a flipflop model RSFF. The following Modelica text shows the Nand gate model as an example:

```
model Nand "NAND"
import MEA = Modelica.Electrical.Analog
MEA.Sources.RampVoltage VDD;
MEA.Semiconductors.PMOS TP1, TP2;
MEA.Semiconductors.NMOS TN1, TN2;
MEA.Basic.Capacitor C4, C7;
MEA.Basic.Ground Gnd;
MEA.Basic.Ground Gnd1, Gnd2, Gnd3;
MEA.Interfaces.Pin inp1, inp2, out;
equation
connect(inp1, TN1.G); connect(TN2.G, inp2);
connect(TP2.G, TN2.G); connect(VDD.p, TP2.D);
connect(VDD.p, TP2.B); connect(Gnd1.p, VDD.n);
connect(TP1.D, VDD.p); connect(TP1.B, VDD.p);
connect(C4.n, Gnd2.p);  connect(TP1.S, C4.p);
connect(C4.p, out); connect(TN1.D, TP1.S);
connect(TN1.S, TN2.D); connect(C7.n, Gnd3.p);
connect(TN2.D, C7.p); connect(TN2.B, Gnd3.p);
connect(TN2.S, Gnd3.p); connect(TN1.B, Gnd.p);
connect(TP2.S, TP1.S); connect(TN1.G, TP1.G);
end Nand;
```

The MOS transistor models are used to be able to observe the electrical behavior in a great detail. Otherwise the number of variables becomes rather high. In practice this accurate level is not often necessary.

'Boole'-Libary

The basic logic gates and the flipflop as well were modeled using the Boolean signals 'true' and 'false' of Modelica (two-valued logic) according to (Tiller, 2001). Delay times are neglected. Only the flipflop needs a very small delay to avoid loops without delay. The following Modelica text shows the Nand gate of the 'Boole' library:

```
model Nand
import D = Boole.Interfaces;
extends D.DISO_wide;
D.LogicValueType out_immed(start=false);
equation
out_immed = not (in1 and in2);
out = pre(out_immed);
end Nand;
```

Due to the simplicity of 'Boole' the number of variables of the control unit model is much less than of the model based on 'CMOS'.

The 'Boole' libary is a very preliminary stage of the digital electronic library which is under development to become a part of the Modelica Standard Library. The digital electronic library follows essentially the IEEE 1164 standard (VHDL IEEE-Package).

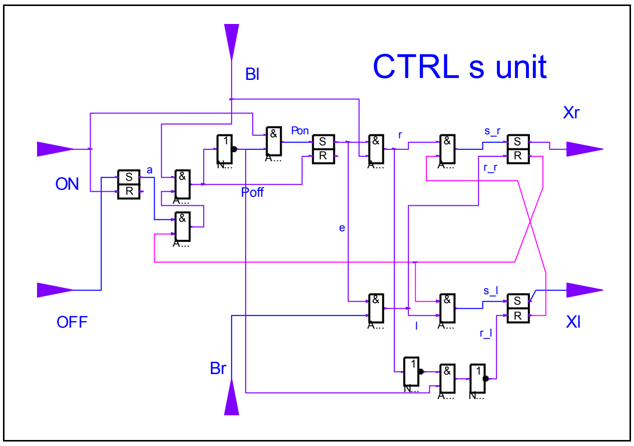

Figure 6 Control unit scheme

5 Results

With Dymola version 5 (Dymola, 2003) the model of the drive was composed graphically, analysed, translated into executable code, and simulated. The simulations started at the quiescent state (all voltages are zero, the pneumatic pressures are equal to the environment pressure) at time zero and finished after 2 seconds. In the following figures the behaviour of some variables is shown.

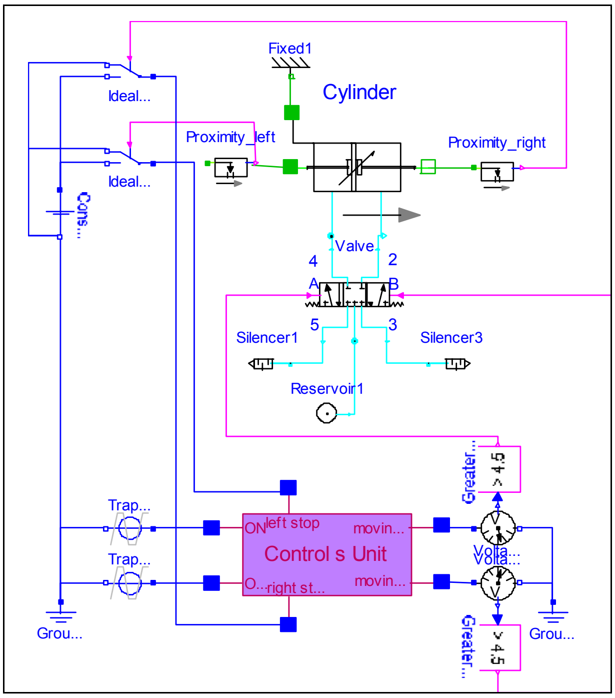

Figure 7 Object diagram of controlled drive

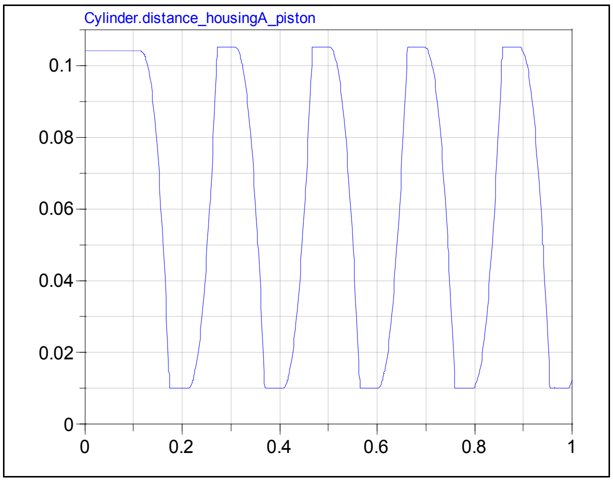

Figure 8 Piston position as function of time

To get a better feeling of the model the detailed subsystems “Pneumatic” and “CMOS” were replaced by much simpler models, “Integrator” and “Boole”. They had the same input-output signals, e. g. an input to drive to the right, i. e. increase the state of an integrator linearly with time. Using the simpler models the complexity of the model and the required CPU time can be considerably reduced.

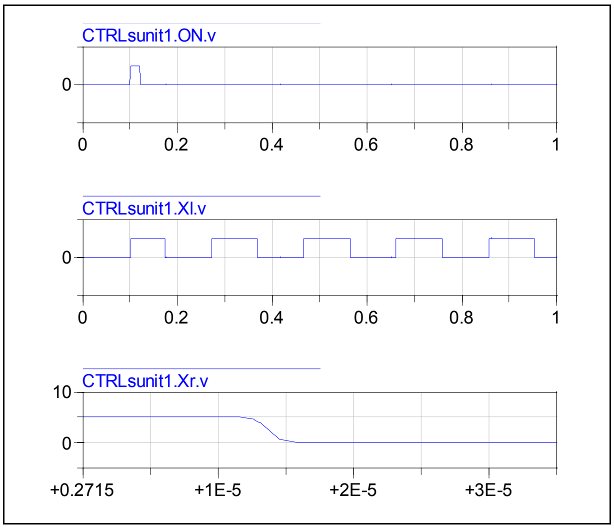

Figure 9 Start signal, command to go left, command to go right (zoomed)

Table 1 shows the simulation times and also that the multidomain model needs more computing time than the added times for Pneumatic/Boole and Integrator/CMOS. The "additional burden for multi domain" depends very much on the chosen tolerance for the DASSL integrator. In the best case, TOL = 1e-5, the complete model needs less than double the time than the model Integrator/Boole (Table 3). This effect has also been observed with a previous multidomain system (Clauß and Beater 2002).

Table 2 gives the eigenvalues of the complete system which can be uniquely associated with the pneumatic or electronic subsystem, respectively. The pneumatic system adds 14 states but the additional eigenvalues lie almost within those of the CMOS model.

6 Conclusion

A rather complicated multidomain example could be modeled and simulated in an easy way without simulator coupling. Depending on the task each of the two main subsystems was modeled more or less detailed. As a consequence the CPU times varied considerably but even for the most detailed model the "burden of multidomain" was acceptable.

However, to get more insight in the multidomain simulation with regard to both modeling and numerical aspects much more complex examples are desible.

Table 1 Comparison of model parameters and simulation times (Dymola 5.1, Windows 2000, 2.6 GHz)

Typ	Equations	States	CPU time
Pneumatic-Boole	262	14	0.984 s
Pneumatic-CMOS	1027	57	78.3 s
Integrator-Boole	176	1	0.031 s
Integrator-CMOS	903	44	15.5s

Table 2 Eigenvalues of the system CMOS/Pneumatics

CMOS	Pneumatic
-3.1474e+006	
	-1.0000e+005 two times
	-4.8898e+004
	-1.3571e+004 two times
	-4.3576e+002
	-3.0000e+002
	-2.5353e+001
	-8.4914e+000
-2.9867e-001 seven times	
-2.5719e-001 eight times	
-1.8650e-001	
-1.4450e-001 eleven times	
-9.2833e-002 seven times	
-7.3814e-002 eight times	
	-1.4388e-012
	0 two times
	1.2342e-013

Table 3 CPU times as function of tolerance

TOL	Pneumatic CMOS	Integrator CMOS
1e-4	78.3	15.5
1e-5	39.0	20.4
1e-6	49.4	26.0
1e-7	61.0	31.7
1e-8	74.3	38.9
1e-10	104	54.6

References

Beater, P. (1999) Entwurf hydraulischer Maschinen - Modellbildung, Stabilitätsanalyse und Simulation hydrostatischer Antriebe und Steuerungen. Berlin, Heidelberg, NewYork, Springer-Verlag

Clauss, C., Beater, P. (2002) Multidomain Systems: Electronic, Hydraulic, and Mechanical Subsystems of an Universal Testing Machine Modeled with Modelica. Proceedings of the 2nd International Modelica Conference, DLR München. pp 25 - 30

Clauß, Chr., Leitner, Th.,Schneider, A., Schwarz, P. (2000) Modelling of electronic circuits with Modelica. Lund, Modelica Workshop 2000, pp 3-11

Dymola. 2003. www.Dynasim.se

Herpy, M. (1976) Analoge integrierte Schaltungen. Akadémiai Kiadó. Budapest

HyLib (2002) Library of hydraulic components. www.HyLib.com

Johnson, B., Quarles, T., Newton, A.R., Pederson, D.O., Sangiovanni-Vincentelli, A. (1991) SPICE3 Version 3e, User’s Manual., Univ. of California, Berkeley, Ca., USA

Lehmann, G.; Wunder, B.; Selz, M. (1994) Schaltungsdesign mit VHDL. Francis Verlag.

Modelica (2002) Modelica Language Specification 2.0. www.Modelica.org

Modeltech (2002) ModelSim Reference Manual

Palnitkar, S. (1996) Verilog HDL: A Guide to Digital Design and Synthesis. SunSoft Press, Prentice Hall

PneuLib (2002) Library of Pneumatic Components. www.Pneulib.com

Tietze, U., Schenk, Ch. (1980) Halbleiter-Schaltungselektronik. Berlin, Heidelberg, New York, Springer-Verlag

Tiller, M. (2001) Introduction to Physical Modeling With Modelica. Kluwer Academic Publishers.

VHDL IEEEPackage: http://tech-www.informatik.uni-hamburg.de/vhdl/packages/ieee_1164