Defense Technical Information Center Compilation Part Notice

ADP012707

TITLE: Object-Oriented Dynamic Bayesian Network-Templates for Modelling Mechatronic Systems

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: Thirteenth International Workshop on Principles of Diagnosis [DX-2002]

To order the complete compilation report, use: ADA405380

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:
ADP012686 thru ADP012711

Object-Oriented Dynamic Bayesian Network-Templates for Modelling Mechatronic Systems

Harald Renninger and Hermann von Hasseln

DaimlerChrysler AG Research and Technology (REM/E)
D-70546 Stuttgart, Germany
{harald.renninger, hermann.v.hasseln}@daimlerchrysler.com

Abstract

The object-oriented paradigma is a new but proven technology for modelling mechatronics, i.e. multidisciplinary modelling. For many reasons the object-oriented approach is very much desirable also for qualitative models in system design, diagnosis or verification. Bayesian networks are a very robust technology for qualitative probabilistic modelling. In this paper we present a first approach in using the Bayesian networks modelling technique with the quantitative object-oriented method. Analogous to Modelica, an object-oriented modelling language, we constructed a Bayesian network library for modelling hydraulic systems. These Bayesian networks are called Object Oriented Dynamic Bayes Nets (OODBNs). Our method is easily transferable to any other physical domain or logic. In this contribution our motivation and the construction steps are described. Simulation results for a sample hydraulic system are given.

Introduction

Future system architectures will be characterized by highly modular and reusable components, and by abstract description languages widely independent of implementation details. Typical components of system architectures are software and hardware (sub-)systems. On the Software side the object-oriented paradigma is by now (at least in industrial applications) the *de facto* description or modelling language standard, mostly represented by the Unified Modelling Language (UML). On the Hardware side, which is our focus here, we have mechatronic hardware components, the constituent parts of which are control logics and controlled physical or chemical systems. Modelling mechatronic systems challenges the engineer due to different physical domains. In order to reach the goal of a truly unified description of system architectures comprising Software and Hardware systems, the description or modelling languages of mechatronic systems have to be lifted to a similar abstract level as their Software counterparts.

Model based techniques play an important role in concurrent and future engineering processes. Models and simulations are a basis for system design and analysis, e.g. for geometric layout of hydraulic systems. On the other hand, model based control and model based diagnosis are state of the art.

Many different philosophies have been developed to support the modelling task. In the control engineering area tools like Matlab/Simulink [1] or MatrixX SystemBuild [2]

are widespread. For modelling mechanical systems ADAMS [3] or SIMPACK [4] are frequently used. For electronic systems PSPICE [5] is an appropriate tool. Other specific tools are used to solve modelling tasks in flow dynamics, thermal flow or chemical processes. Each of these programs are specially tailored for the specific domain.

A mechatronic system consists of a control logic, electronics and a controlled mechanical, hydraulic or any other physical or chemical system. The entire system is composed of subsystems of different domains. This shows the restriction of all classical modelling systems, since the control part can be easily described for example in Matlab/ Simulink, but it is nearly impossible to model an electrical subsystem. So, a method is needed for a multidisciplinary modelling.

Methods and tools, e.g. Omola, Dymola or Smile, have been developed which allow multidisciplinary modelling. Modelica [6], [7] is the latest step in this direction. It is a standardized object-oriented modelling language which is supported by the tool Dymola [8] for example.

Dymola/Modelica comes with libraries for different physical domains like electrics/electronics, mechanics, thermal flow or hydraulics, see Figure 1. It also contains a signal block and a Petri net library. A library consists of a set of templates for different physical or logical objects. The user can extend a library for example by inheritance or can create completely new libraries. A model is described by an object diagramm. Most tools contain a graphical interface with a simple drag and drop technique for the templates and interconnections at the object interfaces. The interconnections have the meaning of constraints. More precisely, two types of equations are generated when two physical objects are connected: a flow and a potential equation. With the definition of the flow and the potential variables, the *energy flow* in the interface is uniquely defined. This is valid for all lumped parameter systems. The great advantage is that the system can now be modelled by *local* behaviour and not by global analysis [9], which supports the general idea of modularity.

Qualitative Models and Bayesian Networks

Qualitative modelling offers many well-known advantages for system design, diagnosis or verification, see [14] for a very extensive survey of techniques and applications. Some of these advantages are:
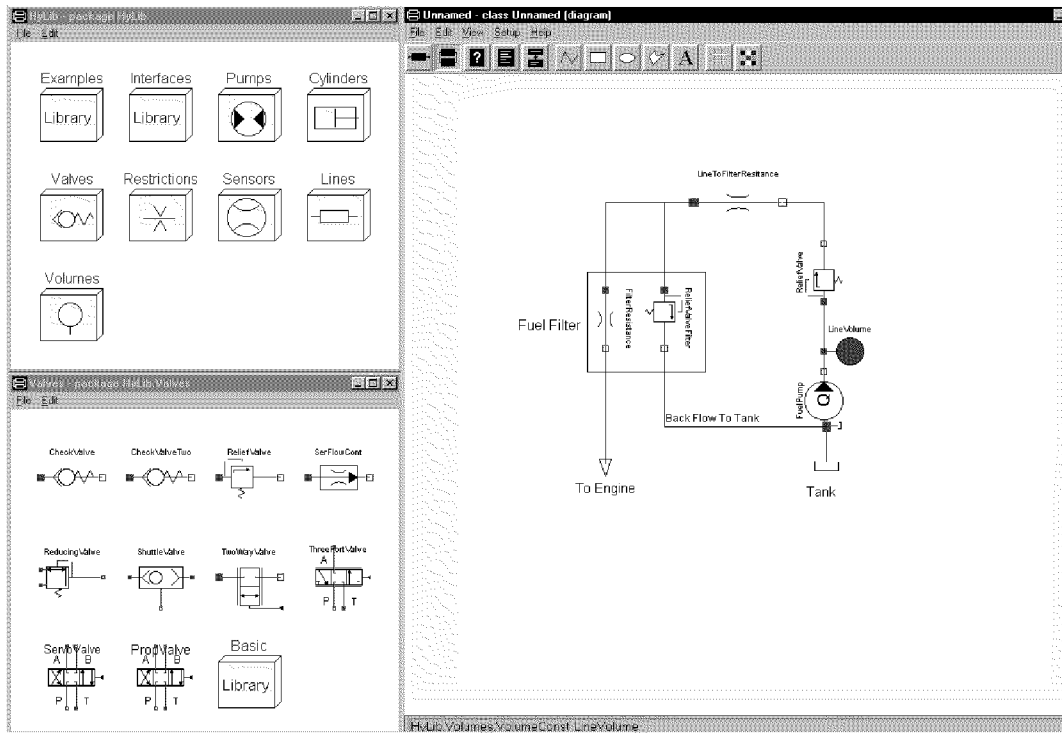
Figure 1: The hydraulics library and an object diagram in Dymola.

- handling of incomplete and imprecise knowledge,
- robustness,
- easy comparison of system alternatives, e.g. parameters variations,
- direct interpretation of simulation results,
- complexity.

Our vision is an object-oriented method using Bayesian networks for modelling physical systems, especially system dynamics. Bayesian networks are a well-suited method for handling imprecise knowledge in a consistent way. Efficient learning and adaption algorithms are known for Bayesian networks, which is a very interesting option for automatic model calibration. The definition of a Bayesian network *BN* is as follows: BN = {DAG, CPDs}, where DAG is a directed acyclic graph, consisting of nodes and directed edges or links, and CPDs are conditional probability distributions. The nodes in a Bayesian network represent propositional variables of interest (e.g., the temperature of a device). The links of a *BN* represent informational or causal dependencies among the variables. These dependencies are quantified by conditional probabilties (the CPDs) for each node given its parental nodes in the DAG. We do not cite the Bayesian networks fundamentals in this paper, but refer to the relevant literature, see [10] for some Bayesian networks basics or [11] for an excellent textbook.

Object-oriented Bayesian networks were introduced in [13], and are now supported by the newest version of the commercial Software tool *HUGIN* [12] for example.

Template construction

In this section we describe the conversion steps from Modelica to Bayesian network templates. The conversion will proceed in four major steps. First, given a dynamic component, the differential equations will be discretized in time using Euler's rule. Second, the equation part of a Modelica template will be reformulated with qualitative operators. Third, the qualitativ landmarks have to be chosen for each state variable and each parameter. Fourth, the resulting qualitative equations will be graphically programmed with Bayesian networks.

An fuel reservoir called "VolumeConst" will serve as an example. The icon used in the Modelica HyLib library [15] is shown in Figure 2. Note, that the component VolumeConst has one port (portA) and that the flow into the component has a positive sign . PortA can be viewed as a real physical flange with some pressure p and an oil flow q. The behavior of the component VolumeConst is described in Modelica by the equation block. Other definition blocks like the graphical, interfaces or parameter block are omitted.
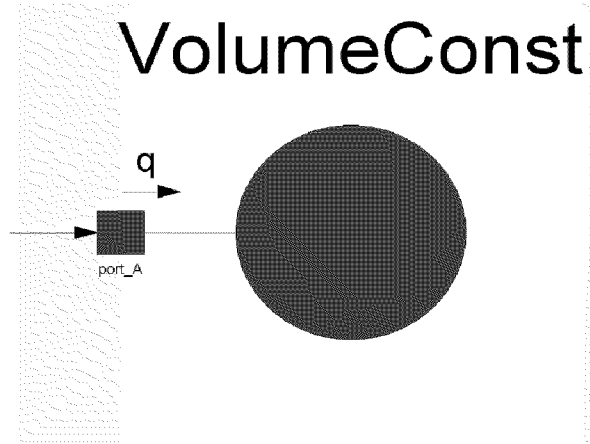
Figure 2: The Dymola representation for a fuel reservoir.

model VolumeConst
```
   graphical block
   interfaces block
   parameter block
```
equation
   der(portA.p) = beta/volume * portA.q;
end VolumeConst

The equation block consists of one differential equation with *beta* and *volume* being fixed parameters defining the effective bulk modulus of the liquid and the volume in square meters, respectively. After chosing a time step h the time discrete version is as follows:

model VolumeConstDiscr
   equation
   1\h (portA.p(t) − portA.p(t-h)) =
   (beta/volume * portA.q(t))
end VolumeConstDiscr

Next, qualitative operators are inserted.

model VolumeConstQual
   equation
   portA.p(t) ⊕ -portA.p(t-h) =
   const ⊗ portA.q(t)
end VolumeConstQual

Now we have to choose a quantity space, the "landmarks", for the variables and parameters. For clearness, we choose a three valued quantity space $x \in \{-, 0, +\}$ for all variables x. Some qualitative calculus has to be defined for the chosen quantity space. Qualitative addition ⊕ for the three valued quantity space can be defined straightforward [14] as in Table 1.

⊕	$x = -$	$x = 0$	$x = +$
$y = -$	$z = -$	$z = -$	$z = ?$
$y = 0$	$z = -$	$z = 0$	$z = +$
$y = +$	$z = ?$	$z = +$	$z = +$

Table 1: Qualitative addition defined.

The $z = ?$ entry marks the ambiguity of the result, when the ⊕ operator is applied on $x = -$ and $y = +$ or vice versa, respectively.

Now the Bayesian network template for VolumeConst can be constructed. The basic idea is to identify each qualitative variable with a Bayesian network node, the qualititative values with the states of this node, and the qualitative calculus with CPDs.

We give an example for the ⊕ operator applied on variables x and y. The principle Bayesian network is shown in Figure 3. The entries in the CPD table in Figure 3 are probabilities, where each column sums to 1. The $z = ?$ entries in the operator table can be represented by the colomns with the uniform distribution, i.e. 1/3 for each entry in this case.

Any other algebraic operation can also be reformulated as a Bayesian network fragment. In this way the complete template for VolumeConst is constructed. The result is shown in Figure 4. The port nodes, which correspond to the port variables in portA are marked with a rectangle. When the Bayesian network template is instantiated in a system model only the input and output nodes, i.e. the port nodes, are visible. Note that this is a dynamic Bayesian network, because node PA0 carries the state of the pressure at time slice $t-h$ and PA1 the state at time slice t. The difference is calculated in node dP01.
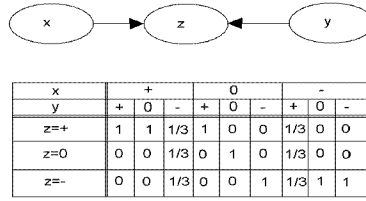
x | + | | | 0 | | | - | |
---|---|---|---|---|---|---|---|---|---
y | + | 0 | - | + | 0 | - | + | 0 | -
z=+ | 1 | 1 | 1/3 | 1 | 0 | 0 | 1/3 | 0 | 0
z=0 | 0 | 0 | 1/3 | 0 | 1 | 0 | 1/3 | 0 | 0
z=- | 0 | 0 | 1/3 | 0 | 0 | 1 | 1/3 | 1 | 1

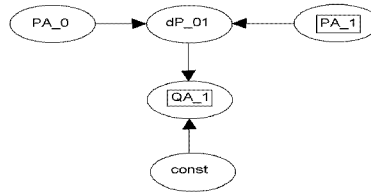**Figure 3:** Bayesian network fragment and the CPD for the $\oplus$ operator.

Figure 4: Bayesian network template for VolumeConst.

What is missing yet are the constraint templates, which serve as connectors between components. We need two different templates, one expressing that there is equal pressure at two connected ports, and a second one, expressing that the flows sum up to zero at a hydraulic node. We present these two templates with the CPDs in Figure 5 and Figure 6, respectively. For the pressure we assume three values: zero pressure (0), low pressure (+), high pressure (++). Note, that the arcs are directed to the "inner" constraint node, such that the resulting Bayesian network model is always acyclic. In
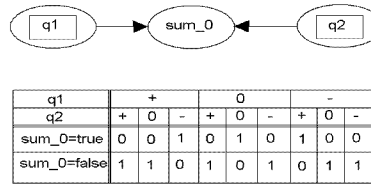

q1 | + | | | 0 | | | - | |
---|---|---|---|---|---|---|---|---|---
q2 | + | 0 | - | + | 0 | - | + | 0 | -
sum_0=true | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0
sum_0=false | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1

Figure 5: Bayesian network template ZeroSumFlows2 for the flow constraint.

p1 | 0 | | | + | | | ++ | |
---|---|---|---|---|---|---|---|---|---
p2 | 0 | + | ++ | 0 | + | ++ | 0 | + | ++
eq_p=true | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1
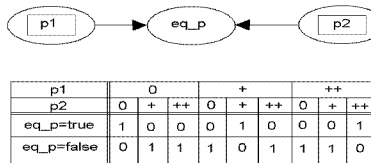eq_p=false | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0

Figure 6: Bayesian network template EqPressure2 for the pressure constraint.

Figure 5 and Figure 6 we show the simplest scenario that two components are connected in series. In Figure 1, on the right hand side, this is the case for the "ReliefValve" and the "LineToFilterResistance". In the Bayesian network for this tank system, the ReliefValve-template and the LineToFilter-Resistance-template will be connected via the flow constraint and the pressure constraint, see also Figure 7. Note that the table entries are hard 0/1 decisions. Before propagating the Bayesian network, the "true"-state of the sum_0 and the eq_p nodes must always be set evident. Doing this, the pressures on both sides are forced to be equal. The flow constraint then simply states, that the mass flow coming out of the first component equals the mass flow into the second component. In the general case, where more than two components meet, for example the "LineToFilterResistance", the "FilterResistance" and the "ReliefValveFilter", the flow constraint template must be assembled from the ⊕ -operation fragment, see Figure 3, and the flow constraint template of Figure 5. This new object is then called ZeroSumFlows3 and is shown in Figure 7.

Results

A basic library for constructing simple hydraulic circuits has been developed. It contains an ideal flow source, a reservoir, a hydraulic resistance, a tank, a relief valve, a real flow source, and the constraint templates. Differing from the pre-

viously discussed three state nodes, each dynamic variable here has five states. We used the object-oriented Bayesian network software Hugin. Currently, only discrete valued nodes are used. This is reasonable, because many mechatronic systems are hybrid or switching. Discrete valued nodes allow us to model arbitray dynamics, whereas continuous valued node models result in Kalman models, thus linear models.

We will shortly discuss the hydraulic library. The ideal flow source has two ports, namely A and B, or a "positive" and a "negative" port, with only one flow variable which can be controlled, i.e. set evident. A real flow source called Real-FuelPump is derived from this ideal flow source. Additionally it contains the volume model, which was described above. So the port B of RealFlowPump delivers a pump flow and a pump pressure. The tank model has only a flow variable at the ports A and B. It is dynamic, modelling the change of the fuel volume over time.

The relief valve has a switching behaviour in Modelica. Pressure and flow is specified at the ports. In Modelica the valve logic is modelled with a state machine. We modelled this valve logic with a Markov model, having the two states "open" and "closed". At last, the hydraulic resistance has two ports specifying pressure and flow. It models laminar flow, i.e. the pressure drop over a hydraulic line.We used this element also to model the resistance of the fuel filter.
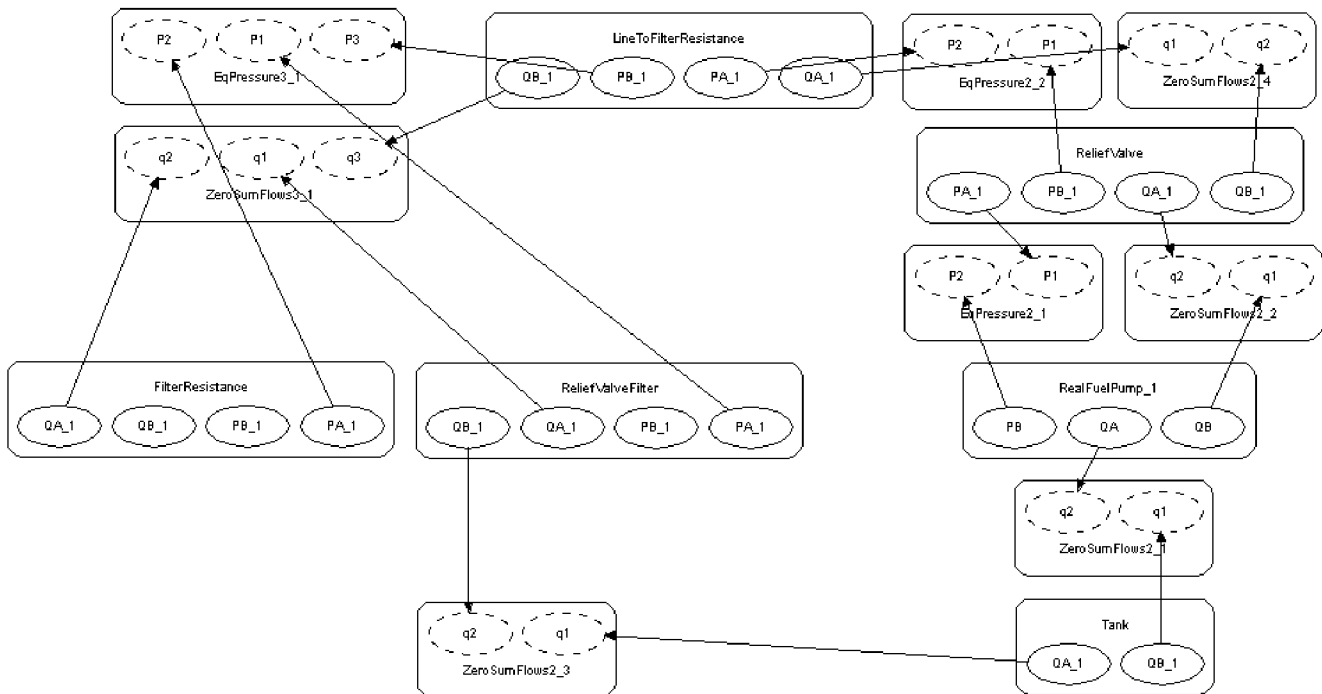

Figure 7: The Bayesian network tank system model.

We present a little hydraulic circuit, that is, a fictitious tank system. This system was first modelled for reference in Dymola/Modelica, see Figure 1 on the right hand side. Then we built this system using the dynamic Bayesian network templates. The Bayesian network tank system is shown in Figure 7. For the dynamic simulation, we set evident all constraint nodes and the pump flow. All other nodes are hidden, that is, they were calculated by propagation. The results for 100 time steps are shown in Figure 8, Figure 9, and Figure 10. These figures show the evolution of the probability distributions. The darker the colour bars are, the higher the probability. We added mean values for conveniance. The plots were produced using the Qualitative Modelling Toolbox for Matlab SIMULINK [16].
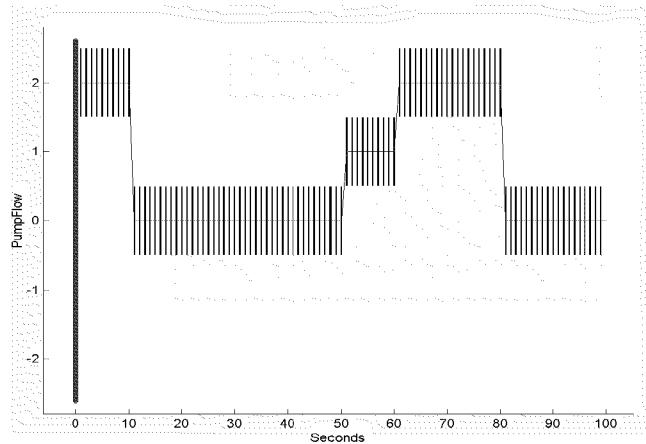

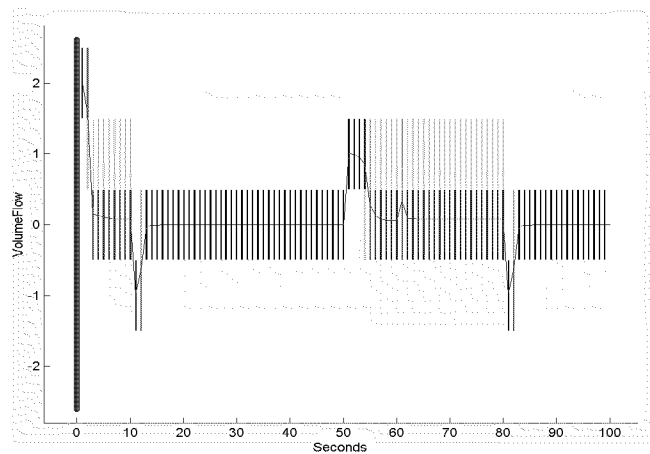**Figure 8:** The pump flow (evident node).

Figure 9: The flow into the fuel reservoir VolumeConst (inside the RealFuelPump template - hidden node).

Conclusion and future work

In this contribution we have motivated the need for intelligent modelling techniques. For system design, diagnosis or verification qualitative models are a very good choice. We favor the Bayesian network technology due to their robustness, intuitivity and practicability.

We seeked a qualitative modelling technique for mechatronic systems, i.e. for dynamic, multidomain systems. The object-oriented physical modelling technique gave us the hint for the construction of our OODBNs. The simulation results encourage us to proceed in this direction. Recent success has been made to select the states (the "landmarks") upon measurements or quantitative simulations, using a simple heuristic from system identification. Furthermore, learning respectively adapting the CPDs using HUGINs adaption API was very promising.

References

[1] SIMULINK. Homepage: http://www.Mathworks.com/
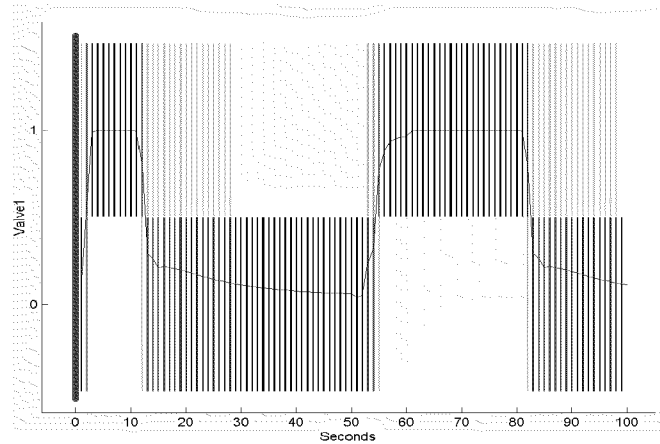
[2] SYSTEMBUILD. Homepage: http://www.isi.com/

Figure 10: The ReliefValve behavior (hidden node), where '0' means closed, and '1' means open.

Products/MATRIXx/

[3] ADAMS. Homepage: http://www.adams.com/

[4] SIMPACK. Homepage: http://www.simpack.de/

[5] PSPICE. Homepage: http://www.microsim.som/

[6] Elmqvist, H.; Mattsson, S.E.; Otter, M.1999. Modelica - A Language for Physical System Modeling, Visualization and Interaction. IEEE Symposium on Computer-Aided System Design, CACSD'99, Hawaii, August 22-27, 1999.

[7] Modelica. Homepage http://www.modelica.org

[8] Dymola. Homepage http://www.dynasim.se

[9] Otter, M. et al. 1999. Objektorientierte Modellierung Physikalischer Systeme, Teil 1. at - Automatisierung-stechnik 47, pp.A1-A4. Continued in the following issues of the at.

[10] Jensen, F. 1996. Bayesian networks basics. AISB Quarterly 94, pp. 9-22.

[11] Cowell, R.G., Dawid, A.P., Lauritzen, S.L., Spiegelhalter, D.J. 1999. Probabilistic Networks and Expert Systems. Statistics for Engineering and Information Science. Jordan, Lauritzen, Lawless, Nair (eds.), Springer-Verlag New York Inc. ISBN 0-387-98767-3.

[12] Hugin. Homepage: http://www.hugin.com/

[13] Koller, D., Pfeffer, A. 1997. Object-oriented bayesian networks, Proceedings of the thirteenth conference on Uncertainty in Artificial Intelligence, UAI 97.

[14] Dague, P.et al. 1995. Qualitative Reasoning: A Survey of Techniques and Applications.AICOM Vol. 8, Nrs. 3/4, Sept./Dec. pp. 119--192.

[15] Beater, P. 1998. Object-oriented Modeling and Simulation of Hydraulic Drives. Simulation News Europe, SNE 22, pp. 11-13.

[16] G. Lichtenberg et.al. Qualitative Modelling Toolbox User's Guide. Release 5.2. Download: http://pcweb.rts.tu-harburg.de/inter/homepage.nsf