Aspects on simulation of switched bond graphs

Krister Edström

(edstrom@isy.liu.se)

Dept. of Electrical Engineering

Linköping University

S-581 83 Linköping, Sweden

Jan-Erik Strömberg

(janerik@s3.kth.se)

Dept. of Sensors, Signals and Systems

Royal Institute of Technology

S-100 44 Stockholm, Sweden

Jan Top

(j.l.top@ato.dlo.nl)

Agrotechnological Research Institute

ATO-DLO, P.O. Box 17

NL-6700 AA Wageningen, The Netherlands

Abstract

By mode-switching systems we mean *physical systems* characterized by transitions which are at least an order of magnitude faster than the overall time-scale. There are essentially two different ways to model such systems for simulation. One is to employ continuous models and stiff solvers. This approach is becoming more feasible as more sophisticated stiff solvers become available, but still suffers from poor performance for larger models.

An alternative approach is to employ hybrid models with instantaneous transitions between continuous mode-models. In this case solvers capable of detecting zero-crossings are required. The advantage of this approach is an improved simulation performance but the disadvantage is that one has to deal with more complex models. To handle the model complexity matter we have previously proposed switched bond graphs as a solution. Employing switched bond graphs, the problem of complex models is modified to the problem of translating the bond graph to a computational model. This is the problem addressed here.

transitions. By using modern solvers capable of determining the zero-crossing of transition conditions, such simulations have the potential of becoming highly efficient. The disadvantage though, is a more complex modelling process. On the other hand, once such a mode-switching model has been successfully constructed, that same model is much more clear from a conceptual point of view than its continuous counterpart. In other words, assumptions made by the modeller, often left implicit, now become much more clear from the model itself.

In this paper we will briefly review the difficulties associated with the second approach above. We will also review some potential solutions as provided by *switched bond graphs*; earlier presented in [18, 14, 17]. Finally, for a specific subset of switched bond graphs, we elaborate on the automated compilation of executable code to be linked with a specific numerical solver capable of detecting zero-crossings. Hence this paper contributes to the overall goal of providing more powerful tools for the efficient modelling and simulation of physical systems undergoing abrupt mode-transitions.

1 Introduction

By a mode-switching system we mean a *dynamic system* undergoing fast transitions between continuous modes of operation. To simulate such systems, there are essentially two fundamental approaches. One is to handle the fast transitions by means of stiff solvers. This approach is becoming more feasible as more powerful stiff solvers are developed [8, 9, 4]. The advantage of this approach is that the modelling process is not further complicated.

The other fundamental approach is to approximate the fast transitions by ideal instantaneous mode-

2 Idealisation in terms of instantaneous mode-transitions

Consider a system where fast (with respect to the overall time-scale) transitions are approximation by means of ideal instantaneous mode-transitions. This is modelled by a set of continuous dynamic mode-models and a set of discrete mode-transitions. Each mode-transition is further associated with a transition condition. A transition condition is a Boolean combination of relations over variables in continuous mode-models. A transition condition specifies when a specific mode-transition is to take place.

The most fundamental problem with this approach is that of complexity. The number of mode-models grows very quickly with the number of switching mechanisms in the system. Making sure that all mode-models have been covered, quickly becomes an overwhelming problem. Not the least, we also have the non-trivial problem on how to correctly specify the mode-transitions to ensure proper switching behaviour over all input trajectories. Furthermore, once a model has proved to be correct, the same problem occurs when the physical system itself or the model requirements changes.

Following the traditions of bond graphs [12, 11], we have proposed an approach [18, 13, 16, 20] to handle these problems by making a clear distinction between 'low-level' computational matters and 'high-level' physical concepts. In this view, mode-models and mode-transitions are considered as low-level computational details. By introducing proper high-level physical concepts, we hope to reduce the complexity of the modelling process. In other words, we aim at reducing the complexity as seen by the modeller, while maintaining the computational complexity, as dealt with by the numerical solver.

Adopting this approach, the original problem of the modelling process is shifted to that of properly translating a conceptual model into a computational one. Preliminary results obtained using switched bond graphs [20, 16, 13], and similar approaches using, *e.g.*, object-oriented modelling [7, 5, 1, 19, 2, 6], indicate that this process can be automated in many cases of practical interest.

3 Switched bond graphs

The conceptual modelling language proposed for mode-switching physical systems is referred to as switched bond graphs. One reason for the name is that the language heavily relies on the theory of 'classical' bond graphs as developed by numerous researchers [3] since their introduction in 1958 by H.M. Paynter. In a sense, switched bond graphs can be seen as an extension of classical bond graphs. For instance, switched bond graphs only add one more physical concept to the classical list of nine elements, namely the primitive *ideal switch* concept, denoted **Sw**.

On the other hand, the role of causality plays a significantly different role in switched bond graphs. For instance, while causality is a fixed property in classical bond graphs, it is a time-varying property in the switched counterpart. In addition, the notion of a *constitutive relation* associated with each primitive physical element, is extended by the notion of a *switch control structure* associated with the primitive switch concept. Both these important aspects of switched bond

Figure 1: Definition of the **Sw**-element.

graphs are clear from the following definition of the primitive ideal switch concept.

Definition 1 *The (primitive) switch element is a bond graph element, i.e. a node in the graph, defined by Fig. 1, where $\mu \in \{\mathcal{E}, \mathcal{F}\}$ is a Boolean **state variable** and α an arbitrary label; both associated with the switch. State $\mu = \mathcal{F}$ ($\mu = \mathcal{E}$) is referred to as the *flow* (*effort*) state.*

□

The state μ of the **Sw**-element is governed by a *switch control structure* (SCS) associated with each **Sw**-element. The SCS is, in its simplest case, a state-transition system with two states only. This particular instance of SCS is referred to as a *switch transition system* (STS).

As for all other primitive elements of the bond graph language, the switch element is used in combination with the other elements to form models of mechanisms found in physical systems. For instance, to model the electrical mechanism illustrated in Fig. 2, we employ rules from classical bond graph theory to end up with

Figure 2: A simple electric mechanism.

the switched bond graph in Fig. 3. Switch **Sw**₁ is different from **Sw**₂ in the sense that it is not only controlled by the external signal u_1 , but also by e_1 , as will be clarified later.

Figure 3: The switched bond graph for the electric mechanism above.

To simulate a system by means of a switched bond graph we will have to translate it to a form acceptable for a numerical solver. Once again, following the traditions of classical bond graphs, this translation is divided into two steps. First the bond graph is translated to a solver-independent mathematical representation, and secondly, this intermediate representation is translated into solver-specific executable code. For strictly continuous systems and classical bond graphs, the intermediate form is a differential (and algebraic) equation (DAE) system.

In the classical bond graph literature, the DAE form itself as well as the process of automatically deriving the DAE from a bond graph, are both thoroughly studied [21]. However, for switched bond graphs there is neither a corresponding form, which is generally accepted, nor a good understanding of the automated derivation of such a form. In what follows we will therefore present an intermediate form and also look into some aspects of the derivation of the same from switched bond graphs. The final step to produce the executable solver-specific code, will then be fairly straightforward, provided the intermediate form is chosen carefully enough.

4 Mode transition systems

It is clear that the intermediate solver-independent structure, underlying a switched bond graph, must be a hybrid system in the sense that it combines continuous dynamics with discrete mode-transitions. More precisely, the formal structure proposed here, consists of the following sets of objects:

- $\mathcal{L}$ is a non-empty set of mode labels
- Z is a non-empty set of real-valued model variables (power and energy variables)

- $U \subseteq Z$ is a non-empty set of real valued (independent) input variables
- $\mathcal{M}$ is a (possibly empty) set of continuous mode-models over the variables in Z , such that variables in U appear only as independent variables
- $\mathcal{G}$ is a non-empty set of binary decisions over the variables in Z
- $\mathcal{A}$ is a (possibly empty) set of initialisation rules of the form $x := \phi(z_1, z_2, \dots)$ where $x \in Z - U$, $z_i \in Z$ and $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ is any (static) map.

From these sets we then form a composite *hybrid model structure*

$$Q = \langle \mathcal{L}, Z, U, \mathcal{M}, \mathcal{G}, \mathcal{A} \rangle$$

The hybrid model structure can be viewed as a 'library' containing the necessary components for describing the hybrid system. In order to describe an instance of a hybrid system we need to combine the elements in the 'library' in a suitable manner. This leads us to the definition of a *mode transition system* (MTS).

Definition 2 Let Q be a properly defined hybrid model structure. Then a **mode transition system** (MTS) is a three-tuple $\langle M, T, Q \rangle$ where M is a non-empty set of modes m and T a non-empty set of transitions τ .

A mode m is a pair (l, λ) where l is a discrete state, i.e., an assignment of the state variables in $\mathcal{L}$ and $\lambda \in \mathcal{M}$ is a continuous mode-model.

A transition τ is a three-tuple $\langle e, g, a \rangle$ where $e \in M \times M$ is an ordered pair (m, m') of modes $m, m' \in M$, $g \in \mathcal{G}$ is a Boolean transition condition and $a \in \mathcal{A}$ is an initialisation rule.

In addition we require that there is only one transition τ for each distinct pair (m, m') of modes and that there are no self-loops, i.e., reflexive transitions of the type $\langle (m, m), g, a \rangle$.

□

The MTS is hence the intermediate solver-independent form proposed for mode-switching physical systems. In order to discuss the translation of a switched bond graph to an MTS we now need a formal specification of individual switches. To simplify and clarify the translation problem, we will restrict ourselves to switched bond graphs with only one-port switches that have 'simple' transition conditions of a particular kind. This particular sub-class of switches are called *primitive switches*.

We will also restrict ourselves to a procedure where no simplification of the generated MTS is done. Our last

assumption is that there does not exist any causal conflicts in any mode. This is the most restrictive assumption since it is very common to have causal conflicts in some modes, and it is made to avoid a too detailed discussion.

For the specification of individual primitive switches, from which the composed MTS is to be derived, we now formalise the concept of a *primitive switch transition system*.

Definition 3 A **primitive switch transition system (PSTS)** is a three-tuple $\langle M, T, Q \rangle$ where

- Q is a hybrid model structure $\langle \mathcal{L}, Z, U, \mathcal{M}, \mathcal{G} \rangle$ where
 - $\mathcal{L} = \{l_{\mathcal{E}}, l_{\mathcal{F}}\}$
 - U is a set of external real-valued (independent) input variables
 - $Z = \{e, f\} \cup U$, where the variables e and f are the port variables of the primitive switch itself
 - $\mathcal{M} = \{\{e := 0\}, \{f := 0\}\}$
 - $\mathcal{G} = \{g_{\mathcal{EF}}(z), g_{\mathcal{FE}}(z')\}$, where

$$z = (z_1, \dots, z_k)^T \text{ and } z' = (z'_1, \dots, z'_{k'})^T$$
 are vectors composed by the variables $z_1, \dots, z_k, z'_1, \dots, z'_{k'} \in Z$
- $M = \{m_{\mathcal{E}}, m_{\mathcal{F}}\}$ where

$$m_{\mathcal{E}} = (l_{\mathcal{E}}, \{e := 0\})$$

$$m_{\mathcal{F}} = (l_{\mathcal{F}}, \{f := 0\})$$
- $T = \{\tau_{\mathcal{EF}}, \tau_{\mathcal{FE}}\}$ where

$$\tau_{\mathcal{EF}} = \langle (m_{\mathcal{E}}, m_{\mathcal{F}}), g_{\mathcal{EF}}(z) \rangle$$

$$\tau_{\mathcal{FE}} = \langle (m_{\mathcal{F}}, m_{\mathcal{E}}), g_{\mathcal{FE}}(z') \rangle$$

□

Note that the PSTS is in fact an MTS, and furthermore, the most simple MTS (with more than one mode) which one can derive from a switched bond graph.

To exemplify the PSTS, reconsider the example presented in Fig. 2 and Fig. 3. Let the transition conditions for $\mathbf{Sw}_1$ be $\{g_{\mathcal{EF}} = u_1 > 0, g_{\mathcal{FE}} = e_1 < 0 \wedge u_1 < 0\}$ and for $\mathbf{Sw}_2$ $\{g_{\mathcal{EF}} = u_2 > 0, g_{\mathcal{FE}} = u_2 < 0\}$, where u_1 and u_2 are external control variables. The PSTS for $\mathbf{Sw}_1$ is hence a three-tuple $\langle M_1, T_1, Q_1 \rangle$ where

- Q_1 is a hybrid model structure $\langle \mathcal{L}_1, Z_1, U_1, \mathcal{M}_1, \mathcal{G}_1 \rangle$ where

- $\mathcal{L}_1 = \{l_{\mathcal{E}}, l_{\mathcal{F}}\}$
- $U_1 = u_1$
- $Z_1 = \{e_1, f_1, u_1\}$
- $\mathcal{M}_1 = \{\{e_1 := 0\}, \{f_1 := 0\}\}$
- $\mathcal{G}_1 = \{g_{\mathcal{EF}} = u_1 > 0, g_{\mathcal{FE}} = e_1 < 0 \wedge u_1 < 0\}$

- $M_1 = \{m_{\mathcal{E}}, m_{\mathcal{F}}\}$ where

$$m_{\mathcal{E}} = (l_{\mathcal{E}}, \{e_1 := 0\})$$

$$m_{\mathcal{F}} = (l_{\mathcal{F}}, \{f_1 := 0\})$$

- $T_1 = \{\tau_{\mathcal{EF}}, \tau_{\mathcal{FE}}\}$ where

$$\tau_{\mathcal{EF}} = \langle (m_{\mathcal{E}}, m_{\mathcal{F}}), g_{\mathcal{EF}}(e_1, f_1, u_1) \rangle$$

$$\tau_{\mathcal{FE}} = \langle (m_{\mathcal{F}}, m_{\mathcal{E}}), g_{\mathcal{FE}}(e_1, f_1, u_1) \rangle$$

The PSTS of $\mathbf{Sw}_2$ will essentially be the same as the PSTS for $\mathbf{Sw}_1$. Simply remove the condition $e_1 < 0$ and change the indices from 1 to 2.

5 Composition of MTS

The interesting problem is how to compose an MTS from individual STS's in such a way that the semantics of the switched bond graph is preserved. The problem is interesting partly because there is no single answer to the question. One of the first questions needed to be answered is whether two or more switch elements are allowed (or are able) to switch at the same time. As for every model, the answer to such a question depends on the purpose of the model, as well as on assumptions made by the modeller. The answer also depends on whether memory or computational speed is important.

The important thing is that the assumptions about the simultaneous switching is *not* encoded in the switched bond graph itself. Following the traditions of classical bond graphs, this type of 'computational' issue has been postponed to the composition phase of the modelling process. Therefore we need to provide two different composition operators depending on the preferences of the modeller. Here we will only present one of these two operators, namely the *interleaved composition operator*. The operator corresponds to the case in which only one switch is allowed to switch at a time. This will increase the execution time, but decrease the size of the generated code for the model. The motivation for this operator, the problems associated with it as well as the alternative composition operator can be found in [16, 13].

Definition 4 Let $S_1 = \langle M_1, T_1, Q_1 \rangle$ and $S_2 = \langle M_2, T_2, Q_2 \rangle$ be two MTS. The **interleaved composition** between the two systems S_1 and S_2 , denoted $S_1 | i | S_2$, is an MTS, $S = \langle M, T, Q \rangle$ where:

- Q is a hybrid model structure $\langle \mathcal{L}, Z, U, \mathcal{M}, \mathcal{G} \rangle$ where

$$\begin{aligned}
- \mathcal{L} &= \mathcal{L}_1 \times \mathcal{L}_2 \\
- Z &= Z_1 \cup Z_2 \\
- U &= U_1 \cup U_2 \\
- \mathcal{M} &= \{ \lambda_{12} \mid \lambda_{12} = \lambda_1 \cup \lambda_2, \\
&\quad \lambda_1 \in \mathcal{M}_1 \text{ and } \lambda_2 \in \mathcal{M}_2 \} \\
- \mathcal{G} &= \mathcal{G}_1 \cup \mathcal{G}_2
\end{aligned}$$

- $M = \{ m_{12} \mid m_{12} = ((l_1, l_2), \lambda_1 \cup \lambda_2), \\ (l_1, \lambda_1) \in M_1 \text{ and } (l_2, \lambda_2) \in M_2 \}$

- $T = T'_1 \cup T'_2$ where

$$\begin{aligned}
&\langle ((l_1, l_2), \lambda_1 \cup \lambda_2), ((l'_1, l_2), \lambda'_1 \cup \lambda_2), g_1 \rangle \in T'_1 \text{ iff} \\
&\quad \langle (l_1, \lambda_1), (l'_1, \lambda'_1), g_1 \rangle \in T_1 \text{ and } (l_2, \lambda_2) \in M_2 \\
&\langle ((l_1, l_2), \lambda_1 \cup \lambda_2), ((l_1, l'_2), \lambda_1 \cup \lambda'_2), g_2 \rangle \in T'_2 \text{ iff} \\
&\quad \langle (l_2, \lambda_2), (l'_2, \lambda'_2), g_2 \rangle \in T_2 \text{ and } (l_1, \lambda_1) \in M_1
\end{aligned}$$

□

Since the PSTS is a special case of MTS, the composition operator defined holds for PSTS as well. From the construction of the interleaved composition operator it also follows that the composition of two arbitrary MTS becomes an MTS.

6 Simulation of switched bond graphs

Once the overall MTS for a given switched bond graph has been generated, the generation of executable solver-specific code is fairly straightforward. Typically the input to a solver, capable of detecting zero-crossings, is a set of equations defining the continuous dynamics in a given mode, and a set of equations defining the Boolean condition for exiting the given mode. These two data structures can now be directly derived from the MTS.

From a practical point of view, however, this approach may be problematic due to the exponential growth in complexity. Therefore the practical solution must typically be a modification of the ideal approach outlined above. Nevertheless, the MTS still plays an important role in theoretical studies and formal analysis of hybrid systems [10, 15].

Returning to our example circuit in Fig. 2 and employing the strategy outlined above, we end up in the MTS $S = \langle M, T, Q \rangle$, where

- Q is the hybrid model structure $\langle \mathcal{L}, Z, U, \mathcal{M}, \mathcal{G} \rangle$ where

$$\begin{aligned}
- \mathcal{L} &= \{ l_{\mathcal{E}\mathcal{E}}, l_{\mathcal{E}\mathcal{F}}, l_{\mathcal{F}\mathcal{E}}, l_{\mathcal{F}\mathcal{F}} \} \\
- U &= \{ u_1, u_2 \} \\
- Z &= \{ e_1, f_1, u_1, e_2, f_2, u_2 \} \\
- \mathcal{M} &= \{ \{ e_1 := 0, e_2 := 0 \}, \\
&\quad \{ e_1 := 0, f_2 := 0 \}, \\
&\quad \{ f_1 := 0, e_2 := 0 \}, \\
&\quad \{ f_1 := 0, f_2 := 0 \} \} \\
- \mathcal{G} &= \{ g_{\mathcal{E}\mathcal{E}|\mathcal{E}\mathcal{F}} = u_2 > 0, g_{\mathcal{E}\mathcal{E}|\mathcal{F}\mathcal{E}} = u_1 > 0, \\
&\quad g_{\mathcal{E}\mathcal{F}|\mathcal{E}\mathcal{E}} = u_2 < 0, g_{\mathcal{E}\mathcal{F}|\mathcal{F}\mathcal{F}} = u_1 > 0, \\
&\quad g_{\mathcal{F}\mathcal{E}|\mathcal{F}\mathcal{F}} = u_2 > 0, \\
&\quad g_{\mathcal{F}\mathcal{E}|\mathcal{E}\mathcal{E}} = e < 0 \wedge u_1 < 0, \\
&\quad g_{\mathcal{F}\mathcal{F}|\mathcal{F}\mathcal{E}} = u_2 < 0, \\
&\quad g_{\mathcal{F}\mathcal{F}|\mathcal{E}\mathcal{F}} = e < 0 \wedge u_1 < 0 \}
\end{aligned}$$

- $M = \{ m_{\mathcal{E}}, m_{\mathcal{F}} \}$ where

$$\begin{aligned}
m_{\mathcal{E}\mathcal{E}} &= (l_{\mathcal{E}\mathcal{E}}, \{ e_1 := 0, e_2 := 0 \}) \\
m_{\mathcal{E}\mathcal{F}} &= (l_{\mathcal{E}\mathcal{F}}, \{ e_1 := 0, f_2 := 0 \}) \\
m_{\mathcal{F}\mathcal{E}} &= (l_{\mathcal{F}\mathcal{E}}, \{ f_1 := 0, e_2 := 0 \}) \\
m_{\mathcal{F}\mathcal{F}} &= (l_{\mathcal{F}\mathcal{F}}, \{ f_1 := 0, f_2 := 0 \})
\end{aligned}$$

- $T = \{ \tau_{\mathcal{E}\mathcal{F}}, \tau_{\mathcal{F}\mathcal{E}} \}$ where

$$\begin{aligned}
\tau_{\mathcal{E}\mathcal{E}|\mathcal{E}\mathcal{F}} &= \langle (m_{\mathcal{E}\mathcal{E}}, m_{\mathcal{E}\mathcal{F}}), g_{\mathcal{E}\mathcal{E}|\mathcal{E}\mathcal{F}} \rangle \\
\tau_{\mathcal{E}\mathcal{E}|\mathcal{F}\mathcal{E}} &= \langle (m_{\mathcal{E}\mathcal{E}}, m_{\mathcal{F}\mathcal{E}}), g_{\mathcal{E}\mathcal{E}|\mathcal{F}\mathcal{E}} \rangle \\
\tau_{\mathcal{E}\mathcal{F}|\mathcal{E}\mathcal{E}} &= \langle (m_{\mathcal{E}\mathcal{F}}, m_{\mathcal{E}\mathcal{E}}), g_{\mathcal{E}\mathcal{F}|\mathcal{E}\mathcal{E}} \rangle \\
\tau_{\mathcal{E}\mathcal{F}|\mathcal{F}\mathcal{F}} &= \langle (m_{\mathcal{E}\mathcal{F}}, m_{\mathcal{F}\mathcal{F}}), g_{\mathcal{E}\mathcal{F}|\mathcal{F}\mathcal{F}} \rangle \\
\tau_{\mathcal{F}\mathcal{E}|\mathcal{F}\mathcal{F}} &= \langle (m_{\mathcal{F}\mathcal{E}}, m_{\mathcal{F}\mathcal{F}}), g_{\mathcal{F}\mathcal{E}|\mathcal{F}\mathcal{F}} \rangle \\
\tau_{\mathcal{F}\mathcal{E}|\mathcal{E}\mathcal{E}} &= \langle (m_{\mathcal{F}\mathcal{E}}, m_{\mathcal{E}\mathcal{E}}), g_{\mathcal{F}\mathcal{E}|\mathcal{E}\mathcal{E}} \rangle \\
\tau_{\mathcal{F}\mathcal{F}|\mathcal{F}\mathcal{E}} &= \langle (m_{\mathcal{F}\mathcal{F}}, m_{\mathcal{F}\mathcal{E}}), g_{\mathcal{F}\mathcal{F}|\mathcal{F}\mathcal{E}} \rangle \\
\tau_{\mathcal{F}\mathcal{F}|\mathcal{E}\mathcal{F}} &= \langle (m_{\mathcal{F}\mathcal{F}}, m_{\mathcal{E}\mathcal{F}}), g_{\mathcal{F}\mathcal{F}|\mathcal{E}\mathcal{F}} \rangle
\end{aligned}$$

Note that this MTS does not contain any explicit reference to mode-models. Rather the reference is implicit in the $\mathcal{L}$ and $\mathcal{M}$ structures. To derive the mode-models we simply substitute the **Sw**-elements by appropriate zero-sources; see Fig. 4.

Figure 4: Simplified mode-models, *i.e.*, classical bond graphs. Left: mode $m_{\mathcal{E}\mathcal{E}}$. Right: mode $m_{\mathcal{E}\mathcal{F}}$.

7 Summary

We have outlined two fundamentally different approaches to model and simulate mode-switching physical systems. Adopting the approach of idealising fast

transitions by instantaneous mode-transitions, we provide a conceptual modelling language for handling the complexity. The switched bond graph language presented is an extension to classical bond graph theory. Therefore the language inherits a solid basis of theory and practical modelling experience.

In the paper we present the proposed formal framework to represent the hybrid system underlying a switched bond graph. This system plays the same solver-independent role as the DAE system of continuous systems. We also present the algorithmic derivation of the proposed mode transition system from a switched bond graph. Finally we discuss some practical matters concerning the generation and execution of solver-specific code.

References

- [1] M. Andersson. *Object-Oriented Modeling and Simulation of Hybrid Systems*. PhD thesis, Lund Institute of Technology, 1995.
- [2] G.M. Asher and V. Eslamdoost. A novel causality changing method for the bond graph modelling of variable topology electronic switching systems. In *Proc. of IMACS Symposium*, pages 371–376, Lille, 1991.
- [3] P.C. Breedveld, R.C. Rosenberg, and T. Zhou. Bibliography of bond graph theory and application. *J. of the Franklin Institute*, 5/6(328):1067–1109, 1991.
- [4] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.
- [5] F.E. Cellier, M. Otter, and H. Elmqvist. Bond graph modeling of variable structure systems. In *Second International Conference on Bond Graph Modeling and Simulation (ICBGM'95)*, New Orleans, LA, 1995.
- [6] G. Dauphin-Tanguy, C. Sueur, and C. Rombaut. Bond-graph approach of commutating phenomena. In *Proc. of Advanced Info. Processing in Automatic Control*, pages 339–343, Nancy, 1989. IFAC.
- [7] H. Elmqvist, F.E. Cellier, and M. Otter. Object-oriented modelling of hybrid systems. In *the European Simulation Symp (ESS '93)*, Delft, 1993.
- [8] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*. Springer-Verlag, 1987.
- [9] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer-Verlag, 1991.
- [10] T. Henzinger et al., editors. *Proc. of the DIMACS International Workshop on Verification and Control of Hybrid Systems*. Springer Verlag, 1996.
- [11] D.C. Karnopp, R.C. Rosenberg, and D. Margolis. *System dynamics - A unified approach (2nd edition)*. John Wiley & Sons, New York, 1990.
- [12] H.M. Paynter. *Analysis and design of engineering systems*. MIT Press, Cambridge, M.A., 1961.
- [13] U. Söderman. *Conceptual modelling of mode switching physical systems*. PhD thesis, Linköping University, Linköping, 1995. Dissertation no. 375.
- [14] U. Söderman, J.L. Top, and J.E. Strömberg. The conceptual side of mode switching. In *Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics*, volume 1, pages 245–250, Le Touquet, 1993. SMC.
- [15] J.E. Strömberg, S. Nadjm-Tehrani, and J.L. Top. Switched Bond Graphs as Front-end to Formal Verification of Hybrid Systems. In Henzinger et al. [10].
- [16] J.E. Strömberg. *A mode switching modelling philosophy*. PhD thesis, Linköping University, Linköping, 1994. Dissertation no. 353.
- [17] J.E. Strömberg, J.L. Top, and U. Söderman. Modelling mode switching dynamic systems. In *Proc. Second European Control Conference (ECC '93)*, volume 2, pages 848–853, 1993.
- [18] J.E. Strömberg, J.L. Top, and U. Söderman. Variable causality in bond graphs caused by discrete effects. In *Proc. First Int. Conf. on Bond Graph Modeling (ICBGM '93)*, number 2 in SCS Simulation Series, volume 25, pages 115–119, San Diego, 1993.
- [19] J.H. Taylor and D. Kebede. Modeling and simulation of hybrid systems in matlab. In *Proc. of IFAC'96*, pages 275–280. IFAC, 1996.
- [20] J.L. Top. *Conceptual modelling of physical systems*. PhD thesis, University of Twente, Enschede.
- [21] J. van Dijk. *On the role of bond graph causality in modelling mechatronic systems*. PhD thesis, University of Twente, Enschede, 1994.