SIMULATION OF DISCONTINUITIES IN PHYSICAL SYSTEM MODELS BASED ON CONSERVATION PRINCIPLES

Pieter J. Mosterman[†*] **Gautam Biswas**[‡] **Martin Otter**[†]

[†]DLR Oberpfaffenhofen, D-82230 Wessling, Germany, `Pieter.J.Mosterman@dlr.de/Martin.Otter@dlr.de`
[‡]Department of Computer Science, Vanderbilt University, Nashville, TN 37235, `biswas@vuse.vanderbilt.edu`

KEYWORDS

Modelica, hybrid systems, physical system modeling, bond graphs, hybrid bond graphs

ABSTRACT

Bond graphs are a powerful and elegant formalism for modeling the continuous dynamics of physical systems. *Hybrid bond graphs* augment traditional bond graphs by a variable causality switching element to facilitate models with mixed continuous/discrete, *hybrid*, behavior. This paper describes an experimental hybrid bond graph simulator, HyBrSim, that handles (i) dynamic reconfiguration of the state vector, and (ii) recomputing state variable values based on conservation principles when additional constraints become active. Simulation applies graph propagation as opposed to explicit equation generation. One discrete state change may cause a number of further instantaneous state changes. HyBrSim supports the two basic types of event iteration caused by *parameter* and *time scale* abstraction.

INTRODUCTION

Physical systems may exhibit nonlinear continuous behaviors with steep gradients that complicate simulation. To enable efficient and accurate simulation, phenomena causing these nonlinearities can be abstracted into discontinuous changes, resulting in models of a mixed continuous/discrete nature, *hybrid systems.*[1]

Hybrid systems typically operate in piecewise continuous *modes*, each modeled as a system of explicit ordinary differential equations, ODE, or differential and algebraic equations, DAE. Discrete switching logic selects the active mode by activating and deactivating parts of the overall system which may cause (i) the causal structure to change, (ii) state variables to become inactive, (iii) dependencies between state variable to become active. Apart from these structural changes, a number of discrete transitions may be invoked before the system converges to a new configuration governed by differential equation behavior.

Previous work (Mosterman & Biswas 1998b), studied the nature and effects of discontinuities in physical system models and established a number of principles that govern hybrid physical model behavior. This paper describes the experimental *hybrid bond graph* simulator HyBrSim[2] which is based on these principles. HyBrSim facilitates variable causality bond graph models with the *controlled junction* (Mosterman & Biswas 1998b) as ideal switching element. It implements event iteration to handle the two general abstraction types in physical system models (Mosterman & Biswas 1998a): (i) *parameter* and (ii) *time scale* abstraction. Consistent initial value calculation applies the physical principle of conservation of state (Mosterman & Biswas 1997) which also handles structurally changing higher index DAEs (Mosterman 1998). Note, that this paper does not deal with hybrid simulation issues such as event detection, precision, and root-finding (Cellier 1979; Park & Barton 1996), and chattering behavior (Mosterman, Zhao, & Biswas 1997).

HYBRID MODELING

In this section, the basic model components of HyBrSim are described.

Bond Graphs

Conservation of energy and *continuity of power* govern physical system behavior. Bond graphs (Karnopp, Margolis, & Rosenberg 1990) apply *reticulation* to capture continuous physical system behavior by a set of nine primitive elements:

- *Ideal reversible processes* are modeled as storage elements C and I. For example, in the rotational me-

---

[*] Pieter J. Mosterman is supported by a grant from the DFG Schwerpunktprogramm KONDISK.

[1] See also the IEEE Control Systems Society CACSD Working Group on Hybrid Dynamic Systems web site at http://www-er.df.op.dlr.de/cacsd/hds.

[2] See also http://www.op.dlr.de/~pjm/hybrsim.

chanics domain, this corresponds to ideal torsional elasticity and inertias, respectively.

- *Ideal irreversible processes* are modeled as dissipation R, e.g., the damping process in a non-ideal spring in the mechanical domain.
- *Ideal sources* model the system context and are represented by sources of *flow*, S_f, and *effort*, S_e. In the rotational mechanics domain, this corresponds to ideal generators of angular velocity and torque, respectively.
- *Transformers* modulate efforts and flows (e.g., a lever) and are represented by transformer, TF, and gyrator, GY, elements.
- *Junctions* represent ideal connections and can be of equal effort (e.g., torque) type, 0 junctions, or equal flow (e.g., angular velocity) type, 1 junctions.

Because generic, these nine elements can be used to model a wide variety of physical systems in a number of domains, e.g., electrical, mechanical, and fluid, and they allow to combine model fragments from different domains.

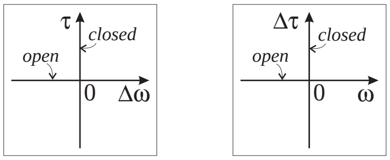
Power *bonds* connect elements. Associated with each bond is a pair of conjugate variables, the *intensive* variable, *effort*, and the *extensive* variable, *flow*. The *effort* $*$ *flow* product equals power (flow of energy) across a bond.

Energy interaction is often modulated as a function of system variables, e.g., because of geometrical constraints. Therefore, modulated versions of sources (MS_e and MS_f) and transformers (MTF and MGY) are introduced. An *active bond* into these elements provides the modulation parameter. Active bonds only carry one variable and do not transfer power, and, therefore, this models ideal modulation.

Block diagram elements perform mathematical operations on modulating signals (e.g., summing or integrating). Connections between block diagram elements are called *signals*, and they interact with bond graphs by active bonds as input and modulating signals as output. Presently, HyBrSim supports a clock, summing, step, sine, cosine, and integrator element. To distinguish signal carrying block diagram elements from energy elements in the bond graph, they are shown in a different color.

Bond Graph Simulators

A number of simulators (e.g., *20sim* (Broenink 1998)) have shown the viability of bond graphs as a modeling and simulation environment. However, they do not facilitate hybrid bond graph modeling and simulation. HyBrSim can be used to model and simulate traditional bond graph models (no multi-port elements or multi-bonds) and incorporates discontinuous behavior embodied by hybrid bond graph theory (Mosterman & Biswas 1998b).



(a) (b)

Figure 1: Causality of an ideal (a) clutch and (b) brake.

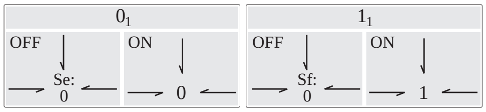


Figure 2: Operation of the controlled junction.

Hybrid Bond Graphs

In many situations, it is convenient to allow abstraction of nonlinearities into discontinuous changes.

The Controlled Junction An ideal switching element with variable causality facilitates discontinuities in bond graphs (Strömberg, Top, & Söderman 1993). For example, in rotational mechanics an ideal clutch can be modeled as a rigid, 0 angular velocity difference, connection when closed, or a freewheeling, 0 torque, connection when open (Fig. 1a). This switching is modeled by a controlled junction that can be in one of two states *on* and *off*. When *on*, the junction behaves as a traditional junction and when *off* the junction enforces either 0 effort or 0 flow on all its connections, in case it is a controlled 0 or 1 junction, respectively (Fig. 2). Therefore, a clutch is modeled by a controlled 0 junction. It enforces the difference in velocities (flow) to 0 and requires equal torque (effort) on its connections when it is closed. When it is open, it enforces 0 torque and there is no velocity constraint (ideal source). On the other hand, if it is connected as a brake, it is modeled by a controlled 1 junction that enforces 0 velocity on all connections when closed (Fig. 1b).

Consider the hybrid bond graph with the brake Sw_1 in Fig. 3. When Sw_1 is open the weight m_1 enforces a torque on I_1 through the rope with elasticity C and the closed clutch Sw_2 with friction R_1. When Sw_1 closes, the rotation of the connecting rod is forced to 0, but I_1 continues to move because of the friction R_1 in Sw_2, which causes a decay in angular velocity of I_1.

The controlled junction provides a convenient mechanism to model discontinuities and structural changes in bond graph models. It implements a primitive ide-

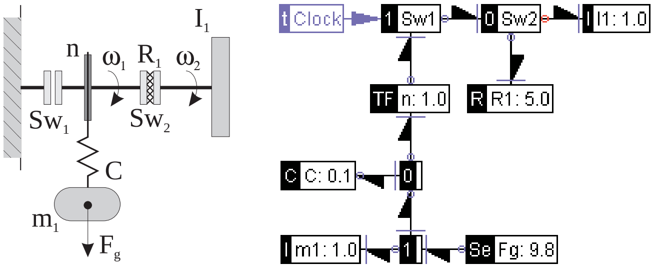


Figure 3: Variable causality hybrid bond graph.

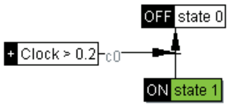


Figure 4: Finite state machine of clutch.

alized behavior which can be combined with other idealized processes to build complex behavior (e.g., the clutch resistance that is active when the clutch is closed is modeled as a separate ideal process).

The Discrete Model Part A finite state machine controls whether a controlled junction is *on* or *off*. State transitions are enabled by signal values in the hybrid bond graph that cross prespecified threshold values. These signals are shown as arrows connecting the effort or flow value of a power bond or the signal of a block diagram element to a junction (e.g., in Fig. 3 the 1 junction switching logic has access to time through the *Clock* element).

The finite state machine can be modified by selecting the junction *properties*. Signals that are available to a controlled junction show up as *signal ports* in the associated finite state machine. Fig. 4 shows the finite state machine for the brake Sw_1 in Fig. 3. The properties of a signal port specify the threshold value and inequality for the associated signal to trigger an event, and whether the event generation applies to *a priori* or *a posteriori* values, explained later.

The finite state machine modeling environment allows to (i) select the initial state, (ii) connect transitions between states, and (iii) enable these transitions by events generated by the signal ports. If multiple signal ports are connected to one transition, the conjunction of their truth values applies.

The Simulation Approach

HyBrSim performs simulation by graph propagation where each object contains its constituent equations. Before simulation, first a *causal* analysis is invoked, using the SCAP algorithm (Karnopp, Margolis, & Rosenberg 1990). Storage elements, C and I, that are dependent operate in *derivative causality* and signal a higher index DAE. Algebraic loops between R elements are presently not handled. When causal conflicts between sources, S_e and S_f, arise, this is reported and analysis is aborted.

After causal analysis, four stages are executed (i) propagate current values, (ii) change controlled junction states, (iii) perform integration step, and (iv) find consistent initial conditions.

1. *Clock* values are propagated to compute time modulation. Next, input values from sources, controlled junctions that are *off*, and state variables (independent I and C elements) are propagated throughout the bond graph. When propagation terminates, values on the active bonds are propagated into the block diagram structure. Along with values of *Integrator* and *Clock* elements these are the root-values that determine all block diagram variables.
2. When block diagram propagation terminates, switching conditions are evaluated whether transitions from active states are enabled.
3. After resulting mode changes have converged, the simulator performs one integration time step. Presently, this is simply done by calculating the new state values of independent I and C and *Integrator* elements using a forward Euler integration scheme.
4. If there are dependencies between storage elements, consistent values for the next integration step are computed by an iteration procedure applying conservation of state.

After convergence, the new values are propagated throughout the graph as described by the first step.

Output Variables

Variables associated with power bonds (effort and flow), signals, and the state of controlled junctions (*on* = 1 and *off* = 0) can be plotted and stored. Plotted variables are marked in the hybrid bond graph by colors that correspond with the colors in the graph. Because the plotting facilities of HyBrSim are not as sophisticated as, e.g., those of Matlab,[3] data of a simulation run can be stored on file in rows of variables at one sample point, k, including the time stamp, t. This allows plotting against k to analyze the behavior when several discontinuous changes occur at the same point in time. In a plot against time, this may be difficult to observe.

---

[3] All plots in this paper are generated using Matlab.

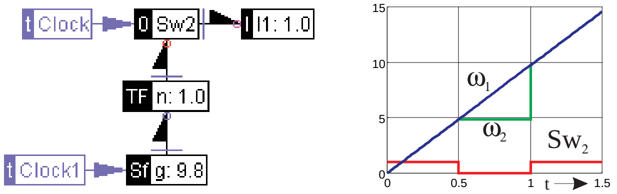


Figure 5: Simplified clutch system, $m_1 \gg I_1$.

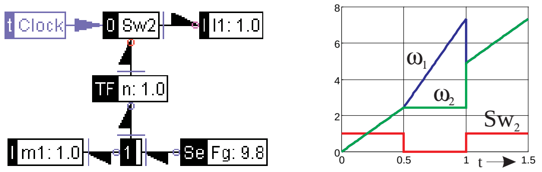


Figure 6: Simplified clutch system, $m_1 = I_1$.

HYBRID SIMULATION

Because of variable causality, every change of state of a controlled junction requires causal analysis. Changes in the state vector may occur, which requires a method to find consistent initial conditions given state vector values of a different dimension.

Variable Causality

In other work (Breedveld 1996), a *switching junction* was introduced that applies fixed causality. When *off*, it enforces 0 value on each of its connecting bonds without changing their causality. Though this solves the simulation problem of variable causality, it may generate incorrect behaviors. In the system in Fig. 3 when Sw_1 is closed (*off*), it means 0 torque is enforced on I_1 which would not show the exponential decay because of friction in Sw_2.

Reconfiguring the State Vector

When controlled junctions change their state, state vector variables may become inactive. For example, consider the rotational mechanics system in Fig. 3. If Sw_1 is open, the rope is not very elastic (C is small), and the clutch resistance, R_1, is large, this model can be simplified to the one in Fig. 5. Because $m_1 \gg I_1$, m_1 is modeled as a MS_f with value $9.8t$. Initially, the clutch, Sw_2, is closed (value 1) and the rigid connecting rod forces ω_1 on I_1, and, therefore its angular momentum is algebraically related to the input $L_1 = I_1\omega_1$ (Fig. 5). L_1 is not treated as a state variable and the state vector has dimension 0. When at $t = 0.5\ s$ the clutch opens, I_1 freely determines its angular momentum, i.e., its state L_1 with initial value the current value. The state vector now has dimension 1. When the clutch closes again, I_1 becomes dependent and its angular momentum is again algebraically determined by the velocity source causing a discontinuous change in value. In this mode, L_1 is not treated as a state variable anymore and the effect of I_1 reduces to that of a simple load.

Dynamic reconfiguration of the state vector is difficult to achieve in compiled simulation environments. Typically, the maximum size of the state vector needs to be specified and entries that are inactive are assigned a 0 value derivative (e.g., gPROMS (Barton 1992)). This may lead to problems when a state vector entry is added but the previous dependence requires a new initial value (e.g., at $t = 0.5\ s$ in Fig 5). This has to be explicitly specified as a state mapping function.

Because of the distributed nature in which HyBrSim performs simulation, it automatically deals with variable state vectors. States that are inactive are computed based on the enforced effort or flow value instead of with 0 value derivative. In compiled environments this may require a differentiation step during simulation.

Finding Consistent Initial Values

A more complicated problem arises when states become dependent on one another, as opposed to external sources. This requires collapsing multiple states into a vector with smaller dimensions. State values before the reduction have to be consistently mapped onto the reduced states. In previous work (Mosterman & Biswas 1997), we have identified the principle of *conservation of state* to govern this mapping as an algorithmic computation based on power gain between dependent elements.

For example, consider the system in Fig. 6 when $m_1 = I_1$. When at $t = 0.5\ s$ the clutch, Sw_2, opens, the mass m_1 and inertia I_1 build up different vertical velocities (Fig. 6). When the clutch closes, given a rigid connection, m_1 and I_1 become dependent and constitute one combined inertia. Governed by conservation of momentum (the state), the angular velocity of I_1 and vertical velocity of m_1 change discontinuously to satisfy both the constraint that $2\pi r v_1 = \omega_1 = \omega_2$ as well as $2\pi r \Delta p_1 + \Delta \mathrm{L}_1 = 0$. Here, Δp_1 is the change of state of m_1, ΔL_1 is the change of state in I_1, and r (the radius of the wheel with the rope attached) determines the power gain $2\pi r$ between the two. If $2\pi r = 1$ and m_1 and I_1 have equal values, the momentum after closing the clutch is distributed equally among m_1 and I_1 ($t = 1.0\ s$ in Fig. 6).

EVENT ITERATION

Compositional modeling techniques play an important role in analyzing behaviors of large, complex physical systems. Local switching functions avoid the difficult task of pre-defining *global* transition functions, but dynamic coupling (Mosterman & Biswas 1998b) may cause a *local* switch to trigger a sequence of additional local discrete switches before a new mode is arrived at where the system behavior continues to evolve continuously.

Two Iteration Types

In previous work (Mosterman & Biswas 1998a), we show there are two types of iteration loops. One that does affect state variable values and the other does not. The latter case results from abstracting small parasitic parameters, and thus behaviors, away which should not affect the state values, called a *parameter abstraction*. Sometimes, however, these small parameters are not abstracted away but their effect is collapsed into a discontinuous change at a point in time, referred to as *time scale* abstractions. In this case, the state values are affected.

The distinction between two types of event iteration seems to be not available in other simulation environments, although it has been shown that this is an indispensible feature for true hybrid system simulators (Mosterman & Biswas 1998a). HyBrSim allows specifying event conditions in terms of (i) the initial state values in the newly inferred mode immediately after switching, *a posteriori* values, and (ii) the state values immediately before switching, the *a priori* values. Event iteration that does not affect the state values is based on events specified by *a posteriori* values. Algorithm 1, a high-level simulation description, shows this as the *iterate*(α, x) procedure where α is the last mode before event iteration started and x the *a priori* state. Event iteration proceeds until no further mode changes occur, and the *a priori* state vector is updated to the *a posteriori* values, x^+. This change in *a priori* values may trigger a new event iteration by the *iterate*(α, x) procedure (Mosterman & Biswas 1998a).

Algorithm 1 Hybrid Simulation Algorithm

```
$x = \text{timeStep}(\alpha, x)$
$[\alpha^+, x^+] = \text{iterate}(\alpha, x)$
if $\alpha^+ \neq \alpha$ then
  repeat
    $\alpha = \alpha^+$
    $x = x^+$
    $[\alpha^+, x^+] = \text{iterate}(\alpha, x)$
  until $\alpha^+ = \alpha$
end if
```

Mythical Modes

To illustrate event iteration that does not affect the state vector, consider the freewheeling clutch in Fig. 7. Initially, brake Sw_1 is open and the clutch Sw_2 is closed. The inertia I_1 builds up momentum because of the gravitational force working on m_1. At $t = 0.2\ s$ (Fig. 8) Sw_1 closes which would cause the angular velocity in the system to go to 0 and I_1 would instantaneously discharge all its momentum. However, this results in a force that opens Sw_2 and I_1 continues to rotate with angular velocity equal to the value immediately before Sw_1 closed.

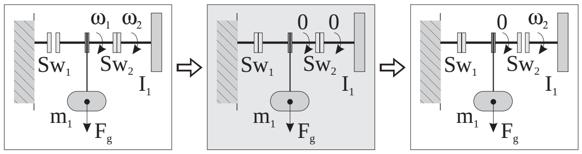


Figure 7: **Mythical mode in analyzing a freewheeling clutch.**

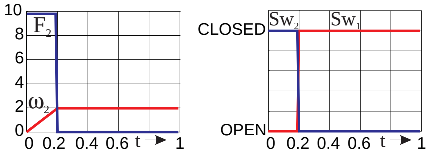


Figure 8: **A mythical configuration has no representation in real time.**

Because the intermediate configuration where both Sw_1 and Sw_2 are closed does not affect the state vector, it is called a *mythical* mode (Mosterman & Biswas 1998b). This is a parameter abstraction, and the switching conditions of the freewheeling clutch are, therefore, in terms of *a posteriori* values, indicated by a + sign in the signal ports of the finite state machine (e.g., Fig. 4).

Pinnacles

Another type of event iteration is caused by time scale abstraction. Consider the system in Fig. 9 where I_1 initially rotates freely. The brake Sw_1 closes at a given angle, and forces ω_1 to 0 ($t = 0.6\ s$ in Fig. 10). If the stiffness of the connecting rod is such that it cannot be abstracted away, it causes an elastic collision governed by the rotational analog of Newton's elastic collision rule which states $\omega_1^+ = -\epsilon\omega_1$, where ϵ represents the coefficient of restitution (= 1 for a perfectly elastic collision). If perfectly elastic, the angular velocity reverses instantaneously (Fig. 10), the inertia starts to rotate in the opposite direction, and the brake opens. The collision rule captures the torsional compression and expansion of the connecting rod on collision into an algebraic relation active at a point in time, called a *pinnacle* (Mosterman & Biswas 1998a), which does affect the system state.

Fig. 11 shows the hybrid bond graph model of the

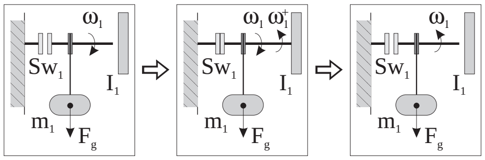


Figure 9: A pinnacle due to an elastic collision.

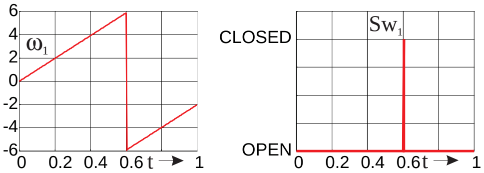


Figure 10: Pinnacles occur at a point in time.

elastic collision. The wall is modeled as a 0 value flow source (bottom-left). The clutch is modeled as a controlled 0 junction that closes when a 0 angle is reached. The angle is a geometric state and modeled by the *Int* element with initial value 0.5. Upon collision, the modulated $S_f : \epsilon$ becomes active and forces the opposite velocity difference between the wall and inertia. This causes dependency of the inertia and after consistent restart values are calculated, the *off* transition in the finite state machine is enabled because the velocity constraint for transition ($\omega_2 = f7 = f6 > 0.0$) is satisfied. The switching conditions are based on *a priori* values, indicated by a $-$ sign in the signal ports, because this is a time scale abstraction.

CONCLUSIONS

This paper describes an experimental hybrid bond graph simulator, HyBrSim, implemented to test and verify important concepts in hybrid system simulation. In variable structure systems the dimension of the state vector and the index of the DAE may change and initial conditions may have to be recomputed based on conservation principles. Two types of discrete event iteration loops are required to facilitate parameter and time scale abstraction. It is planned to generalize the concepts and algorithms to incorporate them in the object-oriented modeling language *Modelica*^{TM} (Modelica 1997).

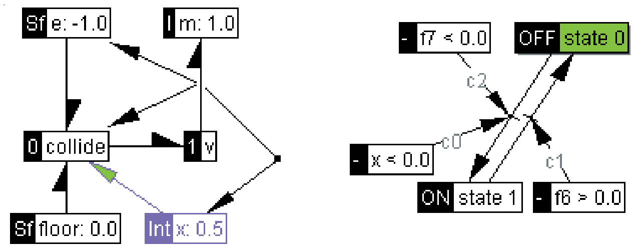


Figure 11: An elastic collision.

ACKNOWLEDGEMENT

The graphical appearance of the HyBrSim elements is based on a bond graph simulator by Róbert Bajzát.

REFERENCES

Barton, P. I. 1992. *The Modelling and Simulation of Combined Discrete/Continuous Processes.* PhD dissertation, University of London.

Breedveld, P. C. 1996. The context-dependent trade-off between conceptual and computational complexity illustrated by the modeling and simulation of colliding objects. In *CESA '96 IMACS Multiconference.* Lille, France: Ecole Centrale de Lille.

Broenink, J. F. 1998. Modelling, simulation and analysis with 20-sim. *Journal A* 38(3):22–25. Special CACSD issue.

Cellier, F. E. 1979. *Combined Continuous/Discrete System Simulation by Use of Digital Computers: Techniques and Tools.* PhD dissertation, The University of Arizona.

Karnopp, D.; Margolis, D.; and Rosenberg, R. 1990. *Systems Dynamics: A Unified Approach.* New York: John Wiley and Sons, 2 edition.

Modelica. 1997. A unified object-oriented language for physical systems modeling. Modelica homepage: http://www.Dynasim.se/Modelica/.

Mosterman, P. J., and Biswas, G. 1997. Hybrid Modeling Specifications for Dynamic Physical Systems. In *1997 International Conference on Bond Graph Modeling and Simulation (ICBGM '97)*, 162–167. Phoenix, AZ: Society for Computer Simulation.

Mosterman, P. J., and Biswas, G. 1998a. Modeling and simulation semantics for hybrid dynamic physical systems. In *Hybrid Systems V.* Springer-Verlag. in review.

Mosterman, P. J., and Biswas, G. 1998b. A theory of discontinuities in dynamic physical systems. *Journal of the Franklin Institute* 334B(6):401–439.

Mosterman, P. J.; Zhao, F.; and Biswas, G. 1997. Model semantics and simulation for hybrid systems operating in sliding regimes. In *AAAI Fall Symposium on Model Directed Autonomous Systems*, 48–55.

Mosterman, P. J. 1998. State Space Projection onto Linear DAE Manifolds Using Conservation Principles. Technical Report #R262-98, Institute of Robotics and System Dynamics, DLR Oberpfaffenhofen, P.O. Box 1116, D-82230 Wessling, Germany.

Park, T., and Barton, P. I. 1996. State event location in differential-algebraic models. *ACM Transactions on Modeling and Computer Simulation* 6(2):137–165.

Strömberg, J.-E.; Top, J.; and Söderman, U. 1993. Variable causality in bond graphs caused by discrete effects. In *Proceedings of the International Conference on Bond Graph Modeling*, 115–119.