Formal Specifications from Hybrid Bond Graph Models

Pieter J. Mosterman and **Gautam Biswas**
Center for Intelligent Systems
Box 1679, Sta B
Vanderbilt University
Nashville, TN 37235.
pjm,biswas@vuse.vanderbilt.edu

Abstract

Physical system behavior is continuous, but the use of modeling abstractions to simplify system description can result in behavior analysis on a hierarchy of time scales. At a given level of detail, behaviors on faster time scales are perceived to be instantaneous, therefore, the resulting *hybrid models* encompass continuous behaviors and discrete model configuration changes. These changes cause discontinuities in system behavior generation which violate the physical principle of *continuity of power*, and sometimes cause an instantaneous loss of energy in the system. This paper establishes a formal specification for handling discrete model configuration changes at well-defined points in time, and this allows for a consistent transfer of the continuous system state from a previous model configuration to a new one. This is based on the principle of *invariance of state*. Simulation algorithms designed to operate on hybrid models define behavior generation schemes that operate on the interval (continuous) to point (discrete) to interval (continuous) switches on the time line.

Introduction

Physical systems are inherently continuous and their behaviors are governed by the principles of conservation of energy and continuity of power [3]. Perceived discontinuities are in reality nonlinear continuous behaviors which operate at time scales much smaller than the time scale of interest. For efficient analysis and behavior generation schemes, the differences in time scale may be exploited so that the nonlinear behaviors can be abstracted to manifest as ideal discontinuities at *points in time*. An example is an ideal *elastic* collision between a body and a floor where the velocity of the body reverses instantaneously on impact. In reality, the collision takes a small time interval during which kinetic energy is converted into potential elastic energy, which then reverts back completely to kinetic energy for the body. Discontinuous effects can also be created by *parameter* abstraction [11]. Small parasitic physical effects that cause nonlinear continuous effects are abstracted away to simplify system description. For example, an *ideal non-elastic* collision between two bodies involves instantaneous discontinuous changes in velocity for the two bodies at the point of impact. A more precise model would have included small elasticity coefficients for the two bodies, and the period of impact would be a small but finite time interval, in which the change in velocities for the bodies would occur in a continuous manner. Models that combine continuous and discrete effects are called *hybrid* systems.

During discontinuous changes, physical laws of conservation of energy and continuity of power may be violated [11]. In such situations, the initial state vector following the discontinuous changes is computed using the principle of *conservation of state* along with explicitly modeled interactions with the environment. In previous work [11, 13], this theory of discontinuous configuration changes in physical system models has been developed into a *hybrid bond graph* modeling paradigm that combines traditional bond graph elements with *ideal switching* elements controlled by finite state automata. Formal schemes for verifying the correctness of models based on the principle of *divergence of time* have also been developed [10, 13]. The hybrid bond graph formalism can be effectively applied to systematically design and analyze hybrid models of dynamic physical systems. This paper focuses on developing a formal semantics for analyzing systems with mixed continuous/discrete components.

A Hybrid Modeling Paradigm

Hybrid models operate in continuous modes (typical physical system behavior), but at points in time when signal values cross pre-defined thresholds or when explicit external (control) events are imposed on the system [14], changes in model configurations cause discrete changes in system behavior. An important observation is that the temporal trajectory of system behavior becomes piecewise continuous, where *simple discontinuities* can occur only at well-defined points in time. The key to developing a correct modeling paradigm is to ensure that interaction between the continuous and discrete modeling formalisms is un-

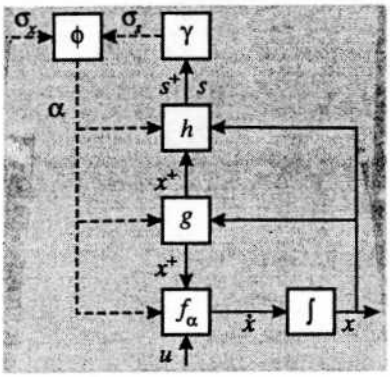

Figure 1: **A general hybrid system.**

ambiguous, rigorous, and consistent.

General Hybrid Dynamical System

The general architecture for a hybrid dynamical system model illustrated in Fig. 1 can be specified by the 9-tuple [14]:

$$H =< I, \Sigma, \phi, X, U, f_\alpha, g, h, \gamma >, \quad (1)$$

Each mode of continuous behavior is given a unique state label $\alpha_k \in I$. Continuous behavior is governed by field f_{α_k} which determines the continuous state vector x_{α_k}. The function h computes signal ($s \in S$) values from the state vector x_{α_k} in mode α_k, which may generate discrete events Σ specified by a mapping γ. γ is usually defined in terms of signal values *reaching* or *crossing* pre-specified threshold values. Occurrence of a discrete event suspends the continuous behavior mode α_k, and a new mode, α_{k+1}, is generated by the discrete transformation ϕ. The function g computes a new state vector, x^+, for the new operational mode α_{k+1} using values of the continuous state vector x_{α_k} in the previous operational mode α_k.

The Continuous Model

Dynamic physical system models are best represented as a set of differential equations on the system state vector. For example, the falling rod in Fig. 2 can be described by three state variables, the rod's linear velocities, v_x and v_y, and its angular velocity, ω. When it is falling freely, only gravity acts on the center of mass and accelerates vertical movement. This can be described by the differential equation

$$\dot{\omega} = 0, \dot{v}_x = 0, \dot{v}_y = a_g, \quad (2)$$

where a_g is the gravitational acceleration.

Differential equation state space models, supplemented by algebraic constraints (DAEs) directly reflect underlying physical principles such as Kirchhoff's laws and phenomenological relations like Ohm's law. Many model parameters have an immediate physical meaning and equations can be systematically derived from bond graphs, network representations, and block diagrams [2]. A general representation of an ODE model derived from DAEs is: $\dot{x}(t) = f_\alpha(x(t), u(t), t)$. The field, f_α, describes continuous temporal evolution of system behavior in a mode of operation, α, with the input vector, u, and the continuous state vector, x. Note that f_α is unique in mode α.

The Discrete Model

Discrete events are modeled by a discrete indexing set, I and a switching function, $\phi : I \times \Sigma \rightarrow I$. The set of discrete states corresponds to

- *real* modes, where system behavior is governed by energy principles, therefore, the state vector changes in time, and
- *mythical* modes [10, 13], where the system behavior transitions are instantaneous and state vector changes are used to infer real modes by ϕ.

Σ captures the *event* set. Events, σ, may be associated with closed loop control, Σ_s, or they can be governed by external, open loop, control signals, Σ_x (see Fig. 1). ($\Sigma = \Sigma_s \times \Sigma_x$.) The closed loop control is a function of the system's physical process variables. ϕ, usually implemented with Petri-Nets or Finite State Automata, determines the next state after an event occurs.

Interactions

Interactions between the continuous and discrete modeling formalisms have to be specified correctly. For states that correspond to modes of continuous operation, ODEs determine behavior. A discrete event causes the system to change operational mode, and the correct state vector in the new mode is determined by the function $g : X \times I \rightarrow X^+$. X defines state vector values just before switching occurs, and X^+ represents state vector values at the initial point in time when a switch or mode change has occurred. A function $h : X \times U \times I \rightarrow S$ determines signal values S and S^+, computed by h from X and X^+, respectively. The function $\gamma : S \times S^+ \rightarrow \Sigma_s$ generates discrete events from the signal values. The interaction between the continuous and discrete part consists of

- discrete events generated by the continuous signals, and
- a change of operational mode by the discrete model, requiring a consistent mapping of the continuous state vector.

An example, adapted from [8], illustrates a rigid body collision of a rod falling to the floor (Fig. 2). On hitting the floor, the rod may disconnect after a point in time where contact occurred, slide along the floor and rotate about its point of contact, or just stick

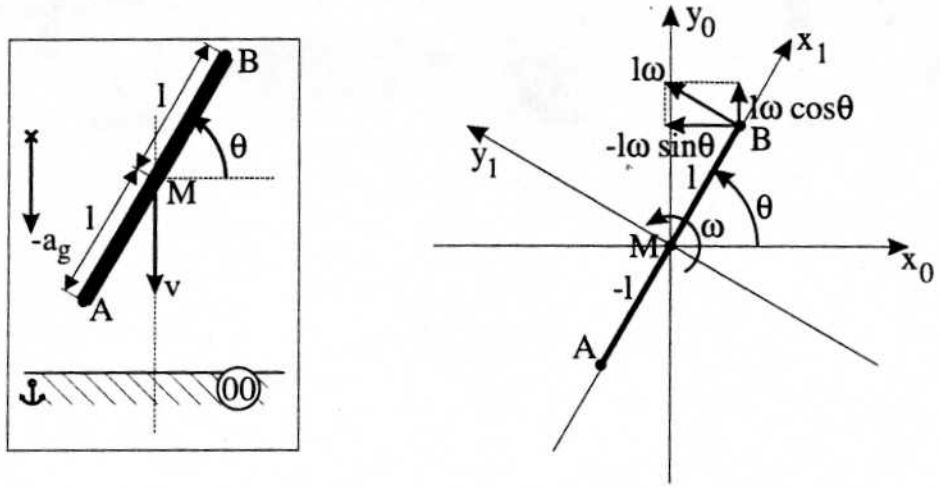

Figure 2: **A collision between a body and a floor.**

at the point of contact and rotate. Whether the rod sticks at the point of contact or slides is determined by a Coulomb friction coefficient, μ, and whether the horizontal force exceeds a threshold value given by:

$$\gamma : \begin{cases} |F_{A,x}| > \mu F_n \Rightarrow \sigma_{slide} \\ |v_{A,x}| \leq v_{th} \Rightarrow \sigma_{stuck}. \end{cases} \quad (3)$$

The friction force cannot be predetermined because it depends on the normal force at the surface. When the σ_{slide} event becomes active, the friction force comes into effect and its direction is always opposed to the direction of velocity. The events, σ_{zero}, σ_{pos}, and σ_{neg}, correspond to states 1, 2, and 3 of the automata in Fig. 3, respectively. The events that cause these internal state changes are defined as:

$$\gamma : \begin{cases} v_{A,x} = 0 \Rightarrow \sigma_{zero} \\ v_{A,x} < 0 \Rightarrow \sigma_{pos} \\ v_{A,x} > 0 \Rightarrow \sigma_{neg}. \end{cases} \quad (4)$$

Since this behavior is piecewise continuous only *simple* behavior discontinuities occur at time points, which implies that operational modes have limit values at discontinuities (e.g., Fig. 3). The complete event set for Coulomb friction is $\Sigma = \{\sigma_{slide}, \sigma_{stuck}, \sigma_{zero}, \sigma_{pos}, \sigma_{neg}\}$. A distinction is made between sliding with 0 velocity and being stuck, though the velocity of the rod at the surface is 0. In case the rod is stuck, the model does not have a degree of freedom in the x-direction.

As an example of a transfer of the continuous state vector between model configurations, consider the falling rod when it first makes contact with the floor, the model moves from mode α_{00} to mode α_{01} in Fig. 4. At this point it reaches a model configuration where v_x and v_y at the rod-tip are forced to 0. This requires the center of mass to move in the x and y direction with a velocity that is completely determined by the

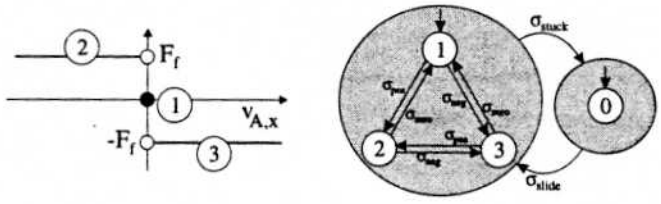

Figure 3: **Coulomb friction.**

angular velocity. Conservation of momentum determines that the initial momentum in the y direction is redistributed over the angular and linear momenta. Fig. 2 shows that the linear velocities can be represented in coordinate frame (x_0, y_0) by

$$v_x = l\omega^+ sin\theta, v_y = -l\omega^+ cos\theta. \quad (5)$$

A detailed derivation (see equations (20) through (24)) yields the new state vector:

$$g_{\alpha_{01}} : \begin{cases} \omega^+ = \frac{1}{J+ml^2}(\omega J + ml(sin\theta v_x - cos\theta v_y)) \\ v_x^+ = l\omega^+ sin\theta \\ v_y^+ = -l\omega^+ cos\theta \end{cases} \quad (6)$$

Model Execution Semantics

A discontinuous change that occurs in given mode α_k has to happen at a point in time, say t_s. The state vector at this point, x_{α_k} labeled $x_{\alpha_k}^- = \lim_{t \uparrow t_s} x_{\alpha_k}(t) = x_{\alpha_k}(t_s)$. This becomes the *a priori* vector for the state computation function g that determines the initial state x^+ in the new mode α_{k+1}. The state vector x^+ is referred to as the *a posteriori* vector computed by g. The new state vector, x^+, may immediately trigger further discrete events determined by h and ϕ, causing a sequence of discrete mode changes till a new operational mode, α_m, is reached at which no

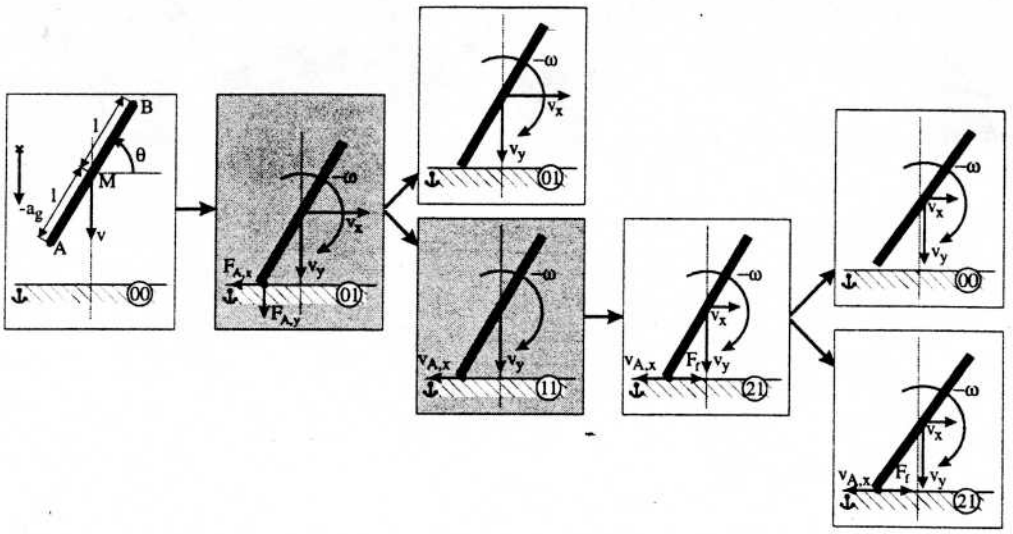

Figure 4: **Operational modes of falling rod.**

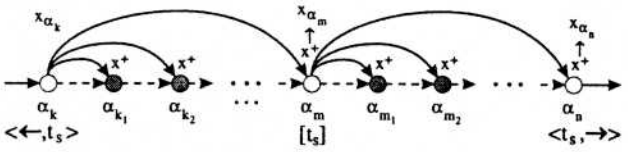

Figure 5: **System state is derived from the original state vector.**

further switching occurs. All the intermediate states traversed between two continuous modes are mythical [10, 13]. The sequence of state and state vector changes is illustrated in Fig. 5. At mode α_m system behavior evolution in time resumes, with the state vector $x_{\alpha_m}(t_s) = x^+$. Sometimes, mode α_m may represent just a point of *continuous* operation (such as, the point of contact in an elastic collision [13]). State vector changes from $x^-_{\alpha_k}(t_s)$ to x^+ in the new real mode may cause the γ function to generate additional events resulting in another sequence of discrete state changes before the next continuous operational mode, α_n, is arrived at (see Fig. 5).

Consider the falling rod in Fig. 4. Initially, it is falling freely under gravity (mode α_{00}). On hitting the floor it exerts a force with two components, $F_{A,y}$ and $F_{A,x}$ (mode α_{01}). Since the floor surface has Coulomb friction, the rod immediately starts to slide if $|F_{A,x}| > \mu F_n$, mode α_{11}. Otherwise, it sticks and rotates around the point of contact (mode α_{01}). When the rod starts to slide, the floor exerts an opposing friction force, F_f. In this case, the initial kinetic energy before contact is redistributed over the angular and vertical momentum to ensure the vertical velocity of the rod-tip, $v_{A,y}$, is 0. The horizontal velocity of the rod-tip, $v_{A,x}$, is determined by the angular velocity, ω, and the horizontal velocity of the center of mass, v_x. Since v_x is independent of ω and determined by F_f, it is initially 0 and the discontinuous change of ω results in a discontinuous change of $v_{A,x}$. Therefore, the system changes from the operational mode where $F_f = 0$ to mode α_{21} where $F_f = \mu F_n$.

The grayed modes of operation in Fig. 4 are mythical. They do not have physical meaning, therefore, no representation on the real time-line. However, they play the role of transition points for locally defined switching functions.

Temporal Evolution of State

Discontinuities are abrupt point changes, caused by modeling abstractions. Discontinuities that persist in time intervals would violate continuity of power and conservation of energy principles. Further, asymmetry in temporal evolution ensures that the state vector in modes of continuous operation has to be *left closed* over the time intervals these modes are active. Mode changes and discontinuous changes in the continuous state vector can only occur at points in time t_s. We have shown in other work [13] that further continuous evolution may cause a mode change, α_m to α_n, at the point of transfer, but no discontinuous change can occur in the continuous state vector between $x_{\alpha_m}(t_s)$ and $x^+_{\alpha_n}(t_s)$ since its initial value would be derived from $\lim_{t \downarrow t_s} x_{\alpha_n}(t)$ which requires knowledge of future behavior and conflicts with the assumption of causality in physical system models [14].

Consider the stiction force when the rod disconnects from the floor as it slides. If this force causes a discontinuous change in the vertical velocity of the rod, $\lim_{t \downarrow t_s} v_y(t)$ differs from the actual value $v_y(t_s)$. However, the value of $\lim_{t \downarrow t_s} v_y(t)$ may be such that its value indicates that the rod would have gotten stuck. This implies that in addition to the current

state $v_y(t_s)$ and model configuration, the operational mode needs to know future modes and the limit values of state variables looking back in time. Such systems are acausal which is physically impossible and results in ill-defined models.

Since no discontinuous change of the state vector can occur, it is continuous over a left closed interval in time. This only requires the system state to operate continuously in left closed intervals but field f is not required to be differentiable. Therefore, other derived system variables may still change discontinuously as a result of configuration changes. These jumps are well-defined by the continuous state vector and model configuration.

Invariance of State

A discontinuous change in the state vector may invoke further mode transitions. The state vector in a new mode is computed from the last continuous state vector, and the state vector in all new modes is computed from the last continuous state vector before switching started. This is the principle of *invariance of state* [13].

To illustrate, consider the falling rod in Fig. 2. When it hits the floor, its vertical momentum is distributed over its angular, horizontal and vertical momentum to ensure its rotation and translation of center of mass are such that the point of contact does not move (mode α_{01}). In this situation, if the force at the rod-tip, $F_{A,x}$, exceeds a threshold value, it immediately starts to slide (mode α_{11}). The rod-tip moves freely in the x-direction, and its initial vertical momentum is distributed only over its *a posteriori* angular momentum and vertical momentum to ensure the y-value does not change at the point of contact. If the continuous state vector in the sliding mode, α_{11}, was computed from the previously inferred mode, α_{01}, it would have a horizontal velocity associated with its center of mass which would keep the rod-tip from moving in the x-direction as well, which is incorrect. This demonstrates the importance of the proper computation of the state vector across a series of discontinuous changes.

Divergence of Time

Discontinuous configuration changes in system behavior are instantaneous so a model verification technique based on the principle of *divergence of time* ensures that the model does not end up in a loop of instantaneous changes where system behavior does not progress in time [13]. In previous work, we have developed a multiple energy phase space analysis that establishes divergence of time before simulation is performed [11].

As an example, consider the falling rod when it starts to slide because its force in the vertical direction exceeds a threshold value. If the rod is specified to stick when the velocity of its rod-tip is below a certain threshold value, it may not have sufficient initial vertical momentum to maintain a high enough vertical velocity. Based on the specifications, this moves the model into the configuration where it sticks and rotates around the point of contact. However, in this configuration, based on the initial vertical momentum, its horizontal force causes it to start sliding and a loop of consecutive changes occurs.

Hybrid Bond Graph Modeling

Modeling of physical systems typically starts out with an ideal configurational representation from which a set of component equations is generated based on first principles. As an intermediate step, a generic representation can be established to aid in a systematic derivation of the set of equations. Depending on the system configuration, the variables of these constituent equations are connected together, often in the form of a block diagram, and a number of mathematical manipulation steps are performed to establish a set of explicit differential equations.

Bond Graphs

To support the modeling process, bond graphs [7] can be used as a generic representation across domains in terms of a small set of primitive elements with well-defined characteristics. Bond graphs are based on the observation that all interaction in dynamic physical systems occurs by energy exchange, and, therefore, provides a unifying modeling approach across domains (e.g., electrical, mechanical, chemical). Energy exchange is captured by its flow, or power, between elements. Power is the product of two conjugate variables, e.g., $Power(P) = Voltage(V) \times Current(I)$ in the electrical domain. In thermodynamics, these variables are distinguished as *intensive* and *extensive* variables, where intensive variables are defined at points and extensive variables are defined as an aggregate property over a region or volume [4]. For example, two bodies with the same velocity that are connected together still have the same velocity but now have twice the momentum. In bond graph terms, the intensive variables are referred to as efforts, e, and the extensive variables as flows, f, e.g., voltage and current, pressure and volume flow.

The basic bond graph elements are energy *storage*, C and I, and energy *dissipation*, R, representing ideal reversible and irreversible physical processes, respectively. These elements are connected by a *junction* structure that consists of two basic types: (i) 0-junctions, the equivalent of electrical parallel connections, that enforce common effort variables like voltage and pressure, and (ii) 1-junctions, the equivalent of electrical series connections, that enforce common flows like current and volume flow. The model context is specified by ideal sources of effort, Se, and flow,

Sf, that supply an effort or flow independent of their load. If loading effects are important, they have to be modeled by R, C, or I elements, and the context has become part of the model. The set of primitive elements is completed by transformers, TF, and gyrators, GY. These elements are used to transform power impedance within and between physical domains.

Based on this primitive set of nine elements, a wide variety of systems can be modeled at different levels of detail [5]. They rigorously specify f, X, U, and, therefore, capture the continuous model aspects unambiguously. The fundamental laws on which bond graphs are based, *conservation of energy* and *continuity of power* [3], prohibit modeling of discontinuities in physical systems. Mosterman and Biswas [13] have been investigating the nature and effects of discontinuities in physical systems and established a theory from which they derived the hybrid bond graph modeling paradigm [13].

Hybrid Bond Graphs

Hybrid bond graphs rely on a local ideal *switching* element to dynamically construct active model configurations. By introducing an ideal element, the concept of reticulation on which bond graphs are based is not violated, and non-ideal switching can be modeled by including ideal energy dissipating or storing elements. A higher level control structure forms a meta-model that controls the state of each of the ideal switching elements. The meta-model is implemented as local finite state automata, one associated with each switch. The switch becomes active when signal values in the current bond graph configuration cross threshold values. The system may transition through one or more configuration changes before it arrives at a bond graph configuration where conservation of energy and continuity of power governs system behavior again. During configuration changes these laws may be violated and model behavior is governed by the principle of invariance of state discussed earlier.

Local switches are implemented as *controlled junctions*. These junction can be

- *on* in which state they operate as normal junctions, and
- *off* in which state they are deactivated.

To ensure correct loading, the deactivated junctions are replaced by either 0 value effort sources or 0 value flow sources, respectively (Fig. 6). When turned *off*, they inhibit transfer of energy between model fragments that are connected through the junction. A finite state automata associated with each controlled junction interacts with the bond graph and determines the *on*/*off* state of the junctions based on events generated by the bond graph, or from external control events.

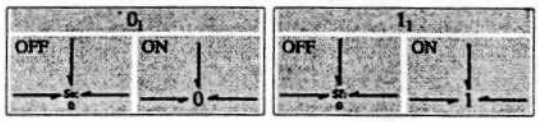

Figure 6: **Operation of the controlled junction.**

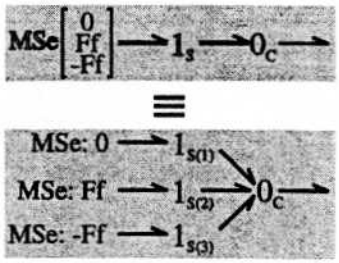

Figure 7: **A multi-bond controlled junction to model Coulomb friction.**

Piecewise continuous functions are represented in a compact form of the controlled junction by relying on multi-bond notations [3]. For example, Coulomb friction in Fig. 3 in the hybrid bond graph framework is represented by the multi-bond representation in Fig. 7, where all source elements now are continuous functions over their active areas. This is required to make mode switches within source elements explicit in the hybrid bond graph framework. The net result is this phenomena is easily incorporated into the mode-switching algorithm. This guarantees consistency in behavior generation, since all discrete phenomena are handled by one mechanism and all other influences are continuous.

Ideal switching behavior is established by enforcing 0 effort or 0 flow [15] and by using signal values from the bond graph rather than power bonds to generate discrete events. Controlled junctions in the bond graph are marked with a subscript, e.g., 0_1, 1_1, that is used to identify the associated finite state machine. The signal values that are used by the finite state machine to generate discrete events are shown as active bonds into the controlled junction (see Fig. 8), and specify the h function. As the finite state automata represent a junction's control specification, they are referred to as CSPECs. A CSPEC may contain sequential logic with any number of internal states. However, each one of these has to map onto either a junction's *on* or *off* state and, since each state reflects a physical manifestation, *on* and *off* states have to alternate in each transition sequence.

Local configuration changes may cause sequences of consecutive configuration changes, and require the continuous state vector to be transferred correctly between modes of continuous operation. When configuration changes occur, buffers may become dependent or modeled δ-sources may become active, which may

result in discontinuous change of the continuous state vector specified by g. Two cases exist: (i) one or more buffers may become dependent on a source, or (ii) two or more buffer elements become dependent on each other or δ-sources become active, and the state vector between the two configurations is different [10, 11]. In the first case, the energy stored in the dependent buffers is determined by the value of the source, u,

$$p_i^+ = r_{S,i} C_i u, \tag{7}$$

where $r_{S,i}$ is the gain of the route from the source to the dependent buffer, i, with value C_i. In the second case, the general formula

$$p_0^+ = p_0 + \sum_{buffers,i} a_{\delta,i} r_{i,0} + \sum_{sources,j} a_{\delta,j} r_{j,0} \tag{8}$$

can be applied, where

$$a_{\delta,i} = p_i^+ - p_i \tag{9}$$

is the loss of generalized charge or momentum in the dependent buffers. The values of dependent states are primed to

$$p_i^+ = r_{0,i} \frac{C_i}{C_0} p_0^+ \tag{10}$$

which, along with (8) and (9), can be applied to determine the new value of the independent state variable, p_0^+. Note that if no δ-sources become active, conservation of state holds because the amount of generalized charge and momentum added to the independent buffer equals the loss by each of the dependent buffers combined. Therefore, the total amount of charge and momentum in both modes remains the same. For n dependent buffers, buffer 0 is chosen in integral causality and the new value of its stored energy, p_0^+, is determined by

$$p_0^+ = p_0 + \sum_{i=1}^{n-1} (p_i^+ - p_i) r_{i,0}. \tag{11}$$

This can be expressed in terms of the value of the independent buffer, p_0^+, by substituting (10)

$$p_0^+ = p_0 + \sum_{i=1}^{n-1} (r_{0,i} \frac{C_i}{C_0} p_0^+ - p_i) r_{i,0} \tag{12}$$

or [12],

$$p_0^+ (1 - \sum_{i=1}^{n-1} r_{i,0} r_{0,i} \frac{C_i}{C_0}) = p_0 - \sum_{i=1}^{n-1} r_{i,0} p_i, \tag{13}$$

where $r_{i,0}$ is the gain of the route from buffer i to buffer 0 and C_i the buffer value of buffer i. Note that this may result in loss of energy to the environment [13].

The meta-level control model separates discrete behaviors as specified by the finite state automata and the continuous state mapping function from the continuous operation of the system where conservation of energy and continuity of power govern behavior. As part of the meta-model, a Mythical Mode Algorithm (MMA) is formulated to govern the global effects of configuration changes [13]. The CSPEC conditions have to be verified to always generate sequences of configuration changes that terminate in a model configuration that has a real manifestation.

Implementation

The bond graph model of the idealized thin rod and idealized floor, and the fragments dynamically generated by simulation [12] are shown in Fig. 8. The rod is assumed to have three degrees of freedom: angular velocity, with a buffer element associated with the J inertia, and linear velocity in the x and y directions, with buffer elements m_x and m_y, respectively. The relation between those velocities is derived geometrically (Eq. (5)), and modeled by a modulated transformer. Gravity is modeled as a constant effort source, ma_g, in the y direction at the center of mass.

The x and y components of the forces and velocities at point A connect to the model at the 0_C junction. If the body is moving freely, this junction is *off*. If the body is in contact with the floor, 0_C is *on* and if no other elements are connected, it enforces a 0 velocity. The friction force, $F_f = \mu F_n$, in the x direction is modeled as a piecewise continuous modulated source, MS_e, producing force values $0, F_f, -F_f$ at A, opposite to the direction of the surface velocity.

The control specifications (CSPEC) of the switching junctions are specified by finite state automata, one for each controlled junction. The controlled junction 1_S is specified by a hierarchical finite state machine, which can be in one of several *on* states, depending on the bond graph signals. Depending on the specific state, a part of the piecewise continuous friction function is active. In its *off* state, the junction enforces 0 flow.

Initially, the rod is moving freely and controlled junctions 0_C and 1_S are *off*. Replacing the junctions with their 0 value sources results in the bond graph (mode α_{00}) shown in Fig. 8. The position of the rod-tip closest to the floor, y_A, is determined by the sum of the position of the center-point, $y_M = \int v_y$, and the distance of the rod-tip from the center point, $-lsin\theta$. If this position, $x_A = \int v_y - lsin\theta$, becomes 0, the rod collides with the floor and 0_C comes *on*, and the model transitions to mode α_{01}. This results in dependency between the linear and angular velocities, and the energy redistribution that may be required, specified by g, is computed. If the rod-length and angle of collision are such that 1_S comes *on* (the model transitions into α_{11}), the rod begins to slide. Based on the for-

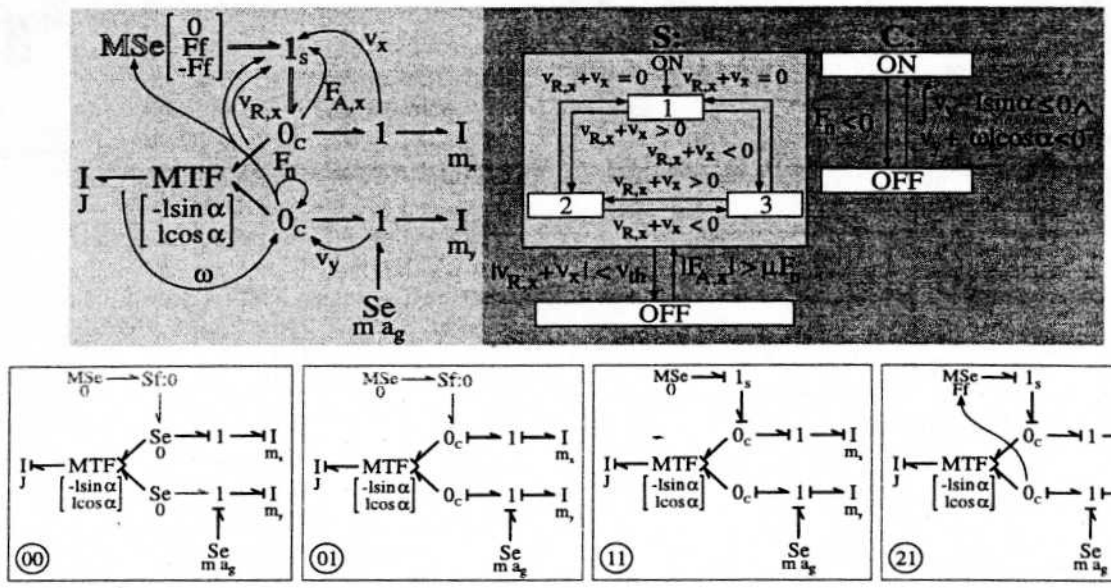

Figure 8: **Dynamically generated models.**

mulas for energy redistribution, the function, g, can be calculated as before, and the piecewise continuous friction function may move into its F_f area, mode α_{21}. As shown by this example, the hybrid bond graph approach provides a seamless integration of configuration changes based on local switches. Other examples of hybrid bond graph models are discussed in [13].

The continuous system model, directly derived from each operational mode of the hybrid bond graph (Fig. 8) is shown below ($f_{\alpha_{00}}$ is given in Eq. (2)):

$$f: \begin{cases} \alpha_{01}: \begin{cases} \dot{\omega} = \frac{-mlcos\theta}{J+ml^2} a_g \\ \dot{v}_x = lsin\theta\dot{\omega} \\ \dot{v}_y = -lcos\theta\dot{\omega} \end{cases} \\ \alpha_{21}: \begin{cases} \dot{\omega} = \frac{-ml(cos\theta-\mu sin\theta)}{J+ml^2cos\theta(cos\theta-\mu sin\theta)} a_g \\ \dot{v}_x = -\mu(lcos\theta\dot{\omega} + a_g) \\ \dot{v}_y = -lcos\theta\dot{\omega} \end{cases} \end{cases} \tag{14}$$

The discrete control model, ϕ, is specified by the two automata functions, **C** and **S**. The specification for the γ and h functions are shown below.

$$\gamma: \begin{cases} y_A < 0 \wedge p_{A,y} < 0 & \Rightarrow \sigma_{contact} \\ F_n < 0 & \Rightarrow \sigma_{free} \\ |F_{A,x}| > \mu F_n \wedge y_A < 0 \wedge p_{A,y} < 0 & \Rightarrow \sigma_{slide} \\ |v_{A,x}| \leq v_{th} \vee F_n < 0 & \Rightarrow \sigma_{stuck} \\ v_{A,x} = 0 & \Rightarrow \sigma_{zero} \\ v_{A,x} < 0 & \Rightarrow \sigma_{neg} \\ v_{A,x} > 0 & \Rightarrow \sigma_{pos} \end{cases} \tag{15}$$

$$h: \begin{cases} y_A = \int v_y dt - lsin\theta \\ v_{A,x} = v_x - l\omega sin\theta \\ p_{A,y} = m(v_y + l\omega cos\theta) \\ F_n = \begin{cases} 0 & \text{if } \alpha_{00} \\ m(\dot{v}_y - a_g) & \text{otherwise} \end{cases} \\ F_{A,x} = \begin{cases} 0 & \text{if } \alpha_{00} \\ m\dot{v}_x & \text{otherwise} \end{cases} \end{cases} \tag{16}$$

To derive g it is observed that in a hybrid bond graph discontinuous state changes only occur if buffers become dependent. Two cases exist:

1. one or more buffers may become dependent on a source, and
2. two or more buffer elements may become dependent on each other and δ-sources become active. In this case, the state vector between the two configurations is different [11].

In the first case, the *a posteriori* energy stored in the dependent buffers, p_i^+, is determined by the value of the source, u (see equation 7). In the second case, Dirac pulses, δ, are induced that enforce a discontinuous change of the independent, integrated, state variable, p_0^+. The area of such a pulse combined with the gain from its origin, either a source or dependent buffer to the independent buffer, specifies a change of p_0^+. This change is given by the general formula given in equation 8. The area $a_{\delta,j}$ is the explicitly modeled interaction with the environment. The area $a_{\delta,i}$ can be calculated from equation 9, which is the loss of generalized charge or momentum in the dependent buffers. The new signals generated by dependent states, $\frac{p_i^+}{C_i}$, are forced to values determined by the new signal from the independent buffer, $\frac{p_0^+}{C_0}$, and the route gain from

the independent buffer to the dependent one. This is described by the equation 10, which along with equation 8 and equation 9 can be applied to determine the new value of the independent state variable, p_0^+.

In the special case that no explicitly modeled δ-sources become active, conservation of state holds because the amount of generalized charge and momentum added to the independent buffer equals the loss by each of the dependent buffers combined. Therefore, the total amount of charge and momentum in both modes remains the same. For n dependent buffers, buffer 0 is chosen in integral causality and the new value of its stored energy, p_0^+, is determined by

$$p_0^+ = p_0 + \sum_{i=1}^{n-1}(p_i^+ - p_i)r_{i,0} \tag{17}$$

This can be expressed in terms of the value of the independent buffer, p_0^+, by substituting Eq. (10)

$$p_0^+ = p_0 + \sum_{i=1}^{n-1}(r_{0,i}\frac{C_i}{C_0}p_0^+ - p_i)r_{i,0} \tag{18}$$

or [12],

$$p_0^+(1 - \sum_{i=1}^{n-1} r_{i,0}r_{0,i}\frac{C_i}{C_0}) = p_0 - \sum_{i=1}^{n-1} r_{i,0}p_i, \tag{19}$$

where $r_{i,0}$ is the gain of the route from buffer i to buffer 0 and C_i the buffer value of buffer i. Note that this may result in loss of energy to the environment [13].

To demonstrate, we derive $g_{\alpha_{01}}$. In the corresponding operational mode, α_{01}, there is dependency between three buffers, J, m_x, and m_y, with stored energy h_ω, p_x, and p_y, respectively. This represents the situation where the center of mass moves with velocity such that, in spite of the rotation around it, the rod-tip A does not move in the horizontal or vertical direction since it is in contact with the floor and stuck. Choosing J as the independent buffer, this results in two dependent buffers, m_x and m_y,

$$\begin{cases} a_{\delta,m_x} = p_{m_x}^+ - p_x, \; p_x^+ = r_{J,m_x}\frac{m_x}{J}h_\omega^+ \\ a_{\delta,m_y} = p_{m_y}^+ - p_y, \; p_y^+ = r_{J,m_y}\frac{m_y}{J}h_\omega^+. \end{cases} \tag{20}$$

So,

$$\sum_{buffers} = (r_{J,m_x}\frac{m_x}{J}h_\omega^+ - p_x)r_{m_x,J} + (r_{J,m_y}\frac{m_y}{J}h_\omega^+ - p_y)r_{m_y,J} \tag{21}$$

There are no δ-sources active upon switching so $\sum_{sources} = 0$. The complete expression for the independent energy, h_ω^+ now yields

$$h_\omega^+ = h_\omega + r_{m_x,J}r_{J,m_x}\frac{m_x}{J}h_\omega^+ - r_{m_x,J}p_x + r_{m_y,J}r_{J,m_y}\frac{m_y}{J}h_\omega^+ - r_{m_y,J}p_y. \tag{22}$$

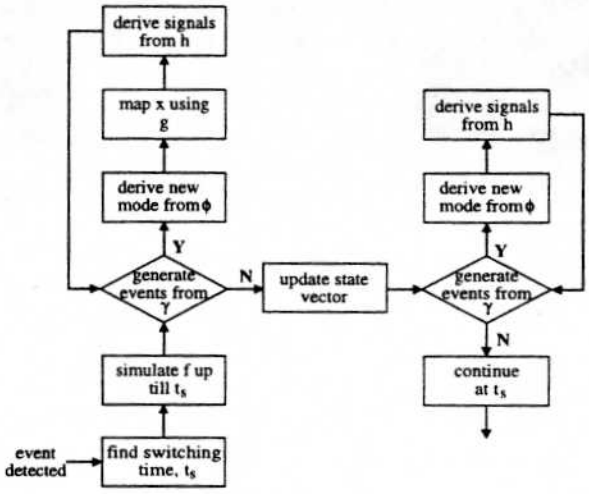

Figure 9: **Flow diagram of hybrid system simulation.**

This can be transformed into the state variables by the translations $h_\omega = J\omega$, $p_x = mv_x$, and $p_y = mv_y$ and by substitution of the gains of the respective routes, found by tracing power amplification along a route following causal strokes (Fig. 8),

$$\begin{cases} r_{m_x,J} = -lsin\theta, \; r_{J,m_x} = lsin\theta \\ r_{m_y,J} = lcos\theta, \; r_{J,m_y} = -lcos\theta, \end{cases} \tag{23}$$

which yields

$$\omega^+ = \frac{\omega J - ml(cos\theta v_y - sin\theta v_x)}{J + ml^2}. \tag{24}$$

Analogously, the state vector mapping can be derived for the other operational modes ($g_{\alpha_{01}}$ is given in Eq. (6))

$$g : \begin{cases} \alpha_{00} : \begin{cases} \omega^+ = \omega \\ v_x^+ = v_x \\ v_y^+ = v_y \end{cases} \\ \alpha_{11} : \begin{cases} \omega^+ = \frac{\omega J - mlcos\theta v_y}{J + ml^2 cos^2\theta} \\ v_x^+ = v_x \\ v_y^+ = -l\omega^+ cos\theta \end{cases} \\ \alpha_{21} : \begin{cases} \omega^+ = \frac{\omega J - ml(cos\theta - \mu sin\theta)v_y}{J + ml^2 cos\theta(cos\theta - \mu sin\theta)} \\ v_x^+ = -\mu(l\omega^+ cos\theta + v_y) + v_x \\ v_y^+ = -l\omega^+ cos\theta \end{cases} \end{cases} \tag{25}$$

In other work [13], we have used phase space analysis to verify the correctness of the model specifications, i.e., to ensure the given transition specifications will result in the model behavior *diverging in time*, a principle that has to be satisfied by all physical systems.

Hybrid System Simulation

The simulator operates in two modes:

1. continuous simulation in temporal intervals, and
2. discrete instantaneous configuration changes that occur at points in time.

The Simulation Scheme

Numerical simulation schemes like Euler and Runge-Kutta can be used for continuous behavior modes. Discrete events generated by γ trigger an event detection module to determine the switching time, t_s, within a margin of tolerance, ϵ (Fig. 9). The continuous field, f_{α_k}, computes $x_{\alpha_k}(t_s)$, then real time is suspended, and the meta-level control model, ϕ, generates the discrete state transition. The original continuous state vector is then transferred to the newly found model configuration using g, and this may trigger further events. The resulting model configuration is established, and $x_{\alpha_k}(t_s)$ is transferred to this model configuration. Again, discrete events may be generated and this process continues until no further transitions occur and the prior continuous system state is updated to $x_{\alpha_m}(t_s)$

Further events are generated when the state vector is updated and the *a priori* switching values change. This may cause a new series of configuration changes that are executed by the discrete model using the different h functions in each mode. There can be no more discontinuous changes in the state vector. In the new continuous mode α_n, f_{α_n}, defines the simulation from time t_s with initial vector $x_{\alpha_m}(t_s)$ $(= x_{\alpha_n}(t_s))$. This implements simulation of f_{α_m} at t_s as a point in time and allows an energy redistribution at a point specified by a function with discontinuities that are not simple. Note that the method only applies under the principle of temporal evolution of state.

Simulation of the Colliding Rod

To derive a numerical model of the continuous function, f, a 0-order, forward Euler, approximation is used. A forward Euler approximation is obtained by using $\dot{x} = \frac{x_{k+1}-x_k}{\Delta t}$, or $x_{k+1} = f\Delta t + x_k$. Derivatives that are part of expressions in f are replaced analogously. For example, $\dot{v}_x = l cos\theta\dot{\omega}$ becomes $v_{x,k+1} = \frac{l(cos\theta_{k+1}\omega_{k+1} - cos\theta_k\omega_k)}{\Delta t}\Delta t + v_{x,k}$, or, $v_{x,k+1} = l cos\theta_{k+1}\omega_{k+1} - l cos\theta_k\omega_k + v_{x,k} = l cos\theta_{k+1}\omega_{k+1}$. The expressions for θ_{k+1} and $y_{M,k+1}$ are uniform across configurations. These equations combined with the rest of the numerical model constitute the simulation of continuous behavior. The only other function of the analytical specification that has to be represented by a numerical equivalent is h:

$$h: \begin{cases} y_A = y_{M,k+1} - l sin\theta_{k+1} \\ v_{A,x} = v^+_{x,k+1} - l sin\theta_{k+1}\omega^+_{k+1} \\ p_{A,y} = m(v_{y,k+1} + l cos\theta_{k+1}\omega_{k+1}) \\ F_n = \begin{cases} 0 & \text{if } \alpha_{00} \\ m(\frac{v^+_{y,k+1} - v_{y,k}}{\Delta t} - a_g) & \text{otherwise} \end{cases} \\ F_{A,x} = \begin{cases} 0 & \text{if } \alpha_{00} \\ m\frac{v^+_{x,k+1} - v_{x,k}}{\Delta t} & \text{otherwise.} \end{cases} \end{cases} \quad (26)$$

The rest of the specifications are not temporal and can be directly used for simulation. Note that $v_{A,x}$ is computed based on *a posteriori* values whereas $p_{A,y}$ (an energy variable) is computed from *a priori* values.

A simulation run for one scenario is shown in Fig. 10. The rod falls down at a specific angle, hits the floor and moves into configuration α_{01}. Based on the state vector an immediate configuration change to mode α_{11} occurs. In this configuration the rod slides with a velocity that decreases in magnitude due to the friction force acting on the rod-tip. At one time, this velocity falls below a preset threshold value and if the state vector is such that the rod gets stuck without immediately satisfying the condition to slide, the system moves into mode α_{01}. In this mode, it is stuck and rotates around the point of contact until it falls flat on the floor.

Conclusions

This work demonstrates a powerful hybrid system modeling scheme that incorporates modeling abstractions and embedded discrete control of physical systems. The modeling formalism, based on a hybrid bond graph methodology, combines continuous bond graph models with local discrete finite state automata. The automata define ideal switching specifications imposed on bond graph junctions to create instantaneous model configuration changes. Configuration changes may result in discontinuities in system variables. The new state variables are then systematically derived using the principle of conservation of state combined with explicitly defined interactions with the environment. Global specifications are derived dynamically based on systematic principles of invariance of state, divergence of time, and temporal evolution of states. This simplifies the modeling task and truly demonstrates the use of *compositionality* in defining system models. This is in contrast with the approach by Alur *et al.* [1] which requires pre-defined global specifications of continuous system behavior in terms of differential equations. Furthermore, global knowledge in specifying discrete behavior is required to ensure no mythical modes exist. Also, unlike the hybrid bond graph modeling paradigm, there is no support for systematic modeling based on physical principles (e.g., conservation of state). The formal specifications are incorporated into a hybrid system simulation scheme that ensures the generation of correct system behavior.

In the past, qualitative reasoning schemes have focused on abstraction of the numerical properties of system behavior variables [6]. This has often led to underconstrained models resulting in an explosion of possible behaviors and the generation of physically inconsistent behaviors. Ours is a more systematic and encompassing approach to abstracting physical system models: (i) time scale abstraction, and (ii) ignoring parasitic parameter effects that often cause sharp nonlinearities. The result is a truly hybrid behavior

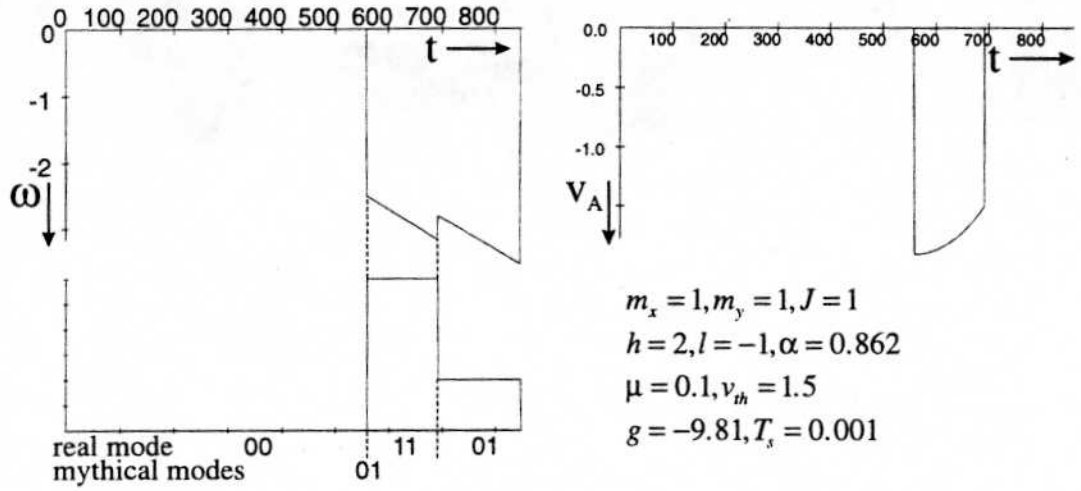

Figure 10: **Physically consistent simulation.**

generation scheme, where the abstractions result in discrete qualitative behaviors (mode and configuration changes), otherwise system behavior evolves continuously (and this can be simulated either by numeric or qualitative methodologies). Future work will be directed toward applying this methodology in embedded (computer-based) control of physical systems.

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. In J.W. Bakkers, C. Huizing, W.P. de Roeres, and G. Rozenberg (eds.), editors, *Proceedings of the 11th International Conference on Analysis and Optimization of Discrete Event Systems*, pages 331–351. Springer-Verlag, 1994. Lecture Notes in Control and Information Sciences 199.

[2] W. Borutzky. Exploiting differential algebraic system solvers in a novel simulation environment. *SAMS*, 17:165–178, 1995.

[3] P.C. Breedveld. Multibond graph elements in physical systems theory. *Jour. of the Franklin Institute*, 319(1/2):1–36, Jan./Feb. 1985.

[4] K. Denbigh. *The Principles of Chemical Equilibrium: With applications in chemistry and chemical engineering.* Cambridge University Press, London, New York, third edition, 1971.

[5] J. Montbrun-Di Filippo, M. Delgado, C. Brie, and H.M. Paynter. A survey of bond graphs: Theory, applications and programs. *Journal of the Franklin Institute*, 328(5/6):565–606, 1991.

[6] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge.* The MIT Press, Cambridge, MA, 1994.

[7] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *Systems Dynamics: A Unified Approach.* John Wiley, New York, 2nd ed., 1990.

[8] P. Lötstedt. Coulomb friction in two-dimensional rigid body systems. *Z. angew. Math. u. Mech.*, 61:605–615, 1981.

[9] P.J. Mosterman and G. Biswas. Modeling discontinuous behavior with hybrid bond graphs. In *1995 International Conference on Qualitative Reasoning*, pages 139–147, Amsterdam, May 1995. University of Amsterdam.

[10] P.J. Mosterman and G. Biswas. Analyzing discontinuities in physical system models. In Y. Iwasaki and A. Farquhar, eds., *Qualitative Reasoning: The Tenth Intl. Workshop*, pp. 164–173, Stanford Sierra Camp, CA, May 1996.

[11] P.J. Mosterman and G. Biswas. A formal hybrid modeling scheme for handling discontinuities in physical system models. *AAAI-96*, pp. 985–990, Portland, Oregon, 1996.

[12] P.J. Mosterman and G. Biswas. Hybrid modeling specifications for dynamic physical systems. *Intl. Conf. on Bond Graph Modeling and Simulation*, pp. 162-167, Phoenix, AZ, Jan. 1997.

[13] P.J. Mosterman and G. Biswas. A theory of discontinuities in dynamic physical systems. *Jour. of the Franklin Institute*, vol. 334B, no. 6, 1997.

[14] P.J. Mosterman, G. Biswas, and J. Sztipanovits. Hybrid modeling and verification of embedded control systems. to appear, *IFAC97*, Gent, Belgium, April 1997.

[15] Jan-Erik Strömberg, Jan Top, and Ulf Söderman. Variable causality in bond graphs caused by discrete effects. In *Proceedings of the International Conference on Bond Graph Modeling*, pages 115–119, San Diego, California, 1993.