Formal Specifications for Hybrid Dynamical Systems

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

Modeling abstractions in physical systems result in *hybrid models* which encompass continuous behaviors with discrete changes, causing discontinuities in system behavior generation which violate the physical laws of *conservation of energy* and *continuity of power*. This paper develops a formal specification for handling discrete model configuration changes at well-defined points in time, and a consistent transfer of the continuous system state from a previous model configuration to a new one 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.

1 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 fast nonlinear continuous behaviors. For efficient analysis, 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 occurs on 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 [5]. 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 [5]. 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 [5; 7], 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 [7]. 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.

2 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, 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 unambiguous, rigorous, and consistent.

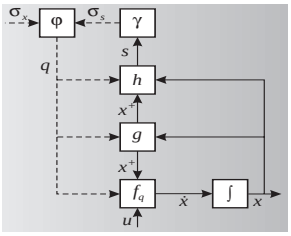

Figure 1: **A general hybrid system.**

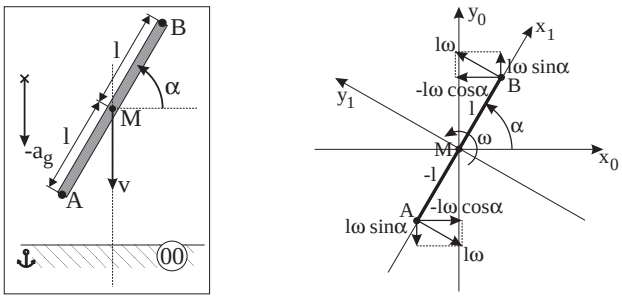

Figure 2: **A collision between a body and a floor.**

2.1 General Hybrid Dynamic System

The general architecture for a hybrid dynamic system model (Fig. 1) can be specified by the 9-tuple: $H =< I, \Sigma, \phi, X, U, f_\alpha, g, h, \gamma >$. 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.

2.2 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 $\dot{\omega} = 0, \dot{v}_x = 0, \dot{v}_y = a_g$, 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.

2.3 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 (a) *real* modes, where system behavior is governed by energy principles, and (b) *mythical* modes [5; 7], where the system behavior transitions are instantaneous. $\Sigma = \Sigma_s \times \Sigma_x$ captures the *event* set. Σ_s, is associated with closed loop control, and Σ_x is governed by external, open loop control signals. 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.

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

An example, adapted from [4], 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 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:

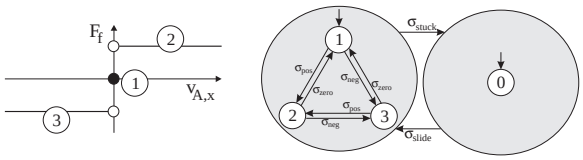

Figure 3: **Coulomb friction.**

$$\gamma : \begin{cases} |F_{A,x}| > \mu F_n \Rightarrow \sigma_{slide} \\ |v_{A,x}| \leq v_{th} \Rightarrow \sigma_{stuck}. \end{cases} \quad (1)$$

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:

$$\gamma : \begin{cases} v_{A,x} = 0 \Rightarrow \sigma_{zero} \\ v_{A,x} < 0 \Rightarrow \sigma_{pos} \\ v_{A,x} > 0 \Rightarrow \sigma_{neg}. \end{cases} \quad (2)$$

Since this behavior is piecewise continuous only *simple* behavior discontinuities occur at time points, which implies that operational modes have limit values at discontinuities. 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. 5. 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 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 $v_x = l\omega^+ sin\theta, v_y = -l\omega^+ cos\theta$. A detailed derivation (see [6]) yields the new state vector:

$$g_{\alpha_{01}} : \begin{cases} \omega^+ = \frac{1}{J+ml^2}(\omega J + ml(sin\theta v_x - cos\theta v_y)) \\ v_x^+ = l\omega^+ sin\theta \\ v_y^+ = -l\omega^+ cos\theta \end{cases} \quad (3)$$

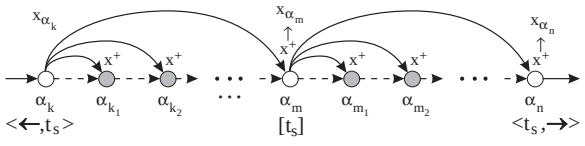

Figure 4: **System state is derived from the original state vector.**

3 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 further switching occurs. All the intermediate states traversed between two continuous modes are mythical [7]. The sequence of state and state vector changes is illustrated in Fig. 4. 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 (e.g, the point of contact in an elastic collision [7]). 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 (Fig. 4).

Consider the falling rod in Fig. 5. 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. 5 are mythical. They do not have physical meaning, therefore, no representation on the real time-line. However, they play

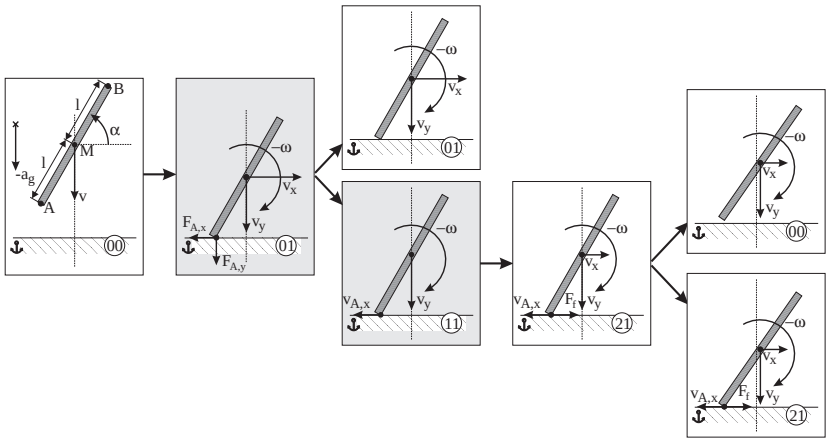

Figure 5: **Operational modes of falling rod.**

the role of transition points for locally defined switching functions.

3.1 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 [7] 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.

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.

3.2 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* [7].

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.

3.3 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

[7]. In previous work, we have developed a multiple energy phase space analysis that establishes divergence of time before simulation is performed [5].

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.

3.4 Implementation

In previous work[5; 7], we have developed a hybrid bond graph modeling methodology that uses an *ideal switching element*, whose on-off conditions are governed by finite state automata, to dynamically construct model configurations as its behavior evolves in time. The bond graph model of the idealized thin rod and idealized floor, and the fragments dynamically generated by simulation[6] are shown in Fig. 6. The rod is assumed to have three degrees of freedom with associated buffers: angular velocity ω (buffer J), and linear velocities v_x (buffer m_x) and v_y (buffer m_y). The relation between those velocities is modeled by a modulated transformer. Gravity is modeled by 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 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*. This bond graph is mode q_{00}. 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, $-lsin\alpha$. When $x_A = 0$ the rod collides with the floor, 0_C comes *on* and the model transitions

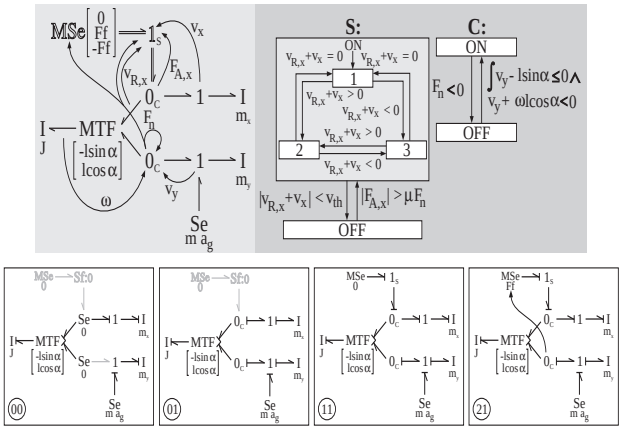

Figure 6: **Dynamically generated models.**

to mode q_{01}. If the rod-length and angle of collision are such that 1_S comes *on* (the model transitions into q_{11}), the rod begins to slide. The function, g, can be recalculated, and the piecewise continuous friction function may move into its F_f area, mode q_{21}. The hybrid bond graph approach provides a seamless integration of configuration changes based on local switches. Details of the derived continuous system and discrete control models appear in [6]. Other examples of hybrid bond graph models are discussed in [7].

4 Hybrid System Simulation

The simulator operates in two modes: (i) continuous simulation during intervals of operation, and (ii) discrete changes during configuration changes. Numerical simulation schemes like Euler and Runge-Kutta can be used for continuous operation but discrete events generated by γ triggers an event detection module to determine the switching time, t_s, within a margin of tolerance, ϵ (Fig 7). The continuous field, f_k, computes $x_k(t_s^-)$, 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 $x_k(t_s^-)$ is transferred to this model configuration. Again, discrete events may be generated and this process continues until no further transitions occur and the continuous system state is established as $x_m(t_s)$

If the new operational mode is valid for a point in time further events are generated till continuous simulation can be resumed for a model configuration valid over an interval of time. To detect possible configuration changes when the system evolves over an infinitesimal amount of time, the system, f_m, is considered continuous over an infinitesimal time, ϵ, i.e., no configuration changes are allowed. Simulation up till $t_s + \epsilon$ may cause a new

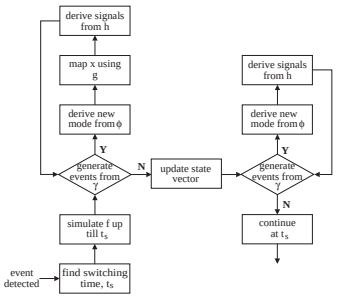

Figure 7: **Flow diagram of hybrid system simulation.**

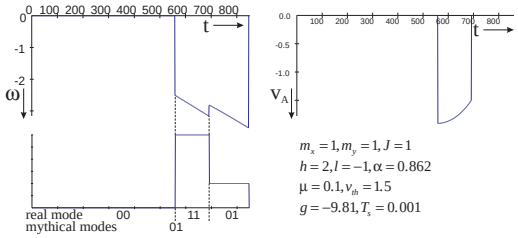

Figure 8: **Physically consistent simulation.**

series of configuration changes. These are executed by the discrete model using the h function. There can be no more discontinuous changes so the state vector does not have to be recomputed. In the new continuous mode q_n, f_n, defines the simulation from time t_s with initial vector $x_m(t_s)$. This implements simulation of f_m at t_s as a point in time and allows an energy redistribution that results from a function with discontinuities that are not simple. Note that the method only applies under the principle of temporal evolution of state.

A simulation run for one scenario is shown in Fig. 8. The rod falls down at a specific angle, hits the floor and moves into configuration 01. Based on the state vector an immediate configuration change to mode 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 q_{01}. In this mode, it is stuck and rotates around the point of contact until it falls flat on the floor.

5 Conclusions

This work demonstrates a powerful hybrid system modeling scheme that incorporates modeling abstractions and embedded discrete control of physical systems. Configuration changes governed by local automata may produce 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 state. 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 predefined 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. Overall, this is a systematic 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 generation scheme, where the abstractions result in discrete qualitative behaviors (mode and configuration changes), otherwise system behavior evolves continuously. Future work will be directed toward applying this methodology in embedded (computer-based) control of physical systems.

References

[1] R. Alur, et al., The algorithmic analysis of hybrid systems. J.W. Bakkers, C. Huizing, W.P. de Roeres, and G. Rozenberg (eds.), *proc. 11th Intl. Conf. on Analysis and Optimization of Discrete Event Systems*, Springer, pp. 331–351, 1994.

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

[5] 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.

[6] 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.

[7] P.J. Mosterman and G. Biswas. A theory of discontinuities in dynamic physical systems. *Jour. of the Franklin Institute*, 334B(6), 1997.