ELSEVIER

Artificial Intelligence 121 (2000) 171–209

Artificial Intelligence

www.elsevier.com/locate/artint

A comprehensive methodology for building hybrid models of physical systems

Pieter J. Mosterman[a,1], Gautam Biswas[b,*]

[a] *Institute of Robotics and Mechatronics, DLR Oberpfaffenhofen, P.O. Box 1116,*
D-82230 Wessling, Germany

[b] *Department of Electrical Engineering and Computer Science, Box 1679, Sta B, Vanderbilt University,*
Nashville, TN 37235, USA

Received 17 September 1999

Abstract

This paper describes a comprehensive and systematic framework for building mixed continuous/discrete, i.e., *hybrid* physical system models. Hybrid models are a natural representation for embedded systems (physical systems with digital controllers) and for complex physical systems whose behavior is simplified by introducing discrete transitions to replace fast, often nonlinear dynamics. In this paper we focus on two classes of abstraction mechanisms, viz., *time scale* and *parameter* abstractions, discuss their impact on building hybrid models, and then derive the transition semantics required to ensure that the derived models are consistent with physical system principles. The transition semantics are incorporated into a formal model representation language, which is used to derive a computational architecture for hybrid systems based on hybrid automata. This architecture forms the basis for a variety of hybrid simulation, analysis, and verification algorithms. A complex example of a colliding rod system demonstrates the application of our modeling framework. The divergence of time and behavior analysis principles are applied to ensure that physical principles are not violated in the definition of the discrete transition model. The overall goal is to use this framework as a basis for developing systematic compositional modeling and analysis schemes for hybrid modeling of physical systems. Preliminary attempts in this area are discussed, with thoughts on how to develop this into a more general methodology. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Hybrid modeling; Dynamic systems; Systematic model abstractions; Parameter abstraction; Time scale abstraction; Model verification

[*] Corresponding author. Gautam Biswas is supported by DARPA SEC contract number F33615-99-C-3611 and a grant from HP Labs, Palo Alto, CA. While on leave at the Knowledge Systems Lab, Stanford University, CA he was supported by NIST grant number 70NANB6H0075-05.

E-mail address: biswas@vuse.vanderbilt.edu (G. Biswas).

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

PII: S0004-3702(00)00032-1

1. Introduction

The need to achieve more optimal and reliable performance in complex physical systems that span from consumer products to aircraft and nuclear plants, while meeting more rigorous safety standards has led to a proliferation of computer-based techniques for modeling and analysis of these systems. The complexity of simulating, analyzing, and understanding system behavior is often handled by replacing nonlinear models by simpler component model constructs that together provide a good approximation to actual system behavior.

Traditional qualitative reasoning methods in Artificial Intelligence (e.g., [9,14,24]) employ abstractions in

(i) the quantity space of system variables, and

(ii) the functional relations among these variables to create simpler discretized piecewise behaviors that emphasize qualitative differences between different components of the behavior [9,22].

However, these methods, tend to generate a large number of behaviors, many of them spurious, thus limiting their usefulness in real world applications. A key to generating correct physical system behavior is to ensure that *energy conservation* and *continuity of power* constraints [51] are properly incorporated into the system model. System models expressed as ordinary differential equations (ODEs) or a combination of differential and algebraic equations (DAEs) generate quantitative behavior in real space. However, the continuous behavior may contain steep gradients, often of a nonlinear nature, and behavior generation algorithms have to deal with the resulting numerical stiffness which hampers systematic analysis and interpretation of the results.

Previous work in qualitative reasoning (e.g., [18,23]) has recognized that complex physical system behaviors often occur at different temporal and spatial scales. Iwasaki and Bhandari [18] have used relative magnitudes of coefficients in an influence matrix (i.e., the A matrix) of a linear system to determine “nearly decomposable” substructures. They assume a linear system whose influence matrix is nearly decomposable. Making a further assumption that the system is stable, they replace the tightly-coupled variables within each submatrix by an aggregated variable using eigenvector techniques, and reformulate the original ODE set into a smaller ODE set, that can be solved to derive the dynamic behavior of the system. This is similar to Kuipers’ [23] approach in QSIM [22] where the relations between tightly-coupled variables are replaced by instantaneous algebraic relations. Methods for simulating across multiple time scales in QSIM are formulated using a hierarchy of constraint networks. Our work generalizes and extends these approaches to linear and nonlinear systems. Analyzing behaviors of complex systems, we realize that small time constant effects cannot always be ignored in generating dynamic system behavior. We identify relatively small and large parameters in the physical system model and apply systematic techniques to abstract away their effects or condense their effect on gross behavior to occur at a point in time [32,34,37,43]. The resultant system model exhibits multiple modes of operation [50], each with simpler *piecewise* continuous behavior, but transitions between the modes may introduce discontinuous changes in the system variables.

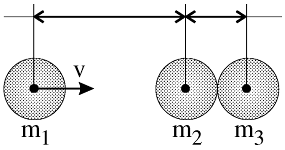


Fig. 1. Physical system with discontinuities.

Consider the system of three masses, shown in Fig. 1. Initially, masses m_2 and m_3 are touching and at rest, and m_1 is moving toward m_2 with a constant velocity v. For simplicity, we assume equal masses, i.e., $m_1 = m_2 = m_3$, and no frictional losses. When m_1 collides with m_2, there is a redistribution of momentum, which results in an abrupt change in the velocities of m_1 and m_2, i.e., $v_1 = 0$ and $v_2 = v$. The mass m_2 begins its continuous motion just after the abrupt change, but immediately collides with m_3. The phenomena repeats, with all of m_2's momentum being transferred to m_3, causing another discontinuous change in velocities. On a more detailed scale, if one takes into account the elasticity of the material of the bodies, the collision process can be described by more complex continuous models that describe the fast exchange of momentum in terms of energy conversion from kinetic to potential to kinetic energy at a very fine grained time scale. However, such models are hard to build and analyze. Even if one could build a good representative model of the detailed phenomenon, accurate estimation of the parameters of the model is a very difficult task. Creating a more abstract but simpler model of the collision phenomena produces a discontinuous change that occurs at a point in time. A convenient formalism for representing phenomena that combine discrete and continuous behaviors are *hybrid dynamic system* models [2,15,43]. These models generate continuous behavior interspersed with discrete events. A sequence of discontinuous changes may occur between two continuous behavior time intervals (for the colliding bodies, a collision between m_1 and m_2 is immediately followed by a collision between m_2 and m_3).

Hybrid behaviors may naturally occur in physical systems with embedded digital control, because the controller can force the system to operate in multiple configurations or modes. For example, the Airbus A-320 fly-by-wire system includes the *take-off, cruise,* and *go-around* operational modes [55]. Within each mode, system behavior evolves continuously but discrete mode changes dictated by a supervisory controller can occur at points in time, resulting in discontinuities in overall system behavior.

Our goal in this work is to develop comprehensive methodologies for analyzing behaviors of abstract models of complex dynamic physical systems that necessarily exhibit mixed continuous and discrete, i.e., hybrid behavior. There is a rich body of work in developing and analyzing hybrid system models [2,5,15,16,25,37,49,54]. We adopt the generic hybrid systems framework and develop a set of unambiguous and consistent physical system principles that govern the model formulation process and define the execution semantics for behavior generation. This paper establishes a formal specification language for building hybrid system models and defining execution semantics for behavior generation. This encompasses model definition as a combination of continuous differential equations and algebraic constraints that define discrete mode transitions in dynamic system behavior, and jumps in the state vector that may accompany the discrete transitions. A well-defined mathematical formulation (DAEs plus finite state machines) that includes the generation and analysis of Dirac pulses facilitates behavior analysis and the definition of simulation algorithms based on physical principles. Examples are presented to demonstrate the effectiveness of this approach. Other papers describe the underlying physics [37], its application to the development of simulation algorithms [36,42] and observer schemes to track dynamic behavior evolution [38].

2. Background

A physical system can be regarded as a configuration of connected physical elements that exhibit ideal reversible or ideal irreversible behavior [7]. Capacitive (e.g., spring, tank, electrical capacitor) and inductive (e.g., mass, fluid inertia, electrical inductor) processes are reversible in that they can store and release energy. Resistive processes (e.g., dashpots, viscous friction, electrical resistances) are irreversible in that they dissipate energy. The total *energy content* in a system at any point in time is the sum of the energy amounts stored in the reversible processes. Therefore, variables associated with these processes reflect the behavioral history of the system, and define the traditional notion of system *state*. Future behavior of the system is a function of its current state and input to the system from the present time. State changes are caused by energy exchange between the system components, expressed as *power*, the time derivative of flow of energy. The notions of system state, energy, and power are independent of the physical domain (e.g., mechanical, electrical, hydraulic, and thermal), and they form the basis for defining a set of mathematical equations that govern physical system behavior.

Differential equations are a common representation for continuous system behavior. The system is described by a minimal set of state variables (generalized momentum and displacement variables that are directly related to the energy content of the inductive and capacitive elements, respectively [6,51]), called the *state vector*, x, that completely captures its behavior history. System behavior over time is specified by a gradient of flow or field, f. Interaction with the environment is specified by *input* and *output* signals, u and y, respectively. The gradient of behavior, $\dot{x}$, is affected by the state and system input expressed as a set of differential equations, i.e., $\dot{x} = f(x, u)$ where f is time invariant. All other variables called *signals*, s, are algebraically related to the state and input variables by mathematical functions, $s = h(x, u)$.

Hybrid modeling paradigms [2,15,43] supplement continuous system descriptions by mechanisms that model discrete mode changes with associated discontinuities in the field description and the continuous state variables. Hybrid dynamic systems consist of three distinct subdomains:

- A *continuous* domain, T, with time, $t \in T$, as the special independent continuous variable. This defines the continuous time line.
- A *piecewise continuous* domain, V_α, that specifies variable flow, $x_\alpha(t)$, uniquely on the time-line.
- A *discrete* domain, I, that captures the operative piecewise continuous domain, V_α, corresponding to the system modes.

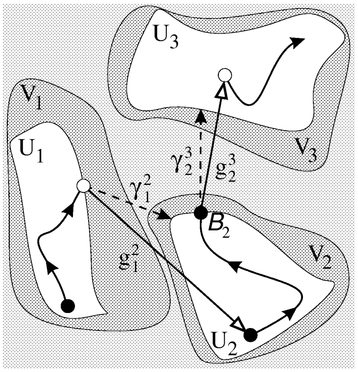


Fig. 2. A planar hybrid system.

We adopt notation similar to Guckenheimer and Johnson [15] and specify I to be a discrete indexing set, where $\alpha \in I$ represents the *mode* of the system. Each mode is defined on its domain V_α of $\mathbb{R}^n$. The behavior trajectory, $\mathcal{F}_\alpha$, is a continuous C^2 flow on a possibly open subset $U_\alpha \subset V_\alpha$ (shown for a planar hybrid system in Fig. 2). The flows constitute the piecewise continuous part of the hybrid system. System behavior at a point in time is specified by $x_\alpha(t)$, a location in mode α at time t. A discrete switching function γ_α^β is defined as a *threshold function* on V_α. If $\gamma_\alpha^\beta \leqslant 0$ in mode α the system transitions to β, and this is defined by the mapping $g_\alpha^\beta : \alpha \to \beta$. The piecewise continuous level curves $\gamma_\alpha^\beta = 0$, denoted as S_α^β, define transition boundaries. If a flow $\mathcal{F}_\alpha$ intersects the level curve, S_α^β, it contains the *boundary point*, $\mathcal{B}_\alpha$ (see Fig. 2, where the flow includes boundary point $\mathcal{B}_2$). In summary, a hybrid dynamic system is defined by the 5-tuple [2]

$$H = \langle I, X_\alpha, f_\alpha, \gamma_\alpha^\beta, g_\alpha^\beta \rangle. \tag{1}$$

A *trajectory* in the system starts at an initial point $x_{\alpha_1}(t_0)$, and if $\gamma_{\alpha_1}^{\alpha_2} > 0\ \forall \alpha_2 \in I$, it continues to flow in α_1 specified by $\mathcal{F}_{\alpha_1}$ until the minimal time t_s at which $\gamma_{\alpha_1}^{\alpha_2}(x_{\alpha_1}(t)) = 0$ for some α_2. Computing $x_{\alpha_1}(t_s^-) = \lim_{t \uparrow t_s} \mathcal{F}_{\alpha_1}(t)$ the transformation $g_{\alpha_1}^{\alpha_2}$ takes the trajectory from $x_{\alpha_1}(t_s^-) \in V_{\alpha_1}$ to $x_{\alpha_2}(t_s) \in V_{\alpha_2}$. The point $x_{\alpha_2}(t_s) = g_{\alpha_1}^{\alpha_2}(x_{\alpha_1}(t_s^-))$ is regarded as the new initial point in mode α_2.

If there exists $\alpha_3 \in I$, such that $\gamma_{\alpha_2}^{\alpha_3}(x_{\alpha_2}(t_s)) \leqslant 0$, the trajectory is immediately transferred to $g_{\alpha_2}^{\alpha_3}(x_{\alpha_2}(t_s)) \in V_{\alpha_3}$. A characteristic of hybrid systems is the possibility of a number of these immediate changes occurring without an intermediate flow of continuous behavior [1,15,32,50]. In general, this situation occurs if $\gamma_{\alpha_k}^{\alpha_{k+1}}$ transports a trajectory to α_{k+1}, and the initial point is transported by $g_{\alpha_k}^{\alpha_{k+1}}$ to a value that results in $\gamma_{\alpha_{k+1}}^{\alpha_{k+2}} \leqslant 0$, i.e., $g_{\alpha_k}^{\alpha_{k+1}}(x_{\alpha_k}) \notin U_{\alpha_{k+1}}$, and another mode α_{k+2} is instantaneously arrived at. These immediate transitions continue till a mode α_m is arrived at where the initial point is within U_{α_m}. To deal with these sequences of transitions, Alur et al. [1,2], Guckenheimer and Johnson [15]

[2] Guckenheimer and Johnson refer to the respective parts as $\langle V_\alpha, X_\alpha, \mathcal{F}_\alpha, h_\alpha^\beta, \mathcal{T}_\alpha^\beta \rangle$ [15].

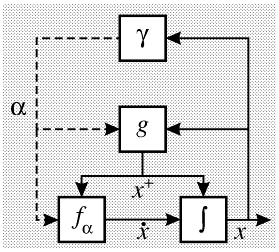


Fig. 3. A model of hybrid dynamic systems.

and Deshpande and Varaiya [11] propose model semantics based on temporal sequences of abutting intervals

$$\underbrace{[t_1 \quad t_2]}_{\alpha_1} \rightarrow \underbrace{[t_2 \quad t_3]}_{\alpha_2} \rightarrow \cdots \rightarrow \underbrace{[t_m \quad t_{m+1}]}_{\alpha_m} \tag{2}$$

with behavior that satisfies the following sequence:

$$\underbrace{\begin{cases} x = x_{\alpha_1} \\ \dot{x} = f_{\alpha_1}(x,t) \end{cases}}_{\alpha_1} \overset{\gamma_{\alpha_1}^{\alpha_2}(x)}{\underset{}{\xrightarrow{x_{\alpha_2}=g_{\alpha_1}^{\alpha_2}(x)}}} \underbrace{\begin{cases} x = x_{\alpha_2} \\ \dot{x} = f_{\alpha_2}(x,t) \end{cases}}_{\alpha_2} \overset{\gamma_{\alpha_2}^{\alpha_3}(x)}{\xrightarrow{x_{\alpha_3}=g_{\alpha_2}^{\alpha_3}(x)}} \cdots$$

$$\overset{\gamma_{\alpha_{m-1}}^{\alpha_m}(x)}{\xrightarrow{x_{\alpha_m}=g_{\alpha_{m-1}}^{\alpha_m}(x)}} \underbrace{\begin{cases} x = x_{\alpha_m} \\ \dot{x} = f_{\alpha_m}(x,t) \end{cases}}_{\alpha_m}. \tag{3}$$

Fig. 3 shows a schematic representation of the semantics that produce a sequence of transitions of this form.

This paper develops the hybrid modeling paradigm for dynamic physical systems by introducing constraints on the state vector function g_α^β during discrete transitions [36]. Furthermore, it modifies the definition of γ_α^β to closely match requirements of object oriented modeling methods [8,10,20].

3. Hybrid modeling of physical systems

In previous work we have applied systematic modeling principles to derive hybrid models in a number of different physical domains, e.g., the freewheeling diode circuit [32, 37,43] in the electrical domain, braking and clutch mechanisms [42] in the mechanical domain, the secondary sodium cooling loop [43] in the combined thermal and fluid domains, and the elevator control system of aircraft [40] in the combined mechanical and fluid domains. The common theme in all of the above work was the abstraction of parasitic elastic, inertial, and dissipative effects, so that complex phenomena at fast time scales could be condensed into discontinuous changes to reduce the complexity in analyzing model behavior. This paper integrates all the past work into a comprehensive and formal approach for specifying hybrid models of dynamic physical systems. The abstracted phenomena, that occur as discontinuous mode changes are represented by discrete switching mechanisms implemented as finite state machines that are then integrated with the continuous ordinary differential equation (ODE) models to generate piecewise continuous behaviors [30,37]. System topology in a mode is generated dynamically, and the switching specifications are derived as inequality constraints on state variables.

This section develops the systematic modeling principles into a formal mathematical framework to facilitate the analysis of hybrid systems behavior. We first discuss how conservation principles can be applied to generate physically consistent behavior. The mechanisms for parameter and time scale abstraction are introduced. The abstractions introduce discrete switching conditions in the behavior trajectory. We show that the transition conditions that result from parameter abstractions have to be in terms of *a posteriori* state vector values (the final value around a discontinuous change), and the transition conditions that result from time scale abstraction have to be in terms of *a priori* state vector values, (the initial value around a discontinuous change). These results are then incorporated into the specifications for the formal mathematical model of hybrid systems.

3.1. Conservation of state

Physical system behavior is governed by macroscopic conservation laws. For example, the vector components of the momentum of a number of bodies that collide is conserved. The overall system momentum is unchanged by the collision process even though there may be instantaneous dissipation due to frictional effects in a nonideal elastic collision resulting in the loss of kinetic energy. Similarly, the charge in an electrical circuit is conserved, but instantaneous loss of electrical energy may occur. In general, except for energy which may be instantaneously dissipated [12,44],[3] the extensive variables that define the physical state of a system are conserved. This conservation law needs to be preserved when generating behavior from hybrid physical system models and can be derived from the system model by integrating the system of differential equations [31].

Consider the two colliding bodies in Fig. 4, where m_1 moves toward m_2 with initial velocity, $v_1 = v$ and m_2 is at rest (mode α_0). This can be described by the system of differential equations using Newton's second law:

$$f^d\colon \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, \tag{4}$$

where F_1 and F_2 represent the forces on masses m_1 and m_2, respectively. Furthermore, the following algebraic constraints hold:

$$f^a\colon \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5}$$

[3] Note that this does not mean there is a loss of energy, but that free energy is instantaneously dissipated.

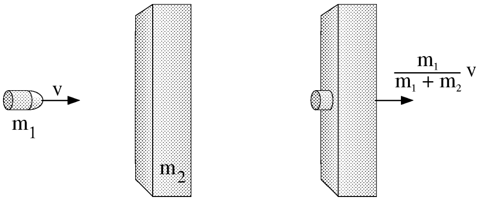


Fig. 4. Nonelastic collision between a bullet and a piece of wood.

indicating the two masses are moving with constant velocities.

For the point masses, collision occurs when $x_1 \geqslant x_2$, and the system transitions into mode α_1 where the momentum is instantaneously redistributed between m_1 and m_2 resulting in

$$v_2^+ - v_1^+ = 0, \tag{6}$$

where v_1^+ and v_2^+ represent the instantaneous change in velocity of masses m_1 and m_2, respectively, after the collision. The values of v_1 and v_2 are known at the point $x_1 \geqslant x_2$ becomes true. At this point, the continuous behavior evolution of v_1 and v_2 stops. However, both v_1^+ and v_2^+ are unknown, and cannot be solved for with one equation. It is also known that in the collision mode, α_1, Newton's third law applies, and $F_1 = -F_2 \neq 0$. These constraints can be added to the system of equations by replacing Eq. (5) with

$$f^a\colon \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{7}$$

Combining Eq. (4) and Newton's third law yields $m_1\dot{v}_1 = -m_2\dot{v}_2$. This can be integrated over an infinitesimal time interval $[t, t^+]$ to express the instantaneous effects around the discontinuous change, i.e.,

$$m_1(v_1^+ - v_1) = -m_2(v_2^+ - v_2). \tag{8}$$

This embodies the conservation of momentum constraint generated from physical principles. Eqs. (6) and (8) together provide a unique solution for v_1^+ and v_2^+:

$$g^{\alpha_1}\colon \begin{bmatrix} v_1^+ \\ v_2^+ \end{bmatrix} = \begin{bmatrix} \dfrac{m_1}{m_1+m_2}v \\ \dfrac{m_1}{m_1+m_2}v \end{bmatrix}.$$

There may be instantaneous dissipation when discontinuities occur resulting in loss of kinetic energy in the system. In case of a nonelastic collision, free energy in the system before collision, $\frac{1}{2}m_1v^2$, exceeds the free energy in the system after collision, $\frac{1}{2}\frac{m_1^2}{m_1+m_2}v^2$. In general, the derived conservation laws are the instantaneous equivalent of the continuous dynamics in a new active mode. This corresponds to a projection onto a manifold defined

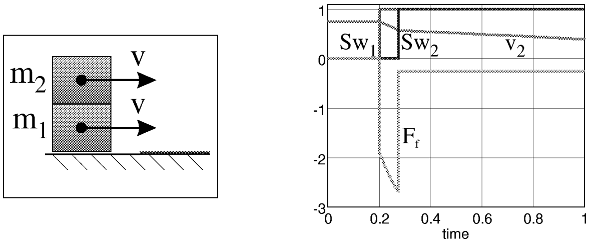


Fig. 5. A body that starts to slide.

by the algebraic constraints that replace the fast continuous behavior. The projection takes place in the *impulse space* or *jump space* and causes the system behavior or state space trajectories to include discontinuous jumps onto the manifold [17,31,56].

3.2. Abstractions in physical system models

The macroscopic view of the physical world is that its behavior is continuous and all physical system behavior is governed by the principle of continuity of power [51]. However, when one looks at complex physical systems, their behaviors combine phenomena at multiple temporal and spatial scales. Depending on the granularity of the observations and the detail at which analysis is performed, certain aspects of a system's behavior may appear to be instantaneous. Consider the example of two bodies one on top of the other, shown in Fig. 5, sliding towards a rough surface that has a very high coefficient of friction. When m_1 reaches the rough surface, shown as switch variable Sw_1 going from 0 to 1 in the graph, the frictional force opposing the motion is large enough to bring it to rest almost immediately. The resulting deceleration force F_f also acts on mass m_2. If the Coulomb friction force between m_1 and m_2 is exceeded, i.e., $|F_f| > F_{th}$, m_2 starts to slide on top of m_1. This new mode is indicated by discrete variable $Sw_2 = 1$ in the graph. Whether $|F_f|$ exceeds F_{th} depends on the initial velocity of the combined mass system. If the initial velocity is sufficiently low, m_1 and m_2 will come to rest together without exceeding the breakaway force, F_{th}. Otherwise, after a very short period of initial deceleration, m_2 starts to slide on top of m_1 with a constant friction force, F_{th}, acting between the two.

In many situations, when analyzing complex models of much larger systems, one would like to avoid such detailed analysis of phenomena and the corresponding computational complexity in behavior generation. In such cases, it may suffice to model the transition that occurs when $Sw_1 = 1$ by an instantaneous change where m_1's velocity goes to zero, and m_2 continues to move with a finite velocity. In effect, what this avoids is generating behavior that captures the details of the quick build-up of the frictional force. This is referred to as a *parameter* abstraction.

Definition 1 (*Parameter abstraction*). Parameter abstractions remove small and large, often parasitic, dissipation and storage parameters from the system model causing discontinuous changes in system behavior.

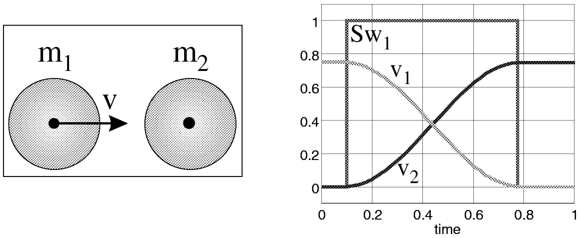


Fig. 6. A collision between two bodies with equal mass.

Another situation where abstraction of fast continuous transient behavior may be applied is the example of two colliding bodies shown in Fig. 6. Upon collision, represented by the discrete change $Sw_1 = 1$, small elasticity effects in m_1 and m_2 become active and store the kinetic energy of m_1 as elastic (potential) energy. Spring-like or elastic effects cause the stored energy to be returned as kinetic energy in a very small time interval. The exchange of momentum as a continuous transient was illustrated for equal masses earlier.

This example again illustrates that complex phenomena over small time intervals (a collision in this case) can be replaced by a more abstract model where the exchange of momentum caused by elasticity effects is modeled as an instantaneous change. In this case, the effects of the elasticity parameters are not abstracted away. Their end effect is explicitly modeled as a discontinuous change in the velocities at the point of collision. This is referred to as a *time scale* abstraction.

Definition 2 (*Time scale abstraction*). Time scale abstractions compress behaviors that occur on a small time scale relative to the primary behavior(s) of interest to explicit discontinuous changes at a point in time.

3.3. The different semantics

Both parameter and time scale abstractions provide mechanisms for reducing model complexity by eliminating higher order derivatives and nonlinear effects that cause fast transients in the ODE formulation of the system model. However, applying these abstractions cause discontinuous changes in the system variables at points in time. Systematic incorporation of the abstractions into the modeling mechanism requires careful analysis of the underlying physical nature of these continuous transients to ensure that the behaviors generated by the simplified hybrid models correspond to the real system behavior. The analysis reveals the need for different semantic specifications for parameter and time scale abstractions. This section demonstrates how these semantics translate to the formal specifications for defining hybrid system behavior.

3.3.1. Parameter abstraction

Consider the two bodies on top of one another in Fig. 5 that slide towards a rough surface as discussed in Section 3.2. If the detailed continuous transients on contact with the rough surface are abstracted away, the resultant hybrid system model is made up of three modes

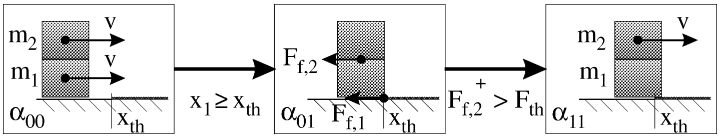


Fig. 7. The top body may continue to slide.

as illustrated in Fig. 7. In the initial mode, α_{00}, the combined masses m_1 and m_2 slide toward the rough surface.

When m_1 makes contact with the rough surface at $x_1 \geqslant x_{th}$, the model switches mode to α_{01}. In this mode, it has to be determined if m_2 comes to rest with m_1, or whether the resultant force on m_2 is large enough to cause it to slide on m_1. This requires computation of the force $F_{f,2}$. Since the impact is idealized, it occurs at a point in time causing the velocity of the masses to change discontinuously. The resultant force on m_2 is an *impulse* [4], $P_{f,2}$, which is represented mathematically as a Dirac function (δ) that occurs at the time point of impact. Its area is determined by the velocities immediately prior to ($v_2 = v$) and after ($v_2^+ = 0$) the impact, i.e., $v_2^+ - v_2$.

$P_{f,2}$ being an impulse, its value cannot be directly compared against a threshold force to determine whether m_2 slides or not. [4] Instead, a linear approximation is applied to compute the resultant force from the change in velocity. This requires us to go back to the more detailed model which includes the small elasticity (modeled as spring capacitance, C) and friction parameters (modeled as damper resistance, R) to compute a more realistic value of the maximum force generated. This results in the system of equations

$$\begin{bmatrix} \dot{p}_1 \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\dfrac{R}{m_1+m_2} & -\dfrac{m_1}{C(m_1+m_2)} \\ \dfrac{1}{m_1} & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ q \end{bmatrix}. \tag{9}$$

From this, the detailed behavior of the force $F_{f,2}$ can be computed as

$$F_{f,2} = \frac{Rm_2}{m_1(m_1+m_2)} p_1 - \left(\frac{m_1}{C(m_1+m_2)} - \frac{1}{C} \right) q, \tag{10}$$

where p_1 is the momentum of mass m_1 and q the displacement associated with the small spring effect. In case of complex eigenvalues, $\lambda_{1,2} = \lambda_r \pm \mathrm{i}\lambda_i$, solving for p_1 and q yields ($q(0) = 0$)

$$p_1 = p_1(0)\mathrm{e}^{-\lambda_r t} \left(\cos(\lambda_i t) + \frac{\lambda_r - \frac{R}{m_1+m_2}}{\lambda_i} \sin(\lambda_i t) \right) \tag{11}$$

and

$$q(t) = p_1(0)\mathrm{e}^{-\lambda_r t} \frac{1}{\lambda_i m_1} \sin(\lambda_i t). \tag{12}$$

[4] Since an impulsive force has infinite magnitude, a direct comparison would imply that $P_{f,2}$ always exceeds the threshold frictional force, implying the mass m_2 will start sliding.

Substitution in Eq. (10) yields an expression for $F_{f,2}$ that can be approximated by a Taylor series expansion [41]. For example, in the reduced order model the $F_{f,2} = m_2\dot{v}_2$ equation can be replaced by a linear approximation $F_{f,2} = K_d v_1(0)$, where $m_1 v_1(0) = p_1(0)$ and K_d embodies the parameters R, C, m_1, and m_2.

This value of $F_{f,2}$ can then be compared against the breakaway force value, F_{th}. If the detailed temporal behavior is such that the breakaway value is exceeded, the α_{11} mode of the hybrid model is activated. In this continuous mode, m_2 slides with initial velocity that equals the final velocity when α_{00} was departed and with a constant frictional force acting on it. Therefore, the velocity in the intermediate mode does not affect the velocity of m_2 in α_{11}, the sliding mode. In such a case, this mode, α_{01}, is called a *mythical mode* [32,50].

Definition 3 (*Mythical mode*). When parameters affecting the fast behaviors of a system are abstracted away and replaced by discrete transitions, they may result in a behavior trajectory that goes through discrete modes of behavior that have no existence on the real time line. These modes are called mythical.

Principle 1 (Invariance of state). *Mythical modes have no effect on the system state vector.*

This follows from the *invariance of state* lemma presented in [43]. The proof is presented in [35].

Formally, the consecutive mode switch to α_{11} has to occur before the state vector is updated to its *a posteriori* values, $x = x^+$. Mathematically this can be represented as

$$\underbrace{\begin{cases} x^+ = g^{\alpha_1}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_1}(x,t) \end{cases}}_{\alpha_1} \xrightarrow{\gamma_{\alpha_1}^{\alpha_2}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_2}(x) \\ \\ \\ \end{cases}}_{\alpha_2} \xrightarrow{\gamma_{\alpha_2}^{\alpha_3}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_3}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_3}(x,t) \end{cases}}_{\alpha_3}, \tag{13}$$

where α_2 is a mythical mode. In this example, it is clear that determining whether a switch occurs has to be based on x^+ rather than x, as shown in Fig. 7.

3.3.2. Time scale abstraction

Consider the elastic collision of two bodies shown in Fig. 8. The mass m_1 moves with initial velocity v towards the mass m_2, that is at rest. As discussed earlier, a detailed analysis of the collision indicates that elasticity effects store the initial kinetic energy as potential (elastic) energy, and then return this energy as kinetic energy over a very small time interval. Often, the time scale of this phenomenon is very small compared to the behavior of interest, so it can be modeled as an instantaneous change at a point in time governed by two equations:

(i) Newton's collision rule [4]

$$v_1^+ - v_2^+ = -\varepsilon(v_1 - v_2), \tag{14}$$

where the coefficient of restitution, ε, determines the amount of energy dissipated upon collision, and

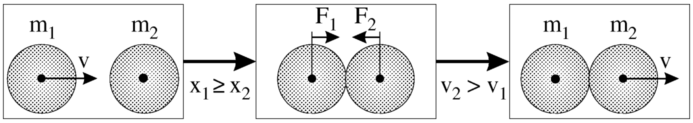


Fig. 8. A collision between two bodies.

(ii) the conservation of momentum principle (discussed in Section 3.1):

$$m_1 v_1^+ + m_2 v_2^+ = m_1 v_1 + m_2 v_2. \tag{15}$$

If the collision is perfectly elastic, $\varepsilon = 1$, and with $m_1 = m_2$ this yields

$$\begin{cases} v_1^+ = v_2, \\ v_2^+ = v_1, \end{cases} \tag{16}$$

which results in $v_1^+ = 0$ and $v_2^+ = v$. As soon as the new velocities are computed, the bodies disconnect. In this case, the new configuration is reached after the state vector is updated. Otherwise, m_1 would still have velocity v, and m_2 would be at rest, and the collision process would repeat. Note that the switching specifications have to ensure that the *collide* mode is departed immediately after the state vector, x, is updated to its *a posteriori* values, x^+, $x = x^+$. This is implemented by the $v_2 > v_1$ constraint. Otherwise, the collision effect in Eq. (16) would be executed again, causing the velocities of m_1 and m_2 to revert to their values immediately before collision, and the process would repeat *ad infinitum*. Mathematically this is represented as

$$\underbrace{\begin{cases} x^+ = g^{\alpha_1}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_1}(x,t) \end{cases}}_{\alpha_1} \xrightarrow{\gamma_{\alpha_1}^{\alpha_2}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_2}(x) \\ x = x^+ \\ \end{cases}}_{\alpha_2} \xrightarrow{\gamma_{\alpha_2}^{\alpha_3}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_3}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_3}(x,t) \end{cases}}_{\alpha_3}, \tag{17}$$

where α_2 is a *pinnacle*.

Definition 4 (*Pinnacle*). An explicitly defined point on the time line that is not part of a continuous behavior interval, V_α, but embodies a change in the continuous system state vector is called a pinnacle, $\mathcal{P}_\alpha$.

Fig. 8 illustrates that switching specifications have to be in terms of *a priori* state variable values.

3.3.3. Summary

The two types of abstraction have a distinctly different effect on how to formulate switching specifications and introduce the fundamentally different behavior between mythical modes and pinnacles. As illustrated, time scale abstraction collapses behavior during small intervals into points, and the switching model uses *a priori* state values.

In contrast, parameter abstraction abstracts away complex nonlinear behaviors, that are modeled by switching conditions based on *a posteriori* state values computed by g_α^β. The mythical modes that result from these conditions are modeling artifacts that have no real representation, and, therefore, do not affect the state vector, x. This is called the principle of *invariance of state* [35,43].

Since mythical modes have no effect on system state, they may be replaced by direct transitions to the final real mode (either a pinnacle, $\mathcal{P}_\alpha$, or a continuous mode, $\mathcal{F}_\alpha$) in a simulation or behavior generation algorithm. However, deriving these direct transitions before hand may require considerable computational effort, and because they describe a global phenomena, they will have to be pre-compiled for every possible local change derived from the model. The complexity of precomputing transitions past mythical modes is further compounded by the fact that a mode may be labeled mythical for certain state vector values, and not for others. Therefore, replacing mythical modes by direct transitions requires the inclusion of ranges of state vector values.

A more pragmatic approach is to incorporate systematic techniques in a compositional modeling formalism to deal with these artifacts. Furthermore, translating a system model into a model where only *a priori* state variable values are used complicates the model verification task considerably. If *a posteriori* values are used, invariance of state can be conveniently applied for model verification purposes [32,33,43].

3.4. A formal representation

We now develop a mathematical model that embodies the physical abstraction semantics discussed above. At the core of this specification is the establishment of the switching function, γ_α^β, that depends on the state vector values x_α, prior to the discontinuous change, and the state vector values x_α^+ immediately after the discontinuous change. The semantics of a mode transition (α_k to α_i) are specified by the recursive relation between $\gamma_{\alpha_k}^{\alpha_i}$ and the function that specifies the change in the state vector across mode transitions, $g_{\alpha_k}^{\alpha_i}$,

$$\begin{cases} x_{\alpha_k}^+ = g_{\alpha_k}^{\alpha_i}(x_{\alpha_k}),\\ \gamma_{\alpha_i}^{\alpha_{i+1}}(x_{\alpha_k}, x_{\alpha_k}^+) \leqslant 0. \end{cases} \tag{18}$$

Note the α_k subscript in $g_{\alpha_k}^{\alpha_i}$ and its argument x_{α_k}. In physical systems, the new continuous state only depends on its previous continuous state and the present mode, but not on the previous mode. Therefore, the α_k dependency of $g_{\alpha_k}^{\alpha_i}$ can be removed. The general form of the resulting sequence of mode transitions is given by:

$$\underbrace{\begin{cases} x^+ = g^{\alpha_1}(x)\\ x = x^+\\ \dot{x} = f_{\alpha_1}(x,t) \end{cases}}_{\alpha_1} \xrightarrow{\gamma_{\alpha_1}^{\alpha_2}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_2}(x)\\ x = x^+\\ \dot{x} = f_{\alpha_2}(x,t) \end{cases}}_{\alpha_2} \xrightarrow{\gamma_{\alpha_2}^{\alpha_3}(x,x^+)} \cdots$$

$$\xrightarrow{\gamma_{\alpha_{m-1}}^{\alpha_m}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_m}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_m}(x,t) \end{cases}}_{\alpha_m}. \tag{19}$$

When comparing with the general sequence in Eq. (3), one notes that the g_α^β operation can be linked either to the transition or the new operational mode. This is equivalent to the difference between Moore and Mealy state machines, and does not result in differences as far as behavior generation is concerned [45]. We choose to follow the Moore-type specification and associate g_α^β with modes for two reasons:

(i) a change in the state vector values implies an energy redistribution in the system, therefore, it should correspond to a physical mode of operation, and

(ii) for mythical mode transitions, the state vector x remains unchanged, therefore, it makes it easier to translate these specifications into a simulation algorithm if g_α^β is associated with a mode rather than a transition.

In the sequence in Eq. (19), a mode α may be departed when any of the three assignments or updates are executed:

(1) computing x^+ from x, and

(2) updating the state vector, x, across the discrete transition,

(3) evolving the state vector, x, in the continuous mode.

The difference between the general form of Eq. (3) and Eq. (19) can be attributed to the application of parameter and time scale abstractions to physical system models where (x, x^+) becomes the argument to γ_α^β, and this can be justified by physical system principles. The computational model corresponding to the use of (x, x^+) is illustrated in Fig. 9. When compared to the model in Fig. 3 (corresponding to Eq. (3)), there is additional feedback, x^+ into γ, which introduces a loop between g and γ. In other words, updating the state vector from x to x^+ may trigger another mode transition resulting in a new mode α, and a new state vector, x^+, and this sequence can repeat.

Overall, three transition specifications follow from the mathematical model in Fig. 9 corresponding to the mode departures listed above and illustrated in Fig. 10 [36].

(a) *Mythical mode:* This occurs when $x^+ = g^{\alpha_i}(x)$ leads to $\gamma_{\alpha_i}^{\alpha_{i+1}}(x, x^+) \leqslant 0$. The immediate transition to mode α_{i+1}, caused by x^+ by-passes the *integrator* ($\int$) and the state vector, x, is unchanged through the transition.

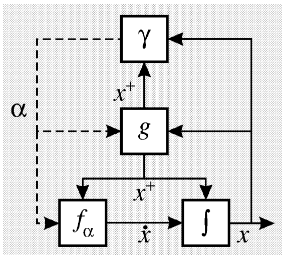
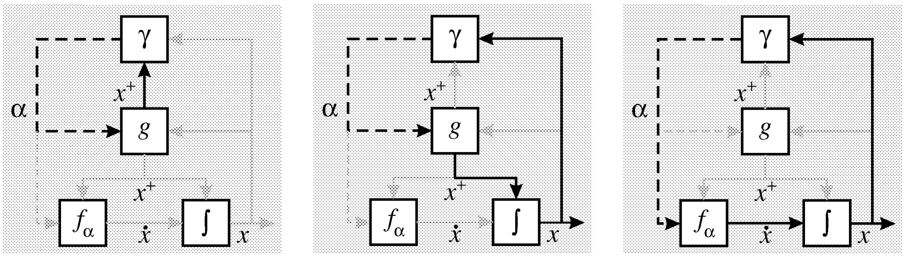


Fig. 9. A mathematical model based on physical semantics.



(a) (b) (c)

Fig. 10. Classes of mode transitions.

(b) *Pinnacle:* This occurs when the assignment of $x = x^+$ is made, and the update of x by the integrator causes an immediate mode transition because $\gamma_{\alpha_i}^{\alpha_{i+1}}(x, x^+) \leqslant 0$. Therefore, mode α_i only exists at a point in time.

(c) *Continuous mode:* In this mode update of the state vector using $\dot{x} = f(x,t)$ results in $\gamma_{\alpha_i}^{\alpha_{i+1}}(x, x^+) > 0$. System behavior evolves continuously till a point in time when $\gamma_{\alpha_i}^{\alpha_{i+1}}(x, x^+) \leqslant 0$, indicating the mode is departed.

Pinnacles and continuous modes are *real* modes because the state vector, x, that defines system behavior on the real time line, is directly affected within that mode of operation.

4. Behavior analysis

A number of important issues arise in analyzing hybrid system models that impact the validity of system behavior in terms of physical principles. The two primary principles discussed in this section are:

(i) *divergence of time*, to ensure the evolution of physical system behavior across discrete mode transitions on the time line, and

(ii) *temporal evolution of state* to ensure systematic update of the state vector about the point of discontinuity requiring state variable values to be continuous in left-closed intervals.

4.1. Divergence of time

Mode transitions defined by Eq. (18) may generate a trajectory that goes through a sequence of mythical modes before a new real mode is reached where system behavior resumes continuous evolution on the time line. In generating this sequence, if a mode can be reached more than once, the implication is that the trajectory can end up in a *loop* of discrete changes. Therefore, this behavior trajectory may not progress to a real mode, which implies that the system behavior evolution stops in time. This conflicts with physical system principles because their behaviors always evolve or diverge in time.

Principle 2 (Divergence of time). *Physical system behavior must evolve in time. Therefore, hybrid models of physical systems cannot include loops of discrete changes.*

Corollary. *Discrete mode changes in a hybrid system model have to terminate in a real system mode where behavior evolves in real time.*

To illustrate the principle of *divergence of time*, consider the elastic collision between the two bodies in Fig. 6 where m_1 with initial velocity, $v_1 = v$ moves towards m_2, that is at rest. The collision event for the point masses is defined as

$$\gamma_{\alpha_0}^{\alpha_1}\colon\ x_2 - x_1 \leqslant 0 \Rightarrow \sigma_{collide}. \tag{20}$$

As discussed earlier, if $m_1 = m_2$ this event results in instantaneous transfer of all of m_1's momentum to m_2.

Combining Newton's collision rule in Eq. (14) and the physical conservation of momentum constraint in Eq. (15), and making the assumption $m_1 = m_2$, we obtain the g function for updating the state vector:

$$g^{\alpha_1}\colon\ \begin{bmatrix} v_1^+ \\ v_2^+ \end{bmatrix} = \begin{bmatrix} v_2 \\ v_1 \end{bmatrix}.$$

As a result, m_1's momentum is transferred to m_2, $v_1 = 0$ and $v_2 = v$. Since, $v_2 > v_1$ the bodies disconnect because the event

$$\gamma_{\alpha_1}^{\alpha_0}\colon\ v_1 - v_2 < 0 \Rightarrow \sigma_{disconnect} \tag{21}$$

becomes true.

The transition function in Eq. (20) specifies that the two bodies *collide* when $x_1 \geqslant x_2$. The collision rule is then applied at the ensuing pinnacle to compute the new velocities v_1 and v_2, and this moves the system back into its *free* mode because $v_2 > v_1$ as specified by Eq. (21). But, since no time has elapsed and $x_1 \geqslant x_2$ still holds, the system switches to the *collide* mode again.[5] This mode is departed immediately because $v_2 > v_1$ still holds (the *a priori* values have not been updated yet), which implies that it is mythical. At this point both of the transition functions that generate $\sigma_{collide}$ and $\sigma_{disconnect}$ are true and g^α only changes the *a posteriori* values. This results in a loop of instantaneous mode changes between α_0 and α_1, and the divergence of time principle is violated.

Divergence of time can be enforced in one of two ways:

- adding more detail to the model so that discontinuous phenomena are modeled as continuous effects, and
- modifying the switching conditions to ensure there is only one possible mode associated with a given value of a state vector.

Adding more continuous detail is undesirable because it is likely to cause a significant increase in the computational complexity of the model. Modification of switching specifications requires revisiting the assumptions under which the discontinuous approximations were made. In case of the elastic collision, the coefficient of restitution is normally a function of the impact velocities [4]. For collisions at low velocities, the collision phenomenon discussed above does not occur, and the coefficient of restitution is a poor discrete approximation of the underlying continuous behavior. Therefore, the transition conditions for the

[5] This also occurs if the state change due to the collision is modeled as a transition action rather than a separate state [45].

collision may be modified by adding the constraint $v_1 - v_2 \geqslant v_{th}$. In the limit, as $v_{th} \to 0$ the original transition condition $v_1 > v_2$ is attained. If the transition condition to *collide* adds on this constraint, one notes that the behavior trajectory does not switch back from the *free* mode to the *collide* mode, and divergence of time is enforced for the collision model.

In previous work, we have shown how analysis of multiple piecewise phase spaces can be applied to establish a necessary condition for divergence of time [32]. The necessary condition is established by mapping each mode of system behavior into a k dimensional phase space representation for a state vector of size k. This requires all discrete switching functions γ to be expressed in terms of the state variables, x. Typically this includes algebraic substitutions, but it may require special computations if Dirac functions are involved. A pairwise region checking algorithm is used to check for overlap between behavior regions for the individual modes. The necessary condition for divergence of time to be satisfied is that there be no overlap between the behavior regions of the individual modes. Sufficient conditions are much harder to establish, because it requires taking into account the mode switching function γ. We are looking into a more systematic computational methodology based on constraint satisfaction techniques [53] in the continuous domain to solve the general n-dimensional phase space problem. A detailed application of the divergence of time algorithm to the falling rod example appears in Section 7.1.

Note the difference between divergence of time and Zenoness [6,19]. Behavior that is non-Zeno does not progress beyond a point in time. However, such behavior may still diverge by taking increasingly smaller steps in time (e.g., a model of a bouncing ball can be constructed so that the ball never comes to rest).

4.2. Temporal evolution of state

When discontinuous state changes occur at a time point t_s on a trajectory that goes from an interval to a pinnacle ($x(t_s^-)$ to $x(t_s)$) or a pinnacle to an interval ($x(t_s)$ to $x(t_s^+)$) (see Fig. 11), discontinuous changes in system variables may result in the generation of Dirac pulses, viz., $\delta_1(t - t_s)$ and $\delta_2(t - t_s)$, respectively. This occurs in the two mass problem discussed in Section 3.3. Dirac pulses have finite area and occur at a point in time. Since both pulses occur at t_s, the total pulse can be defined as the aggregate $\delta_c(t - t_s) = \delta_1(t - t_s) + \delta_2(t - t_s)$. However, $\delta_2(t - t_s)$ is not known at t_s. It can only be

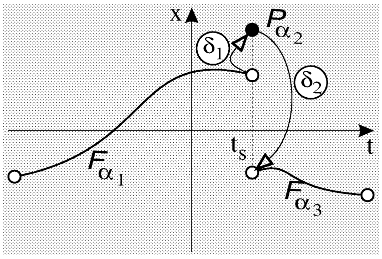


Fig. 11. Switching around a point may require two jumps.

determined when time is advanced. Therefore, $\delta_c(t - t_s)$ is unknown. The variable values in the switching conditions that depend on δ_c and govern the configuration change from t_s^- to t_s are also unknown. Correct physical models are enforced by determining the actual δ_c based on an interval to interval change. To prevent ill-defined noncausal models, execution semantics in the form of temporal evolution of state is imposed on hybrid models so that discontinuous state changes only occur from t_s^- to t_s (i.e., $\delta_2 = 0$) [43]. The variable that is involved has to be continuous on the left-closed interval, $[t_s, \rightarrow>$ in time.

Principle 3 (Temporal evolution of state). *Continuous state variable values have to be continuous in left-closed intervals in time.*

The proof of this lemma is presented in [43].

The requirement that state variable values evolve through left closed intervals in time, implies that discontinuous changes in the state vector can only occur when the system transfers from an interval to a point. This does not prohibit configuration changes occurring from a point to interval transition, provided no discontinuous changes in state variables occur. Further, to ensure δ pulses only occur on left closed interval switching, the transition conditions that result in discontinuous changes in state variable values have to be of the form $\geqslant$ or $\leqslant$. For example, the transition condition from free to collision mode for the colliding bodies is represented as $x_1 \geqslant x_2$. Discontinuous changes in the state vector occur when its size changes and can be derived by inspection of a hybrid bond graph model or by systematic analysis [30,37].

5. A complex example

We describe a more complex example to illustrate the physical principles that are employed in constructing and analyzing hybrid system models. Consider a thin rigid rod falling towards a floor (Fig. 12). The system can be modeled to operate in one of three modes:

(i) *free*, there is no contact between the rod and the floor,

(ii) *stuck*, the rod is in contact with the floor at point A, and this point is fixed while the rod rotates, and

(iii) *slide*, the rod is in contact with the floor at point A, and this point slides along the floor while the rod rotates.

In other work [27,52], the sliding and rotating behavior of the rod when in contact with the floor has been analyzed using complementarity principles. These studies have shown that for certain values of the parameters (friction coefficient, μ, rod length, l, and angle, θ) there are regions in the phase space (i.e., the space spanned by the linear rod velocities, v_x and v_y, and the rotational velocity, ω) where the rod model exhibits multiple behavior trajectories and some regions where no behaviors exist. The parameter values that correspond to these situations are rather extreme, and we do not deal with these particular regions in our analysis of the rod behavior in this paper. Instead, we focus on modeling and analyzing the behavior of the rod upon collision with the floor. We derive physically consistent values of the linear and angular momenta and perform analyses to determine the

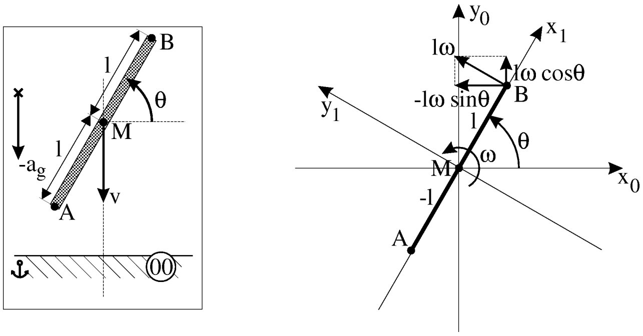


Fig. 12. A collision between a thin rod and a floor.

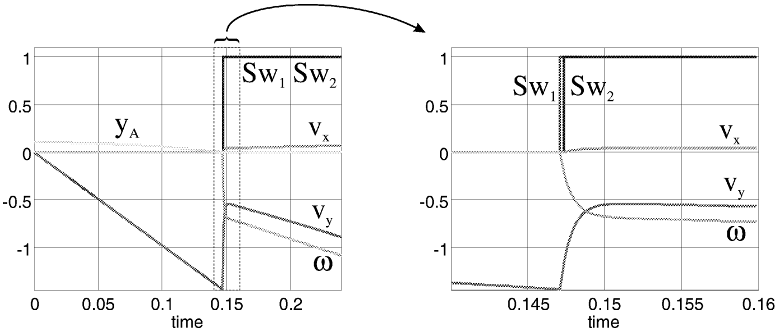


Fig. 13. Simulation of a collision scenario.

new mode of operation. In addition, we ensure that the conditional specifications for the *slide* and *stuck* modes are mutually exclusive. We also formulate switching conditions to ensure the consistency of Dirac pulses generated during mode transitions.

When collision occurs, small deformation effects force the vertical velocity of the rod-tip, A, to quickly become 0. Since the time scale for the collision process is much faster than the time scale for the overall behavior, we assume that the collision phenomenon can be reduced to occur at a point in time. The rod's vertical velocity v_y at its center of mass M changes quickly, so that the resultant angular velocity, ω, satisfies the equation $v_{A,y} = v_y + l\omega\cos\theta = 0$. Further, if the breakaway force is not exceeded by the force along the surface, i.e., $|F_{A,x}| < \mu F_N$, the rod is stuck, and the horizontal velocity of the rod tip becomes zero, i.e., $v_{A,x} = 0$. The resultant change in the center of mass velocity in the horizontal direction, v_x, satisfies $v_{A,x} = v_x - l\omega\sin\theta = 0$.

If the breakaway force is exceeded during the quick continuous transient at collision the rod starts to slide in the next continuous mode, and the $v_{A,x} = 0$ constraint is not included in the set of equations for the active model. Since this happens at initial contact, there is minimal change in the value of v_x through the collision process. The collision process at $y_A = 0$ is illustrated in Fig. 13 at two different time scales for normalized

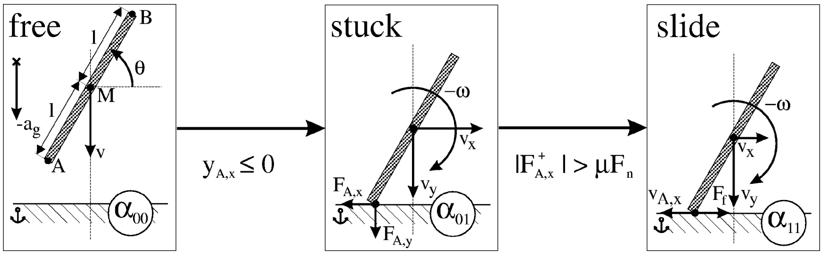


Fig. 14. The rod may start to slide and rotate around the point of contact.

parameters of mass $m_x = 1$, $m_y = 1$, moment of inertia $J = 1$, $F_g = 1$, frictional coefficient $\mu = 0.05$, spring $C = 1$, and damper $R = 750$ to provide a first-order approximation of the detailed behavior. The continuous transient behavior generated from a detailed model of the collision process, occurs in the time interval [0.14, 0.16]. It is illustrated in the behavior plot on the right. The gross behavior just before and after the collision is shown on the left. At the point of collision, $Sw_1 = 1$, the breakaway force is exceeded, therefore, $Sw_2 = 1$, and the rod starts to slide. The amount of change in v_x depends on the friction coefficient μ. In the extreme case, $\mu = 0$, the breakaway force is immediately exceeded, and v_x remains 0, since no horizontal force is active in the new mode.

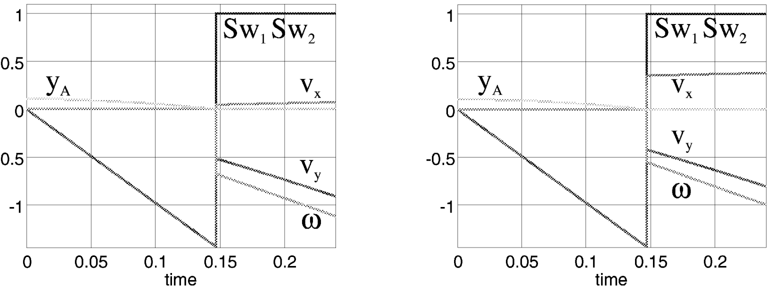
The existence of Coulomb friction [27] between the rod and floor may cause the rod to stick and rotate around the point of initial contact (mode α_{01} in Fig. 14). Alternately, as discussed above, if the rod-tip exerts a force in the horizontal direction that is larger than the product of the normal force and friction coefficient (breakaway force), i.e., $|F_{A,x}| > \mu F_n$, the rod starts to slide (mode α_{11} in Fig. 14). To evaluate which scenario occurs, the force values need to be calculated at the time of collision. Since the impact is idealized, the forces occur as *impulses* [4], and take on the form of Dirac functions (δ). These impulses occur at the time of impact, and their areas are determined by the state vectors immediately prior to and after the impact, x and x^+, respectively.

On contact, i.e., in the mode α_{01}, the linear velocities of the center of mass, v_x and v_y, are completely determined by the angular velocity, ω, and the algebraic relations

$$\begin{cases} v_x^+ = l\omega^+ \sin\theta, \\ v_y^+ = -l\omega^+ \cos\theta. \end{cases} \tag{22}$$

This is illustrated in Fig. 12. Conservation of momentum involving ω^+, v_x^+, and v_y^+ produces one more equation that can be used to solve for the new state vector x^+ [34].

These *a posteriori* values may be such that the corresponding impulses result in $|P_{A,x}| > \mu P_n$ and the rod starts to slide (mode α_{11}). In this mode, v_x^+ is not algebraically dependent on ω, and only $v_y^+ = -l\omega^+ \cos\theta$ holds. Thus the rod-tip moves freely in the x-direction, and its vertical momentum immediately before contact (mode α_{00}), is distributed only over its posteriori angular momentum and vertical momentum to ensure $v_{A,y} = 0$ and y_A does not change (i.e., it satisfies the constraint to remain in contact with the floor). If the continuous state vector in the sliding mode, α_{11}, was computed from the previously inferred mode, α_{01}, because of the v_x^+ dependency on ω^+ in that mode, it would have



(a)

(b)

Fig. 15. The falling rod with (a) parameter abstraction and (b) time scale abstraction semantics.

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. So the consecutive mode switch to α_{11} has to occur before the state vector is updated to its *a posteriori* value, i.e., $x = x^+$.

A behavior trajectory with the same parameter values as in Fig. 13, but with the fast continuous transients removed by parameter abstraction is shown in Fig. 15(a). The computation of the new angular and linear velocities is based on the principles of conservation of state and invariance of state [43] (see Section 3). The results show close conformance with the final values that result from the continuous transients (Fig. 13). As a comparison, the same system is modeled by applying time scale abstraction semantics to the discontinuous change at the point of impact. The results of this are shown in Fig. 15(b). In this case, the intermediate *stuck* mode, α_{01}, that was mythical for parameter abstraction, is now a pinnacle and real. Thus, the state vector is updated in mode α_{01}, and the intermediate dependency between ω, v_x, and v_y causes an instantaneous discontinuous state change in v_x, that becomes the initial velocity value in the *sliding* mode, α_{11}. When comparing the results of the two abstractions to the real behavior, it is clear that parameter abstraction matches the detailed continuous behavior. This is because the collision phenomenon for this example is dominated by the dissipative (frictional) phenomena. In other situations, such as a bouncing ball or an elastic collision of two bodies, the collision phenomenon would be dominated by the capacitive or energy-storage elements, and restitution of momentum would determine the transition behavior. Time scale abstraction would then generate the correct simplified model [44].

Further, the results in Fig. 15 demonstrate that switching in this example has to be based on x^+ rather than x (Fig. 14). Fig. 14 also shows the constraint on the rod-tip position, y_A, to achieve the *contact* mode of operation. As long as the rod exerts a negative, i.e., downward, force on the floor it stays in contact. Otherwise, the normal force, F_n, becomes negative which means the rod disconnects and lifts off the floor.

The complementarity technique applied in other work [27,52] does not require iteration across a series of mode changes, and, therefore, is more efficient. However, it is less general as it does not allow sequential logic to be used in the discrete model specification. All mode changes have to be formulated in complementarity terms.

6. A computational architecture

When sufficiently detailed physical system models are employed for behavior analysis, no discontinuous changes in state variable values occur, and simulation algorithms can be constructed from a single set of ODEs. However, these models often incorporate higher order derivative terms with complex nonlinearities (complex ODE, cODE, systems) that complicate the simulation and behavior analysis algorithms and make them hard to implement [41]. Accuracy of numerical results cannot be guaranteed because of numerical stiffness caused by behaviors occurring at multiple time scales (some very fast), and because the system parameters associated with these models are hard to estimate accurately. Our work has proposed an alternate methodology, which allows the cODE system of equations to be replaced by a set of simpler ODEs (sODEs) derived from the original system model by applying parameter and time scale abstractions.

However, this requires additional effort in the modeling task:

(a) the derivation of discrete event transition conditions, and
(b) determining the corresponding actions for representing discontinuous jumps in the field and state variable values between modes of continuous behavior evolution.

For example, a detailed continuous model for the colliding bodies in Fig. 6 would require modeling of the dynamic phenomena in terms of the position of the bodies and the exchange of force between them during the collision process. The force computation is a function of elasticity coefficients that are characteristic of the body materials and their geometry and frictional forces. This is a very complex modeling task. When the detailed continuous behavior of the system around $\Delta x = 0$ is abstracted away by removing the elasticity effects from the model, the force values are no longer directly computable. The collision process is now defined in terms of discrete events $\{\sigma_{collide}, \sigma_{free}\}$ using position and velocity constraints to define the γ function that ensures the collide mode is immediately departed after the velocities are changed.

Hybrid automata [1] provide a powerful specification formalism for analysis of hybrid dynamic systems. Each state of the automata can be defined in terms of a different set of ODEs that govern system behavior evolution. Therefore, each state corresponds to a continuous mode. Explicit actions specified along with the state transition function (the γ function) can handle the discontinuous changes or jumps in the state vector values (the g function) between mode or state changes.

The mathematical specifications for hybrid system models developed in Sections 3 and 4 are applied to generate the models of dynamic physical systems. For physical system models, the arguments to the event generation function γ are continuous system variables. From systems theory [20], it is known that any continuous system variable can be defined as an algebraic function, h, of the system state and input variables. This implies that the event generation functions can be completely specified in terms of the system state and input variables. A mode change may result in a change in functional relations between state and input variables and other system variables. The recomputed signal values can cause further mode changes, leading to a sequence of mode transitions.

The hybrid system model in Eq. (1) is extended to a complete computational hybrid physical system model defined by the 9-tuple [25,43]

$$H = \langle I, \Sigma, \phi, X_\alpha, U_\alpha, f_\alpha, g^\alpha, h_\alpha, \gamma_\alpha^\beta \rangle. \tag{23}$$

The model consists of three components:

(1) the continuous model, where X_α and U_α denote the state and input vectors, and field, f_α, represents the continuous model in mode α,

(2) the discrete model, where I denotes the discrete indexing set corresponding to the possible modes in the system and Σ represents the set of events that cause mode transitions, and

(3) the interaction model, where the discrete model is represented by the state transition function ϕ, the functions γ and g defined earlier, and h, the system variable computation function represent the interactions between the discrete and continuous models.

These three components are described in greater detail next.

6.1. The continuous model

Continuous physical system behavior is governed by energy interaction. Physical system behavior is typically represented by a state space model with the dynamic behaviors expressed as a set of DAEs. For hybrid models in semi-explicit form, the continuous behavior in each real mode α embodies the time-derivative behavior,

$$\dot{x}(t) = f_\alpha^{dif}\big(x_\alpha(t), u_\alpha(t), t\big),$$

and additional algebraic constraints

$$0 = f_\alpha^{alg}\big(x_\alpha(t), u_\alpha(t), t\big),$$

$t \in \mathbb{R}$ and $\alpha \in \mathbb{N}$. $X_\alpha \in \mathbb{R}^m$ is the continuous state vector, and $U_\alpha \in \mathbb{R}^p$ is the vector of input variables. For each continuous mode α, there is one and only one field, f_α, that defines system behavior.

As an example, the continuous model for the falling rod system (Fig. 12) in the *free* mode representing the rotational behavior at the point of contact, A, is expressed as

$$\begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & J \end{bmatrix} \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_\omega \end{bmatrix} \tag{24}$$

with algebraic constraints

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_\omega \end{bmatrix} = \begin{bmatrix} 0 \\ F_g \\ 0 \end{bmatrix}. \tag{25}$$

At the point of collision, if $|v_y^+ - v_y| > \mu(v_y^+ - v_y)$, the rod starts to slide and another set of DAEs becomes active. Fig. 12 shows that in the *sliding* mode, the constraint $v_y = l\cos\theta$ holds, and this ensures that the rod tip does not move in the vertical direction. The friction force in horizontal direction is related to the normal force by a coefficient μ, resulting in $F_x = -\mu F_y$. The forces in the horizontal and vertical direction are related to the torque F_ω by the equation:

$$-l\sin\theta F_x + l\cos\theta F_y + F_\omega - l\cos\theta F_g = 0. \tag{26}$$

The resulting set of algebraic constraints can be summarized as:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & \mu & 0 \\ 0 & 1 & -l\cos\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & -l\sin\theta & l\cos\theta & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \\ F_x \\ F_y \\ F_\omega \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l\cos\theta F_g \end{bmatrix}. \tag{27}$$

In this mode, v_x and v_y, the linear velocities, are not state variables since they are algebraic functions of ω.[6] Therefore, the size of the state vector changes when the rod moves from the *free* fall mode, α_{00}, to the *sliding* mode, α_{01}. As discussed earlier, this can produce discontinuous changes in the state vector. In contrast, if the rod bounces back up after contact, the system moves from the *sliding* mode to the *free* mode, and the state vector increases in size. If a transition causes a change of algebraic constraints where algebraic relations between state variables are removed, this does not cause discontinuous changes in the state variables.

6.2. *The discrete model*

Discrete events in hybrid dynamic systems are modeling artifacts attributed to parameter and time scale abstractions. The discrete changes are modeled by a transition function, ϕ, and transitions are invoked by events in a set Σ. We adopt a compositional modeling approach and systematically derive ϕ from a set of independent state machines that define local switching effects. Given n independent state machines, a mode is defined as an *n-tuple*, where each element of the tuple is a state of an independent state machine. Theoretically, if each state machine can assume m states, the system has m^n different modes of behavior. However, many of the defined modes may not be reachable for a physical system description. Other modes may only be traversed as mythical modes between two real modes. An important contribution of our work is to establish execution semantics that handle these sequences of mode changes in a consistent manner that ensures behaviors generated are physically correct.

The discrete model can be implemented by Petri nets [47] or finite state machines [21]. We adopt finite state machine models in our representation, defined in terms of the following components:

- $I = \{\alpha_0, \ldots, \alpha_k\}$, a set of states that describe the modes of the system.
- $\Sigma = \{\sigma_0, \ldots, \sigma_l\}$, the set of events that can cause state transitions. Events are generated from signal values in the physical process (Σ_s), or they can be external control signals (Σ_x), $\Sigma = \Sigma_s \times \Sigma_x$.
- $\phi : I \times \Sigma \to I$, a discrete state transition function that defines the new mode after an event occurs.

[6] The choice of ω as state variable is arbitrary. Any algebraic combination of v_x, v_y, and ω may serve as the state variable.

6.3. Interactions

Lygeros, Godbole and Sastry [28] have shown that independent determination and proofs about the continuous behavior and the discrete phenomena in a hybrid model do not constitute proofs of correctness of their combined effects. Hybrid system analysis requires formal specifications of the interactions between the continuous and discrete models. Interactions between the continuous and discrete models are specified by

(i) events generated in the continuous model, and
(ii) mode changes defined by the discrete model.

More formally they can be expressed as:

- $S \in \mathbb{R}^n$, the system variables used for event generation.
- $h : X \times U \times I \to S$, returns system variable values from the input and state variable values in a given mode.
- $g : X \times I \to X^+$, computes the *a posteriori* state vector, X^+, in the new mode from the *a priori* state vector, X. There may be a discontinuous change from X to X^+.
- $\gamma : S \times S^+ \to \Sigma_s$, where Σ_s is the set of discrete events generated from the system variable values. The values S are computed from the *a priori* state vector, X, and the values S^+ from the *a posteriori* state vector, X^+.

The function γ generates discrete events when system variables cross prespecified threshold values. For example, collision events for the falling rod can be defined by the following constraints:

$$\gamma\colon \begin{cases} y_A^+ \leqslant 0 \wedge v_{A,y} < 0 & \Rightarrow \sigma_{contact}, \\ F_n^+ \leqslant 0 & \Rightarrow \sigma_{free}. \end{cases} \tag{28}$$

The function h computes the values of signals that define these events from the continuous state and input vector. For the signals used in the collision transition for the falling rod, this yields

$$h\colon \begin{cases} y_A = \int v_y \, \mathrm{d}t - l \sin\theta, \\ v_{A,y} = m(v_y + l\omega\cos\theta), \\ F_n = \begin{cases} 0 & \text{if } \alpha_{00}, \\ m(\dot{v}_y - a_g) & \text{otherwise,} \end{cases} \end{cases} \tag{29}$$

where a_g is the gravitational constant.

Generated events may imply mode changes, i.e., a change in the DAEs that govern continuous behavior generation. The continuous state vector of the system may also undergo discontinuous change, governed by the transformation function, g. For physical system models this function has to satisfy the principle of *conservation of state* (Section 3). When the falling rod makes first contact with the floor, conservation of state is applied to derive the state vector transformation function [34].

To illustrate the application of the conservation of state principle, we calculate the new state values in the stuck mode, α_{01}. In this mode, point A is in contact with the floor, so $v_{A,y} = 0$, which requires $v_y = -l\omega\cos\theta$. Further, since the rod does not move in the horizontal direction, $v_{A,x} = 0$, which requires $v_x = l\omega \sin\theta$. This imposes algebraic constraints on v_x, v_y and ω, which may result in impulses whose values are determined by applying conservation principles involving the components of the linear velocities and the angular velocity in the initial configuration. The relation between the forces in the x and y direction and the torque F_ω is given in Eq. (26).

The equations representing the rotational behavior at the point of contact, A, are given by Eq. (24) with the additional algebraic constraints

$$\begin{bmatrix} 1 & 0 & l\sin\theta & 0 & 0 & 0 \\ 0 & 1 & -l\cos\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & -l\sin\theta & l\cos\theta & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \\ F_x \\ F_y \\ F_\omega \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l\cos\theta F_g \end{bmatrix}. \tag{30}$$

Combining Eqs. (24) and (30) we can derive

$$-l\sin\theta m_x \dot{v}_x + l\cos\theta m_y \dot{v}_y + J\dot{\omega} = l\cos\theta F_g, \tag{31}$$

and integrating this over an infinitesimal interval $[t, t^+]$ we get

$$-l\sin\theta m_x(v_x^+ - v_x) + l\cos\theta m_y(v_y^+ - v_y) + J(\omega^+ - \omega) = 0.$$

The right-hand side of Eq. (31)

$$\int_t^{t^+} l\cos\theta F_g \,\mathrm{d}\tau = 0$$

evaluates to 0, because this term remains constant over the infinitesimal small interval of time. The above equation combined with

$$\begin{cases} v_x^+ = -l\omega^+ \sin\theta, \\ v_y^+ = l\omega^+ \cos\theta, \end{cases} \tag{32}$$

allows us to solve for ω^+, producing the state projection

$$\omega^+ = \frac{\omega J - ml(\cos\theta v_y - \sin\theta v_x)}{J + ml^2}. \tag{33}$$

This is in conformance with the topological analysis in [30]. Using the relations in Eq. (32), produces the state mapping

$$g^{\alpha_{01}}\colon \begin{cases} \omega^+ = \dfrac{\omega J - ml(\cos\theta v_y - \sin\theta v_x)}{J + ml^2}, \\ v_x^+ = l\omega^+ \sin\theta, \\ v_y^+ = -l\omega^+ \cos\theta. \end{cases} \tag{34}$$

For comparison, in mode α_{00}, the rod has three degrees of freedom, and the state mapping does not cause discontinuous changes

$$g^{\alpha_{00}}\colon \begin{cases} \omega^+ = \omega, \\ v_x^+ = v_x, \\ v_y^+ = v_y. \end{cases} \tag{35}$$

6.4. From DAE to ODE

As shown, the initial modeling phase relies on building a system of differential equations with variable algebraic constraints. The algebraic equations in these DAEs may enforce constraints on state variables, thereby reducing the degrees of freedom of the system. In many cases, when a new set of algebraic equations become active, the state variable values have to instantaneously change to satisfy the new constraints. The exact values of these instantaneous changes are made explicit by using conservation principles and computed by forming the instantaneous equivalent of the dynamic behavior.

Once the instantaneous changes are computed, they become part of the g function. Now that they are explicitly available, the DAE system can be algebraically manipulated into an ODE form. This approach results in models that closely match the hybrid automata modeling paradigm [2] and enables the use of standard simulation packages. Furthermore, the model is amenable for other analyses, such as system verification for control purposes.

7. Verifying the model

We apply the principles of divergence of time and temporal evolution of state to the hybrid dynamic model of the falling rod system to ensure that this model generates physically consistent behaviors.

7.1. Divergence of time

The falling rod (Fig. 12) moves into its *stuck* mode (α_{01}) if it is in contact with the floor and its horizontal tip velocity falls below a threshold value, i.e., $|v_{A,x}| \leqslant v_{th}$, where v_{th} is the threshold velocity. At the same time, if the values of the friction coefficient, μ, angle, θ, and length, l of the rod are such that $|F_{A,x}| > \mu F_n$, the event σ_{slide} is generated, and the divergence of time principle is violated. Since the two events σ_{stuck} and σ_{slide} alternate, the system continues to switch between the *stuck* and *sliding* modes with no progression of behavior in real time.

Systematic checking of the divergence of time principle for the falling rod system, requires that all local switching conditions be expressed in terms of the system state variables. The phase space for this system is 5-dimensional with axes $(\theta, y, v_x, v_y, \omega)$ (across all modes α_{00}, α_{01}, α_{10}, and α_{11}). As discussed earlier, necessary conditions for divergence of time can be established by looking at boundary conditions for the modes

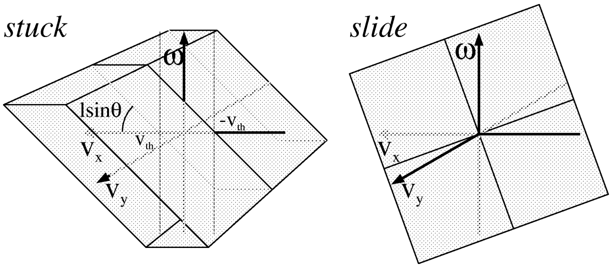


Fig. 16. Transition conditions for S.

pairwise. Like before [32,33], we present a conceptual visual phase space analysis of the divergence of time principle, and apply it to modes α_{01} and α_{11} of the falling rod system. To simplify the phase space representation, we assume θ and l are fixed for the rod, and consider a three-dimensional space with orthogonal axes, v_x, v_y, and ω.

First, the condition for the rod being stuck in mode α_{11}, $|v_{A,x}| \leqslant v_{th}$, is recomputed in terms of state variables. This condition translates to $l\omega\sin\theta - v_{th} < v_x < l\omega\sin\theta + v_{th}$ (see left-hand side of Fig. 16). The translation of the sliding condition, $|F_x| > \mu F_N$, in mode α_{01}, requires the observation that F_x and F_y have derivative relations in this mode. The corresponding switching conditions in terms of the state variables are $|\frac{\mathrm{d}v_x}{\mathrm{d}t}| > \mu(\frac{\mathrm{d}v_y}{\mathrm{d}t} - a_g)$. When switching from *slide* to *stuck*, the linear velocities change discontinuously generating Dirac pulses, which makes the effect of a_g negligible. A comparison of the two Dirac pulse areas results in the switching condition $|v_x^+ - v_x| > \mu(v_y^+ - v_y)$. Consider the situation where $v_x^+ - v_x > 0$ in the mode α_{01} (Fig 14). Switching occurs if $v_x^+ - v_x > \mu(v_y^+ - v_y)$, where v_x^+ and v_y^+ can be expressed in terms of v_x, v_y, and ω. Making the substitutions produces an inequality of the form, $c_1(\mu,\theta,l)v_x + c_2(\mu,\theta,l)v_y + c_3(\theta,\mu,l)\omega > 0$.

In the three dimensional phase space, this switching condition represents one side of the plane (see right-hand side of Fig. 16) with normal vector

$$\big[c_1(\mu,\theta,l)\ c_2(\mu,\theta,l)\ c_3(\theta,\mu,l)\big]^{\mathrm{T}}. \tag{36}$$

This represents the *sliding* region, and because of the strict inequality, the boundary is not part of the region from which a transition occurs. Depending on the direction of the normal vector, any point in the phase space may satisfy the transition condition and overlap with the transition condition for the *stuck* mode. Therefore, if both spaces are intersected for $v_{th} > 0$, there is an area in (v_x, v_y, ω) that causes the rod to be *stuck* as well as to *slide*. In this space, the model can continue to make infinite transitions between *sliding* and *stuck*, and, therefore, cannot reach a new mode of continuous evolution. This violates the divergence of time principle in Section 4.

As discussed earlier, the modeling inconsistency may be eliminated by

(1) adding constraints to the mode switching conditions for *sliding* and *stuck*, or
(2) modeling the system in greater detail by adding parasitic phenomena so that the model exhibits continuous behavior and the discrete transitions are removed.

Additional mode switching constraints may be included by modifying either the condition for sliding or the condition for getting stuck. A modeling decision can be made to generate

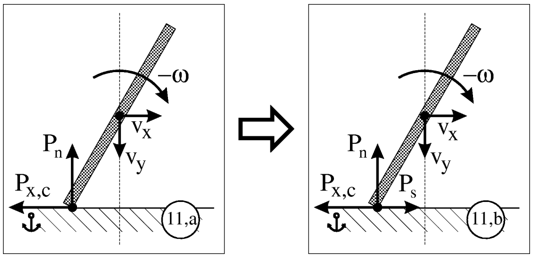


Fig. 17. Impulses upon collision when a sliding mode with stiction is reached.

σ_{stuck} only if the forces in α_{01} are such that σ_{slide} is not generated. This requires the addition of a pre-condition $|F_{A,x}^{\alpha_{01}}| \leqslant \mu F_n^{\alpha_{01}}$ to σ_{stuck}, where $F_{A,x}^{\alpha_{01}}$ and $\mu F_n^{\alpha_{01}}$ are calculated from $h(g^{\alpha_{01}}(x))$.

7.2. Temporal evolution of state

Again consider the falling rod system in Fig. 14. At time t_s when the rod collides with the floor, the horizontal and vertical velocities of the center of mass of the rod change discontinuously. $v_x(t_s^-) = \lim_{t\uparrow t_s} v_x(t)$ differs from $v_x(t_s)$, and this results in a horizontal impulse $P_{x,c}(t_s)$. $v_y(t_s^-) = \lim_{t\uparrow t_s} v_y(t)$ also differs from $v_y(t_s)$, and this results in a normal impulse $P_n(t_s)$ (mode $\alpha_{11,a}$ in Fig. 17). No other forces are active, and the total impulse along the surface is $P_{A,x}(t_s) = P_{x,c}(t_s)$. If $|P_{A,x}| > \mu P_n$ a second mode change occurs to $\alpha_{11,b}$, and the rod starts to slide. A stiction impulse may become active when the rod starts to slide causing a discontinuous change in the horizontal velocity of the rod, and $v_x(t_s^+) = \lim_{t\downarrow t_s} v_x(t)$ differs from $v_x(t_s)$. Now, the aggregate impulse, expressed as $P_{A,x}(t_s) = P_{x,c}(t_s) + P_s(t_s)$, may not satisfy the criterion for sliding, $|P_{A,x}| > \mu P_n$.

The principle of temporal evolution of state (Section 4) implies that the stiction impulse cannot become active after the rod-tip has started sliding. It has to be activated at time point, t_s, when it is determined whether the rod slides or not. If the effect of the stiction impulse is taken into account along with $P_{x,c}(t_s)$, $P_{A,x}(t_s)$ is derived correctly.

For the falling rod, the state vector reduces in size upon collision, and, this may cause discontinuous changes in state variable values. To ensure that the corresponding Dirac pulses are well defined and can be aggregated as was done above, the = sign is included in the transition condition, $y_A \leqslant 0$. Reductions in the state vector may also occur when the rod gets stuck after sliding, which is again why the = sign is included in the $|v_{A,x}| \leqslant v_{th}$ condition.

8. The falling rod specification

In Sections 5 and 6, starting from the physical specifications, we derived the hybrid automata model for the falling rod system. Section 7 applied verification principles to ensure that the model developed did not violate physical principles. In this section we present the complete mathematical specification for the hybrid model of the falling rod system. This specification may be employed for simulating rod behavior [36], and for other forms of behavior analysis (e.g., [43]). It is assumed that the linear inertias are equal, i.e., $m_x = m_y = m$. The three energy storing elements, with associated linear velocities, v_x and v_y, and rotational velocity, ω, define the energy part of the system state vector,

$$X = \{\omega, v_x, v_y\}. \tag{37}$$

Two external input forces act on the rod,

$$U = \{F_f, ma_g\}. \tag{38}$$

F_f is the friction force, and ma_g represents the gravitational force (a_g is the gravitational constant). The signals that generate discrete event transitions in the system model are

$$S = \{y_A, v_{A,x}, v_{A,y}, F_n, F_{A,x}\}, \tag{39}$$

and the corresponding discrete events are:

$$\Sigma = \{\sigma_{contact}, \sigma_{free}, \sigma_{slide}, \sigma_{stuck}\}. \tag{40}$$

The continuous vector field in each of the modes is captured by field functions f, defined as

$$f\colon \begin{cases} \alpha_{00}\colon \begin{cases} \dot{\omega} = 0, \\ \dot{v}_x = 0, \\ \dot{v}_y = a_g, \end{cases} \\ \alpha_{01}\colon \begin{cases} \dot{\omega} = \dfrac{-ml\cos\theta}{J + ml^2} a_g, \\ \dot{v}_x = l\sin\theta\dot{\omega}, \\ \dot{v}_y = -l\cos\theta\dot{\omega}, \end{cases} \\ \alpha_{11}\colon \begin{cases} \dot{\omega} = \dfrac{-ml(\cos\theta - \mu\sin\theta)}{J + ml^2\cos\theta(\cos\theta - \mu\sin\theta)} a_g, \\ \dot{v}_x = -\mu(l\cos\theta\dot{\omega} + a_g), \\ \dot{v}_y = -l\cos\theta\dot{\omega}. \end{cases} \end{cases} \tag{41}$$

Discontinuous changes in the continuous state vector, X, caused by mode changes are described by g. As described earlier, the principle of conservation of state is applied to derive the g function for each mode.

$$g\colon \begin{cases} \alpha_{00}\colon \begin{cases} \omega^+ = \omega, \\ v_x^+ = v_x, \\ v_y^+ = v_y, \end{cases} \\ \alpha_{01}\colon \begin{cases} \omega^+ = \dfrac{\omega J - ml(\cos\theta v_y - \sin\theta v_x)}{J + ml^2}, \\ v_x^+ = l\omega^+ \sin\theta, \\ v_y^+ = -l\omega^+ \cos\theta, \end{cases} \\ \alpha_{11}\colon \begin{cases} \omega^+ = \dfrac{\omega J - ml(\cos\theta - \mu\sin\theta)v_y}{J + ml^2\cos\theta(\cos\theta - \mu\sin\theta)}, \\ v_x^+ = -\mu(l\omega^+ \cos\theta + v_y) + v_x, \\ v_y^+ = -l\omega^+ \cos\theta. \end{cases} \end{cases} \tag{42}$$

The signal generation function, h, for each mode of system operation is defined as:

$$h\colon \begin{cases} y_A = \displaystyle\int v_y \,\mathrm{d}t - l\sin\theta, \\ v_{A,x} = v_x - l\omega\sin\theta, \\ v_{A,y} = v_y + l\omega\cos\theta, \\ F_n = \begin{cases} 0 & \text{if } \alpha_{00}, \\ m(\dot{v}_y - a_g) & \text{otherwise}, \end{cases} \\ F_{A,x} = \begin{cases} 0 & \text{if } \alpha_{00}, \\ m\dot{v}_x & \text{otherwise}. \end{cases} \end{cases} \tag{43}$$

The event generation function, γ, is defined as:

$$\gamma\colon \begin{cases} y_A \leqslant 0 \wedge v_{A,y} \leqslant 0 & \Rightarrow \sigma_{contact}, \\ F_n^+ \leqslant 0 & \Rightarrow \sigma_{free}, \\ |F_{A,x}^+| - \mu F_n^+ > 0 & \Rightarrow \sigma_{slide}, \\ |v_{A,x}^+| - v_{th} \leqslant 0 \wedge |F_{A,x}^{\alpha_{01}}| \leqslant \mu F_n^{\alpha_{01}} & \Rightarrow \sigma_{stuck}. \end{cases} \tag{44}$$

These events cause changes in the model state according to two state transition tables. Table 1 shows the state transition behavior that specifies whether the rod is in contact with the floor, model ϕ_C, and Table 2 specifies the discrete event transition behavior between being free, sliding and being stuck, i.e., model ϕ_S. In these tables, the rows represent the transition from a state as specified in the first entry. The remaining entries represent the new states given an event. The states of both these tables constitute the global mode α_{CS}.

We have employed the mathematical model specifications to develop a hybrid simulation methodology [36,42]. This methodology allows for a direct mapping of system com-

Table 1
State transition table specifying ϕ_C

ϕ_C	$\sigma_{contact}$	σ_{free}
0	1	–
1	–	0

Table 2
State transition table specifying ϕ_S

ϕ_S	σ_{free}	σ_{slide}	σ_{stuck}
0	–	1	–
1	0	–	0

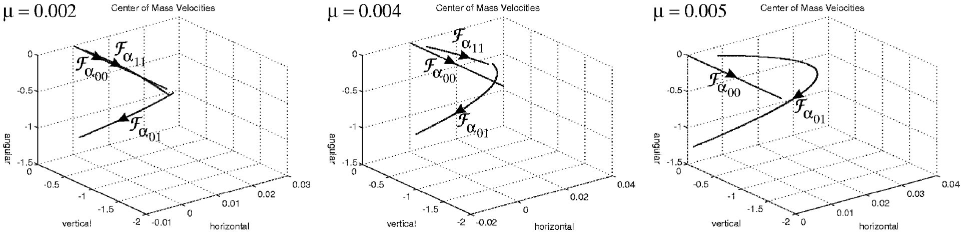


Fig. 18. A number of trajectories in phase space of the colliding rod, $v_{th} = 0.0015$, $\theta = 0.862$, $l = -0.1$, $y_0 = 0.23$.

ponents in Eqs. (41) through (44) into model fragments. The simulation algorithm encompasses discrete switching implemented as instantaneous transition functions, and continuous behavior generation based on ODEs implemented using a forward Euler integrator. A third function computes signal values at points of discontinuous change, such as pinnacles. Details of the hybrid simulation algorithm are presented in [36].

The simulated trajectories of the rod in phase space for three different values of the frictional coefficient (μ) are shown in Fig. 18. The system is initialized with zero angular and linear velocities, $(0, 0, 0)$. Once the rod is released with center of gravity at y_0, flow $\mathcal{F}_{\alpha_{00}}$ applies, and the magnitude of its vertical velocity increases in time. When the rod-tip, point A, touches the floor the rod may start to slide, governed by flow $\mathcal{F}_{\alpha_{11}}$ (happens when $\mu = 0.002$ and $\mu = 0.004$), or it may get stuck and behavior is governed by flow $\mathcal{F}_{\alpha_{01}}$ (happens when $\mu = 0.005$). The discontinuous jumps between flows are illustrated in Fig. 18. Also, for simulations with $\mu = 0.002$ and $\mu = 0.004$, the sliding mode, α_{11} is activated immediately after α_{00} because a force balance computation indicates that the stuck mode α_{01} is departed instantaneously, i.e., it has no real existence at the point of collision.

When sliding, the center of mass of the rod accelerates in the horizontal direction, and the negative velocity at the rod-tip decreases. When it falls below a threshold value, transition conditions determine that the rod gets stuck, which implies a mode change to α_{01} and field $\mathcal{F}_{\alpha_{01}}$. As discussed earlier, the transition conditions had to be properly specified so that the system does not go into a loop of instantaneous mode changes (sliding and stuck), which would violate the divergence of time principle.

If the simulation is repeated with a longer rod, initially the rod may slide on hitting the ground, but the moment it starts sliding, the balance of forces indicates that the rod disconnects and lifts off the ground. In this case the rod is in the sliding mode for a point in time, after which it transitions back to the free mode of operation. Note that this occurs even though the collision is modeled to be perfectly nonelastic, i.e., there is no restitution of

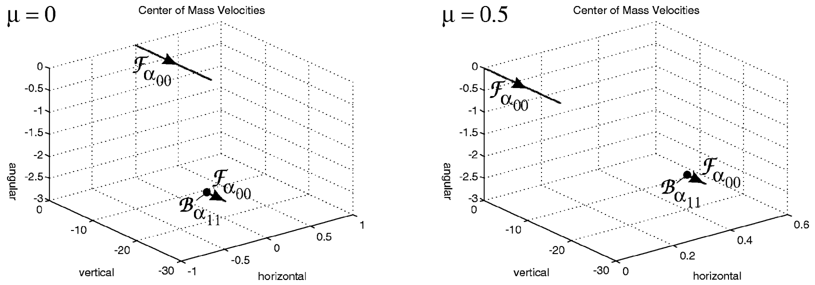


Fig. 19. A boundary in phase space of the colliding rod, $v_{th} = 0.0015$, $\theta = 0.862$, $l = -10$, $y_0 = 23$.

momentum difference in any of the operational modes ($\varepsilon = 0$). Simulation results for this example are shown in Fig. 19. This simulation demonstrates how the boundary point $\mathcal{B}_{\alpha_{11}}$ changes the state vector between the two flows in α_{00}. Note that a field governs behavior in α_{11}, so the corresponding point in phase space is a boundary point rather than a pinnacle.

An important observation one makes from these simulation runs is that the system modes cannot be labeled as mythical, continuous, and pinnacles before hand. For example, in some cases the stuck mode, α_{01}, is departed immediately, whereas in other cases it becomes a real mode, and the rod begins to rotate about its point of contact with the floor. Whether a mode is continuous, a pinnacle, or mythical depends on the state vector values. Therefore, eliminating mythical modes from the model during a compilation stage needs to take state vector value ranges into account. Note that it is not sensible to do this in the model specification as it affects the model structure, and, therefore, the compositionality of the model. However, simulation efficiency can be increased if such processing is performed on the simulation model.

9. General principles for building hybrid systems models

In previous work [30,32,37] we have employed the bond graph modeling language [20] supplemented with finite state automata to represent discrete switching actions as the framework for constructing hybrid models of complex physical system behavior. A set of principles were then developed to model the discrete switching between modes of continuous behavior evolution. This paper extends the work to a more general mathematical specification language that captures the semantics for parameter and time scale abstractions. Formal verification[7] techniques that include the principles of invariance of state, divergence of time, and the temporal evolution of state have been defined. The formal mathematical specifications in conjunction with the physical principles easily translate into computational behavior generation and analysis schemes using hybrid automata [2], where continuous behavior evolution in individual automata states are defined in terms of differential equations (i.e., the field functions f in individual modes

[7] Verification results in a physical system model that obeys the laws of physics. Validation then tests whether the model is an accurate description of the actual system. In a sense, this corresponds to checking meta-model and model characteristics, respectively.

of operation) and discrete mode changes are modeled as state transition functions with algebraic constraints for the hybrid automata.

We have developed preliminary techniques where we start with system behavior descriptions represented as complex ODEs (cODEs), identify fast transients in system behavior, and apply parameter and time scale abstractions to simplify the system model [40]. The resultant hybrid automata are made up of states defined by numerically simpler ODEs (sODEs), but require the definition of the discrete transition function, ϕ, state update function, g^{α}, and the event generation function, γ_{α}^{β} to complete the hybrid automata model. We have illustrated by the falling rod example, that the construction of the γ function must ensure that the principles of divergence of time and temporal evolution of state are not violated. The g function construction requires an understanding of the system configuration and state vectors in the different modes of operation, and the application of the conservation of state principle to derive the change in the state vector across the discrete transition. Besides the falling rod system, we have applied this hybrid modeling methodology to a number of realistic systems, such as the secondary sodium cooling loop of fast breeder reactors [43], elevator control system of aircraft [38,40], cooling system of a Chevrolet V-8 combustion engine [29], braking and clutch mechanisms [42], and a cam-follower mechanism in automobile engines [33].

The next step is to extend the formal specifications to design a declarative model construction framework using compositional modeling techniques employed in [3,13,26, 48]. Preliminary work [39,40] in composing hybrid models of aircraft elevator systems has been demonstrated using hybrid automata. Individual component models were constructed for valves, cylinders, pistons, and the elevator flap as hybrid automata. Each automaton included the continuous behavior descriptions (as DAEs) and mode transition conditions expressed as inequalities on system variables. The component automata models were composed to build the complete simplified hybrid automata model of the elevator system.

This approach needs to be extended to an environment, where model builders, depending on the task, and the goals of their analysis can construct system models with different emphases and at different levels of detail. In previous work (e.g., [26,48]) this has been achieved by the construction of model fragment libraries using a declarative logical specification to enable automated model construction using compositional modeling techniques, and to promote model fragment reuse across multiple applications. Our goal is to construct model fragment libraries, employing hybrid automata as the core modeling language, and augmenting the automata with modeling assumptions that define the level of detail expressed in the model fragment and other conditions that need to be satisfied for the fragment to be applicable [26]. The hybrid automaton composition procedures outlined in [40] can then be applied for model construction.

A new challenge that we face in the design of the compositional modeling algorithm is to ensure the consistency of the composed model fragments at a chosen level of abstraction. In recent work [39], we have demonstrated that the application of abstraction techniques to model fragments may lead to loss of information that makes the model composition task difficult to achieve. Our current solution to this problem is to use the more detailed model fragments and rederive the sODE models and transition conditions for a new system configuration. In future work, we will investigate these issues in greater detail for designing compositional hybrid modeling procedures.

10. Conclusions

This paper has developed a formal mathematical framework for representing hybrid dynamic systems by using DAEs with variable algebraic constraints and analyzing hybrid behaviors of dynamic physical systems. We adopted the conventional modeling framework of Guckenheimer and Johnson [15], and defined the hybrid behavior trajectory in individual modes, α, as continuous flows $\mathcal{F}_\alpha$ interspersed with discrete transitions defined by threshold functions γ_α^β. At discrete transition points, functions g_α^β define jumps in the system state vector, x. The analysis of *parameter* and *time scale* abstractions played a key role in developing systematic and unambiguous switching specifications (i.e., the γ and g functions) based on physical system principles for three modes of operation:

(i) continuous,
(ii) mythical, and
(iii) pinnacles.

An important derivation was to show the exact dependence of the transition conditions on *a priori* and *a posteriori* state vector values. The formal modeling specifications provided a model building framework for physical systems made up of three components:

(i) the continuous model represented as DAEs,
(ii) the discrete model represented as switching conditions for finite state automata, and
(iii) the interaction model defined by the three mathematical functions, h, g, and γ.

The function g is implicit in the DAEs and can be made explicit by applying conservation laws embodies by the instantaneous equivalent of the continuous dynamics. This allows the use of ODE models combined with the g function. This model formulation is the input to our simulation engine but could also be used for other analyses as it is close to the hybrid automata formulation. Along with the model building framework, we have also derived model verification procedures based on the principles of temporal evolution of state and divergence of time. In parallel work [36], the formal specifications have also been translated into a hybrid dynamic simulator, which has been used for generating all the behaviors reported in this paper.

The use of local specifications to define mode transitions often results in the system going through a sequence of discrete changes, but local specifications simplify the modeling task allowing the modeler to create a complex system model as a composition of individual hybrid automata. In other work, we have investigated the issue of building compositional models as time scale and parameter abstractions are applied to simplify the individual component cODEs to sODEs [39]. As discussed earlier, further work remains to be done to develop systematic compositional modeling schemes with hybrid automata. We have undertaken other research effort directed toward incorporating controller actions explicitly into the modeling framework (these will be distinguished from the autonomous jumps discussed in this paper, see [5,46] for a classification of hybrid transitions) so that we can develop methodologies for the design and analysis of embedded (computer-based) control [43], and to use this modeling methodology to build hybrid observers for real time monitoring and diagnosis [38] of complex systems.

References

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

[2] R. Alur, C. Courcoubetis, T.A. Henzinger, P.-H. Ho, Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems, in: R.L. Grossman, A. Nerode, A.P. Ravn, H. Rischel (Eds.), Lecture Notes in Computer Science, Vol. 736, Springer, Berlin, 1993, pp. 209–229.

[3] G. Biswas, X. Yu, A formal modeling scheme for continuous systems: Focus on diagnosis, in: Proc. IJCAI-93, Chambéry, France, 1993, pp. 1474–1479.

[4] R.M. Brach, Mechanical Impact Dynamics, Wiley, New York, 1991.

[5] M.S. Branicky, V.S. Borkar, S.K. Mitter, A unified framework for hybrid control: Model and optimal control theory, IEEE Trans. Automat. Control 43 (1) (1998) 31–45.

[6] P.C. Breedveld, Physical systems theory in terms of bond graphs, Ph.D. Thesis, University of Twente, Enschede, Netherlands, 1984.

[7] H.B. Callen, Thermodynamics, Wiley, New York/London, 1960.

[8] F.E. Cellier, H. Elmqvist, M. Otter, Modelling from physical principles, in: W.S. Levine (Ed.), The Control Handbook, CRC Press, Boca Raton, FL, 1996, pp. 99–107.

[9] J. de Kleer, J.S. Brown, A qualitative physics based on confluences, Artificial Intelligence 24 (1984) 7–83.

[10] T.J.A. de Vries, P.C. Breedveld, P. Meindertsma, Polymorphic modeling of engineering systems, in: Proc. International Conference on Bond Graph Modeling, San Diego, CA, 1993, pp. 17–22.

[11] A. Deshpande, P. Varaiya, Viable control of hybrid systems, in: P. Antsaklis, W. Kohn, A. Nerode, S. Sastry (Eds.), Hybrid Systems II, Lecture Notes in Computer Science, Vol. 999, Springer, Berlin, 1995, pp. 128–147.

[12] G. Falk, W. Ruppel, Energie und Entropie: Eine Einführung in die Thermodynamik, Springer, Berlin, 1976.

[13] B. Falkenhainer, K. Forbus, Compositional modeling: Finding the right model for the job, Artificial Intelligence 51 (1991) 95–143.

[14] K.D. Forbus, Qualitative process theory, Artificial Intelligence 24 (1984) 85–168.

[15] J. Guckenheimer, S. Johnson, Planar hybrid systems, in: P. Antsaklis, W. Kohn, A. Nerode, S. Sastry (Eds.), Hybrid Systems II, Lecture Notes in Computer Science, Vol. 999, Springer, Berlin, 1995, pp. 202–225.

[16] V. Gupta, R. Jagadeeshan, V.A. Saraswat, Computing with continuous change, Sci. Comput. Programming 30 (1998) 3–49.

[17] W.P.M.H. Heemels, J.M. Schumacher, S. Weiland, Linear complementarity systems, Technical Report 97 I/01, Measurement and Control Systems, Department of Electrical Engineering, Eindhoven University of Technology, 1997.

[18] Y. Iwasaki, I. Bhandari, Formal basis for commonsense abstraction of dynamic systems, in: Proc. AAAI-88, St. Paul, MN, 1988, pp. 307–312.

[19] K.H. Johansson, M. Egerstedt, J. Lygeros, S. Sastry, On the regularization of zeno hybrid automata, Systems and Control Letters 38 (1999) 141–150.

[20] D.C. Karnopp, D.L. Margolis, R.C. Rosenberg, Systems Dynamics: A Unified Approach, 2nd edn., Wiley, New York, 1990.

[21] Z. Kohavi, Switching and Finite Automata Theory, McGraw-Hill, New York, 1978.

[22] B. Kuipers, Qualitative simulation, Artificial Intelligence 29 (1986) 289–338.

[23] B. Kuipers, Qualitative simulation using time-scale abstraction, Internat. J. Artificial Intelligence in Engrg. 3 (1988) 185–191.

[24] B. Kuipers, Qualitative Reasoning: Modeling and Simulating with Incomplete Knowledge, MIT Press, Cambridge, MA, 1994.

[25] B. Lennartson, M. Tittus, B. Egardt, S. Pettersson, Hybrid systems in process control, IEEE Control Systems (October 1996) 45–56.

[26] A.Y. Levy, Y. Iwasaki, R. Fikes, Automated model selection for simulation based on relevance reasoning, Artificial Intelligence 96 (1997) 351–394.

[27] P. Lötstedt, Coulomb friction in two-dimensional rigid body systems, Z. Angew. Math. Mech. 61 (1981) 605–615.

[28] J. Lygeros, D. Godbole, S. Sastry, Simulation as a tool for hybrid system design, in: Proc. 1994 AIS Conference on Distributed Interactive Simulation Environments, 1994.

[29] E.J. Manders, G. Biswas, P.J. Mosterman, J. Barnett, Signal interpretation for monitoring and diagnosis: A cooling system testbed, IEEE Transactions on Instrumentation and Measurement (July 2000).

[30] P.J. Mosterman, Hybrid dynamic systems: A hybrid bond graph modeling paradigm and its application in diagnosis, Ph.D. Thesis, Vanderbilt University, Nashville, TN, 1997.

[31] P.J. Mosterman, State space projection onto linear DAE manifolds using conservation principles, Technical Report #R262-98, Institute of Robotics and System Dynamics, DLR Oberpfaffenhofen, Wessling, Germany, 1998.

[32] P.J. Mosterman, G. Biswas, A formal hybrid modeling scheme for handling discontinuities in physical system models, in: Proc. AAAI-96, Portland, OR, 1996, pp. 985–990.

[33] P.J. Mosterman, G. Biswas, Verification of dynamic physical system models, in: Proc. ASME Congress ’96, Atlanta, GA, 1996, pp. 707–714.

[34] P.J. Mosterman, G. Biswas, Formal specifications for hybrid dynamical systems, in: Proc. IJCAI-97, Nagoya, Japan, 1997, pp. 568–573.

[35] P.J. Mosterman, G. Biswas, Principles for modeling, verification, and simulation of hybrid dynamic systems, in: Proc. 5th International Conference on Hybrid Systems, Notre Dame, IN, 1997, pp. 21–27.

[36] P.J. Mosterman, G. Biswas, A modeling and simulation methodology for hybrid dynamic physical systems, SCS Trans. (1998) (in review).

[37] P.J. Mosterman, G. Biswas, A theory of discontinuities in dynamic physical systems, J. Franklin Institute 335B (3) (1998) 401–439.

[38] P.J. Mosterman, G. Biswas, Building hybrid observers for complex dynamic systems using model abstractions, in: F.W. Vaandrager, J.H. van Schuppen (Eds.), Hybrid Systems: Computation and Control, Lecture Notes in Computer Science, Vol. 1569, Springer, Berlin, 1999, pp. 178–192.

[39] P.J. Mosterman, G. Biswas, Deriving discontinuous state changes for reduced order systems and the effect on compositionality, in: Proc. 13th International Workshop on Qualitative Reasoning, Lock Awe Hotel, Scotland, 1999, pp. 160–168.

[40] P.J. Mosterman, G. Biswas, Hybrid automata for modeling discrete transitions in complex dynamic systems, in: Proc. AAAI Spring Symposium Working Notes: Hybrid Systems and AI, Stanford University, CA, 1999, pp. 136–141.

[41] P.J. Mosterman, G. Biswas, Towards procedures for systematically deriving hybrid models of complex systems, in: N. Lynch, B. Krogh (Eds.), Hybrid Systems: Computation and Control, Lecture Notes in Computer Science, Springer, Berlin, 2000, pp. 324–337.

[42] P.J. Mosterman, G. Biswas, M. Otter, Simulation of discontinuities in physical system models based on conservation principles, in: Proc. SCS Summer Simulation Conference, Reno, NV, 1998, pp. 320–325.

[43] P.J. Mosterman, G. Biswas, J. Sztipanovits, A hybrid modeling and verification paradigm for embedded control systems, Control Engineering Practice 6 (1998) 511–521.

[44] P.J. Mosterman, P. Breedveld, Some guidelines for stiff model implementation with the use of discontinuities, in: Proc. ICBGM-99, San Francisco, CA, 1999, pp. 175–180.

[45] P.J. Mosterman, J.F. Broenink, G. Biswas, Model semantics and simulation of time scale abstractions in collision models, in: Proc. Eurosim-98, Helsinki, Finland, 1998, pp. 230–237.

[46] P.J. Mosterman, F. Zhao, G. Biswas, An ontology for transitions in physical dynamic systems, in: Proc. AAAI-98, Madison, WI, 1998, pp. 219–224.

[47] T. Murata, Petri nets: Properties, analysis and applications, Proc. IEEE 77 (4) (1989) 541–580.

[48] P.P. Nayak, L. Joscowicz, S. Addanki, Automated model selection using context-dependent behaviors, in: Proc. AAAI-91, Anaheim, CA, 1991, pp. 710–716.

[49] A. Nerode, W. Kohn, Models for hybrid systems: Automata, topologies, controllability, observability, in: Hybrid Systems, Lecture Notes in Computer Science, Vol. 736, Springer, Berlin, 1993.

[50] T. Nishida, S. Doshita, Reasoning about discontinuous change, in: Proc. AAAI-87, Seattle, WA, 1987, pp. 643–648.

[51] H.M. Paynter, Analysis and Design of Engineering Systems, MIT Press, Cambridge, MA, 1961.

[52] F. Pfeiffer, C. Glocker, Multibody Dynamics with Unilateral Contacts, Wiley, New York, 1996.

[53] D. Sam-Haroud, B. Faltings, Consistency techniques for continuous constraints, Constraints: An International Journal 1 (1996) 85–118.

[54] J.M. Schumacher, Declarative, equation-based modeling of hybrid systems, Simulation News Europe 23 (1999) 6–9.

[55] W. Sweet, The glass cockpit, IEEE Spectrum (September 1995) 30–38.

[56] G.C. Verghese, B.C. Lévy, T. Kailath, A generalized state-space for singular systems, IEEE Trans. Automat. Control 26 (4) (1981) 811–831.