Modeling and Simulation Semantics for Hybrid Dynamic Physical Systems

Pieter J. Mosterman
Institute of Robotics and System Dynamics
DLR Oberpfaffenhofen
P.O. Box 1116
D-82230 Wessling
Germany

Gautam Biswas
Center for Intelligent Systems
Box 1679, Station B
Vanderbilt University
Nashville, TN 37235
U. S. A.

July 21, 1999

Abstract

This paper develops a mathematical framework for hybrid physical system models. Hybrid models of dynamic physical systems are characterized by *time scale* and *parameter* abstractions which lead to sequences of discrete changes in the system between modes of continuous behavior evolution. The goal is to develop formal execution semantics for characterizing hybrid behaviors in terms of three distinct modes of system operation: (i) *continuous*, (ii) *pinnacles*, and (iii) *mythical*. Continuous modes represent normal physical system behavior, where the system variables evolve continuously in time. Pinnacles, an artifact of time scale abstraction, represent behaviors at a point in real time. Mythical modes, an artifact of parameter abstractions, are defined by sequences of instantaneous local switching transitions in the hybrid model, and have no real existence on the time line. A key aspect of the work is the link established between the switching transitions and the *a priori* and *a posteriori* state vector values, which leads to the definition of recursive mode switching functions that govern the interactions between the continuous and discrete components of the system models. The mathematical specifications are developed into an implementation model that allows for a direct mapping of system components into model fragments, and facilitates simulation of physical system behavior. The simulation model encompasses discrete switching implemented as instantaneous transition functions, and continuous behavior generation based on differential equation models.

1 Introduction

Physical system dynamics, governed by the principles of *conservation of energy* and *continuity of power* [27], continuously evolve in real space as a function of time. System behaviors are often complex, and occur at different temporal and spatial scales. At a specified level of interest determined by the task at hand, the system model can be simplified by abstraction, so that the system seems to exhibit simpler *piecewise* continuous behavior. In other situations, physical systems with embedded digital control are designed to operate in multiple configurations or modes. Within each mode system behavior evolves continuously, but discrete mode changes can occur at points in time, resulting in discontinuities in overall system behavior. These mode changes, governed by state and external events, occur when system variables cross threshold values.

Our work is focused on developing formal methodologies for hybrid modeling and analysis of dynamic physical systems [12, 20]. In this framework, we have developed a set of unambiguous and consistent principles that define the interaction between the continuous and discrete modeling formalisms. This paper establishes a formal mathematical model and execution semantics for system behavior generation based on the physical principles derived in the

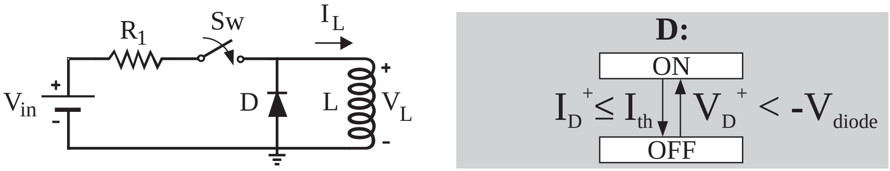


Figure 1: **Physical system with discontinuities.**

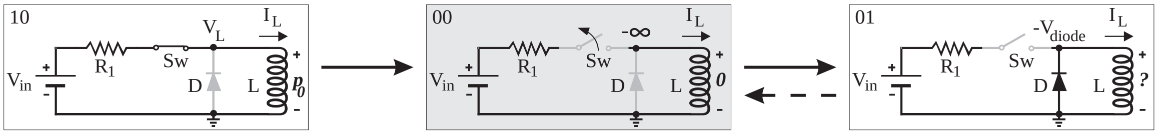


Figure 2: **A series of mode switches may occur.**

previous work.

Consider the electrical circuit shown in Fig. 1. The manually operated switch Sw and the diode D are considered to be ideal, i.e., they switch on and off instantaneously. When the switch is closed (mode 10 in Fig. 2), the circuit is complete, and the inductor draws a current to build up flux, p_0. The diode is inactive (off) in this mode of operation. When the switch is opened, creating an open circuit, the current drawn by the inductor drops to 0, causing its flux, p_0, to discharge instantaneously. The constituent relation for the inductor, $V_L = L\frac{dp}{dt}$, implies an infinite negative voltage across the diode because of the instantaneous change of flux in the inductor. However, as soon as the threshold voltage of the diode, V_{diode}, is exceeded the diode comes on instantaneously. The mode where the switch was open and the diode inactive (mode 00 in Fig. 2) never occurs in real time. If it did, the energy stored as flux in the inductor would be released instantaneously, and the observed behavior where the inductor discharges through the diode cannot be predicted by the idealized model. Therefore, the flux in the inductor is invariant across these idealized instantaneous switching changes in the system.

Typically a diode requires a threshold current $I_{th} > 0$ to remain on. If the inductor has built up a positive flux, the diode comes on when the switch opens. However, if the flux in the inductor is such that it is too low to maintain a current above the threshold value, I_{th}, the diode will switch off instantaneously. But the computed voltage drop when the switch and diode are both off exceeds the threshold voltage, V_{diode}, which implies the diode must come on again. This model predicts that the system goes into a loop of instantaneous changes (see dashed arrow in Fig. 2). Therefore, system behavior in real time does not progress or *diverge*, which conflicts with the notion that behavior of any physical system cannot halt in time.

This simple example illustrates a number of characteristics specific to hybrid system modeling and analysis:

- changes in the model configuration may cause discontinuities, which complicates the computation of the correct system state vector in the new mode of operation. In some cases, configuration changes may cause dependencies among state variables causing the dimensions of the state vector to change.
- in systems with multiple switching elements, a mode change may trigger a sequence of additional instantaneous changes. This further complicates behavior generation because:
    - the sequence of instantaneous mode changes must terminate in a real mode where behavior continues to evolve continuously, and
    - the state vector in the final mode has to be computed across the sequence of mode changes.

The rest of this paper extends these notions into formal mathematical specifications and execution semantics. An implementation model for physical systems and a simulation algorithm are developed. A number of successful applications of this approach are discussed elsewhere [14, 15, 16, 24].

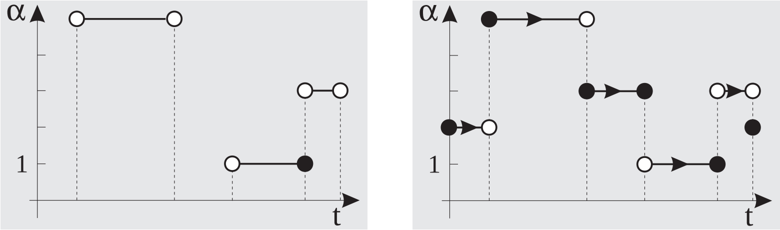


Figure 3: **A hybrid system (left) and a hybrid dynamic system (right).**

2 Background

A general hybrid system, illustrated in Fig. 3, operates on a domain that combines discrete and continuous dimensions. Behavior in this space is specified by piecewise continuous intervals $x_\alpha(p)$ a function of both $\alpha \in \aleph$ and $p \in \Re$ dimensions. Hybrid *dynamic* system behavior [6, 7] evolves over time, has an established direction of flow, and must necessarily cover the complete interval with *no gaps* on the time line (see Fig. 3). Piecewise continuous interval behavior is represented by well-behaved, continuous functions f, called *fields*, often specified as a set of ordinary differential equations [7]. An instance of temporal behavior in a field is called a *flow*, $\mathcal{F}$. Switching from one flow to another occurs at well-defined points in time when system variable values reach or exceed prespecified threshold values. This defines an *interval-point* paradigm where flows are piecewise continuous and any discontinuous changes that occur have to be *simple*, i.e., limit values exist at points of discrete switching [28].

Therefore, hybrid dynamic systems consist of three distinct subdomains:

- A *continuous* domain, T, with time, $t \in T$, as the special continuous variable.

- A *piecewise continuous* domain, V_α, that specifies variable flow, $x_\alpha(t)$, uniquely on the time-line.

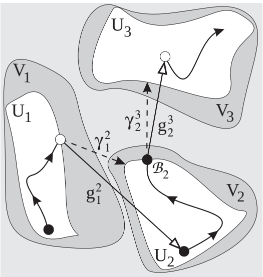


Figure 4: **A planar hybrid system [7].**

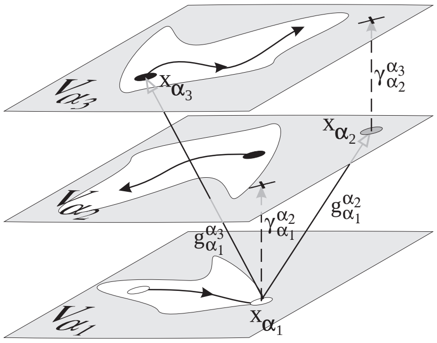
- A *discrete* domain, I, that captures the operative piecewise continuous domain, V_α.

We adapt notation similar to Guckenheimer and Johnson [7] and specify I to be a discrete indexing set, where $\alpha \in I$ represents the *mode* of the system. $\mathcal{F}_\alpha$ is a continuous C^2 flow on a possibly open subset V_α of $\Re^n$, called a *chart* (Fig. 4). The sub-domain of V_α where a continuous flow in time occurs is called a *patch*, $U_\alpha \subset V_\alpha$. The flows constitute the piecewise continuous part of the hybrid system. Points within the system are specified by $x_\alpha(t)$, a location in chart V_α in mode α at time t. An explicitly defined isolated point that does not embody continuous behavior is called a *pinnacle*, $\mathcal{P}_\alpha$. The discrete switching function γ_α^β is defined as a *threshold function* on V_α. If $\gamma_\alpha^\beta \leq 0$ in mode α, the system transitions to β, defined by the mapping $g_\alpha^\beta : \alpha \rightarrow \beta$. The piecewise continuous level curves $\gamma_\alpha^\beta = 0$, denoted as S_α^β, define patch boundaries. If a flow $\mathcal{F}_\alpha$ includes the level curve, S_α^β, it contains the *boundary point*, $\mathcal{B}_\alpha$ (see Fig. 4). A hybrid dynamic system is defined by the 5-tuple[1]

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

Trajectories in the system start at an initial point $x_{\alpha_1}(t)$ and if $\gamma_{\alpha_1}^{\alpha_2} > 0$, $\forall \alpha_2$, the point flows in α_1 as specified by $\mathcal{F}_{\alpha_1}$ until the minimal time t_s at

[1] Guckenheimer and Johnson refer to the respective parts as $< V_\alpha, X_\alpha, \mathcal{F}_\alpha, h_\alpha^\beta, \mathcal{T}_\alpha^\beta >$ [7].



V_{α_3}
x_{α_3}
$\gamma^{\alpha_3}_{\alpha_2}$
x_{α_2}
V_{α_2}
$\gamma^{\alpha_2}_{\alpha_1}$
$g^{\alpha_2}_{\alpha_1}$
$g^{\alpha_3}_{\alpha_1}$
V_{α_1}
x_{α_1}

Figure 5: **Redirected trajectory because the transported point is not in the domain of the new patch.**

which $\gamma^{\alpha_2}_{\alpha_1}(x_{\alpha_1}(t)) = 0$ for some α_2. Computing $x_{\alpha_1}(t_s^-) = \lim_{t \uparrow t_s} \mathcal{F}_{\alpha_1}(t)$ the transformation $g^{\alpha_2}_{\alpha_1}$ 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 $x_{\alpha_2}(t_s) = g^{\alpha_2}_{\alpha_1}(x_{\alpha_1}(t_s^-))$ is regarded as a new initial point.

If there exists $\alpha_3 \in I$, such that $\gamma^{\alpha_3}_{\alpha_2}(x_{\alpha_2}(t_s)) \leq 0$, the trajectory is immediately transferred to $g^{\alpha_3}_{\alpha_2}(x_{\alpha_2}(t_s)) \in V_{\alpha_3}$ (see Fig. 5). A characteristic of hybrid systems is the possibility of a number of these immediate changes occurring before a new patch is arrived at, where again a flow defined by a field governs system behavior [1, 7, 13, 26]. In general, this situation occurs if $\gamma^{\alpha_{k+1}}_{\alpha_k}$ transports a trajectory to α_{k+1}, and the initial point is transported by $g^{\alpha_{k+1}}_{\alpha_k}$ to a value that results in $\gamma^{\alpha_{k+2}}_{\alpha_{k+1}} \leq 0$, i.e., $g^{\alpha_{k+1}}_{\alpha_k}(x_{\alpha_k}) \notin U_{\alpha_{k+1}}$, 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 [7] and Deshpande and Varaiya [6] propose model semantics based on temporal sequences of abutting intervals.

$$\underbrace{\begin{matrix} x_0 & \mapsto & x_1 \\ [t_0 & & t_1] \end{matrix}}_{V_{\alpha_0}} \overset{\substack{\gamma^{\alpha_1}_{\alpha_0}(x) \\ g^{\alpha_1}_{\alpha_0}(x)}}{\hookrightarrow} \underbrace{\begin{matrix} x_1^+ & \mapsto & x_2 \\ [t_1 & & t_2] \end{matrix}}_{V_{\alpha_1}} \overset{\substack{\gamma^{\alpha_2}_{\alpha_1}(x) \\ g^{\alpha_2}_{\alpha_1}(x)}}{\hookrightarrow} \cdots \overset{\substack{\gamma^{\alpha_m}_{\alpha_{m-1}}(x) \\ g^{\alpha_m}_{\alpha_{m-1}}(x)}}{\hookrightarrow} \underbrace{\begin{matrix} x_m^+ & \mapsto & x_{m+1} \\ [t_m & & t_{m+1}] \end{matrix}}_{V_{\alpha_m}} . \tag{2}$$

Since these intervals overlap in time, a trajectory may be in several locations

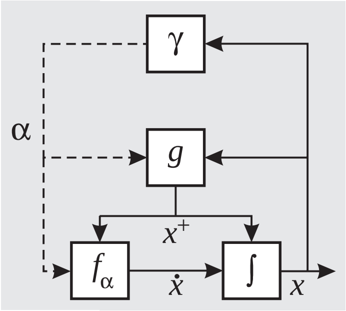


Figure 6: **A model of hybrid dynamic systems.**

at a point in time, t_s. Therefore, these points in time are complemented with an index that specifies their order of transition. During a series of discrete switches, $(t_s, i), (t_s, i+1) \ldots (t_s, n)$ the trajectory moves through these ordered points in time, repeatedly applying g_α^β, and depending on the ordering, different initial points of a new flow may be derived. Fig. 6 shows a schematic representation of semantics that produce a sequence of transitions of the form

$$\underbrace{\left\{\begin{array}{l} x = x_{\alpha_1} \\ \dot{x} = f_{\alpha_1}(x,t) \end{array}\right.}_{\alpha_1} \overset{\gamma_{\alpha_1}^{\alpha_2}(x)}{\xrightarrow{x_{\alpha_2}=g_{\alpha_1}^{\alpha_2}(x)}} \underbrace{\left\{\begin{array}{l} x = x_{\alpha_2} \\ \dot{x} = f_{\alpha_2}(x,t) \end{array}\right.}_{\alpha_2} \overset{\gamma_{\alpha_2}^{\alpha_3}(x)}{\xrightarrow{x_{\alpha_2}=g_{\alpha_2}^{\alpha_3}(x)}} \ldots \overset{\gamma_{\alpha_{m-1}}^{\alpha_m}(x)}{\xrightarrow{x_{\alpha_{m-1}}=g_{\alpha_{m-1}}^{\alpha_m}(x)}} \underbrace{\left\{\begin{array}{l} x = x_{\alpha_m} \\ \dot{x} = f_{\alpha_m}(x,t). \end{array}\right.}_{\alpha_m} \tag{3}$$

This paper extends the mathematical model of hybrid dynamic systems to physical system models by introducing constraints on the state vector function g_α^β during discrete transitions. Furthermore, it modifies the definition of γ_α^β to closely match requirements of object oriented modeling methods [4, 5, 8].

3 Hybrid Modeling of Physical Systems

Our previous work [12, 20], formulated a systematic approach to hybrid modeling of dynamic physical systems based on a local switching mechanism. The dynamically generated topology in a mode is used to translate these switching specifications to conditions based on state variables. Switching conditions may be expressed in terms of the state variables immediately before switching occurred (*a priori* values), or in terms of state variables computed by solving the initial value problem for the newly activated mode (*a posteriori* values). The *a priori* and *a posteriori* values may differ when the number of degrees of freedom in the system reduces and energy storage elements become dependent. Previous work [18, 20] identified two types of abstraction that lead to discontinuities in physical system models: (i) *parameter* abstraction, and (ii) *time scale* abstraction. In this paper, we show that switching conditions that result from parameter abstractions have to be in terms of *a posteriori* values and conditions due to time scale abstraction have to be in terms of *a priori* values. We develop formal modeling specifications as a mathematical model.

3.1 Abstractions in Physical System Models

Parameter abstractions occur when small, often parasitic, dissipation and storage parameters are abstracted away causing discontinuous changes in system behavior. Time scale abstractions represent behavior that occurs on a small time scale by a discontinuous change at a point in time. The behavior is not abstracted away but compressed to occur at a point in time.

3.1.1 Parameter Abstraction

Consider the ideal rigid body collision between a thin rod and a floor in Fig. 7. Upon collision, small deformation effects may occur which forces the vertical velocity of the rod-tip, A, to quickly become 0. If this phenomenon occurs on a time scale much smaller than the time scale of interest, these effects can be abstracted away. As a result, the model will show a discontinuous change when computing $v_{A,y}$ at the point of contact. Given the existence of Coulomb friction [10] between the rod and floor, the rod may stick and

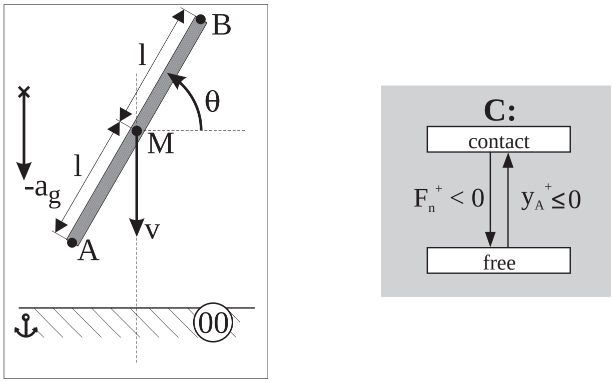


Figure 7: **A collision between a thin rod and a floor.**

rotate around the point of initial contact (mode α_{01} in Fig. 8). Alternately, if the rod-tip exerts a force in the horizontal direction that is larger than the product of the normal force and friction coefficient, i.e., $|F_{A,x}| > \mu F_n$, the rod starts to slide (mode α_{11} in Fig. 8). To evaluate which scenario occurs, the forces need to be calculated at the point of collision. Since the impact is idealized, the forces, referred to as *impulses* [3], 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 the impact, x, and immediately after the impact, x^+.

To calculate x^+, *conservation of state* is applied which requires that the total momentum before impact equals the total momentum after impact [16]. Upon contact, 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

$$\begin{cases} v_x^+ = l\omega^+ sin\theta \\ v_y^+ = -l\omega^+ cos\theta \end{cases} \tag{4}$$

Conservation of state yields the new state vector [12]

$$\omega^+ = \frac{\omega J - ml(cos\theta v_y - sin\theta v_x)}{J + ml^2}. \tag{5}$$

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}). However, in this

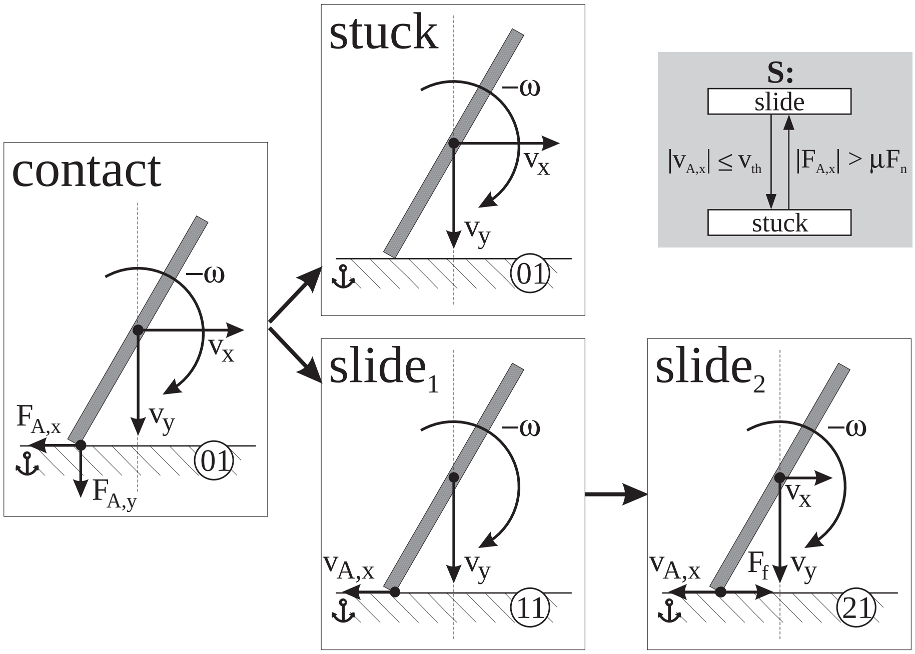


Figure 8: **Two possible scenarios after collision.**

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 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 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* values, $x = x^+$. Mathematically

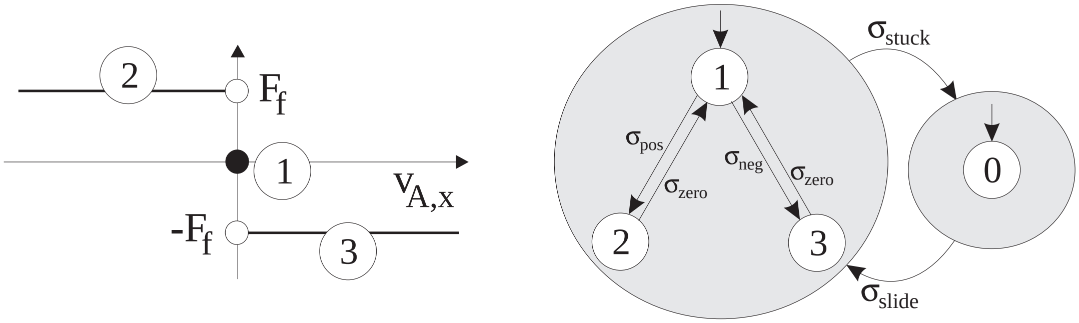


Figure 9: **Coulomb friction causes physical events.**

this can be represented as

$$\underbrace{\left\{\begin{array}{l} x^+ = g^{\alpha_1}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_1}(x,t) \end{array}\right.}_{\alpha_1} \xrightarrow{\gamma_{\alpha_1}^{\alpha_2}(x,x^+)} \underbrace{\left\{\begin{array}{l} x^+ = g^{\alpha_2}(x) \\ \\ \\ \end{array}\right.}_{\alpha_2} \xrightarrow{\gamma_{\alpha_2}^{\alpha_3}(x,x^+)} \underbrace{\left\{\begin{array}{l} x^+ = g^{\alpha_3}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_3}(x,t) \end{array}\right.}_{\alpha_3} \tag{6}$$

where α_2 is a so-called *mythical* mode [13, 26].

It is clear that switching in this example has to be based on x^+ rather than x, as shown in Fig. 7. This figure 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 disconnnects and lifts off the floor. Note that when the rod starts sliding, $v_{A,x} \neq 0$ which causes a friction force F_f in the opposite direction. When Coulomb friction is included, the force changes discontinuously around $v_{A,x} = 0$ (Fig. 9), and that causes another immediate mode change to α_{21}.

3.1.2 Time Scale Abstraction

Consider the perfect elastic collision of two bodies in Fig. 10. In reality, upon collision small elasticity effects store and return the kinetic energy over a short period of time. Becasue the time scale of this phenomenon is very small compared to the behavior of interest, it can be modeled as an instantaneous change at a point in time governed by Newton's collision rule

$$v_2^+ - v_1^+ = -\epsilon(v_2 - v_1) \tag{7}$$

and conservation of momentum ($m_1 = m_2$)

$$v_2^+ + v_1^+ = v_2 + v_1 \tag{8}$$

which yields

$$\begin{cases} v_1^+ = \frac{1-\epsilon}{2}v_1 + \frac{1+\epsilon}{2}v_2 \\ v_2^+ = \frac{1+\epsilon}{2}v_1 + \frac{1-\epsilon}{2}v_2 \end{cases} \tag{9}$$

This algebraic relation only holds at a point in time. Therefore, the switching specifications have to ensure that the *collide* mode is departed immediately after the state vector is updated, $x = x^+$. Mathematically this is represented as

$$\underbrace{\begin{cases} x^+ = g^{\alpha_1}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_1}(x,t) \end{cases}}_{\alpha_1} \xrightarrow{\gamma_{\alpha_1}^{\alpha_2}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_2}(x) \\ x = x^+ \\ \end{cases}}_{\alpha_2} \xrightarrow{\gamma_{\alpha_2}^{\alpha_3}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_3}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_3}(x,t) \end{cases}}_{\alpha_3} \tag{10}$$

where α_2 is a *pinnacle*. Fig. 10 illustrates that switching specifications have to be in terms of *a priori* state variable values. Moreover, a switching specification used for the falling rod to make the bodies disconnect, $F_{12} < 0$, cannot be used here because on collision $F_{12} > 0$, and the constraint would not move the ball into a *free* mode immediately after collision. Thus, $\dot{x} = f_\alpha(x,t)$ would be executed, but f_α does not exist for the collision mode where behavior is defined by algebraic relations (Eqs. 7 and 8).

3.1.3 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

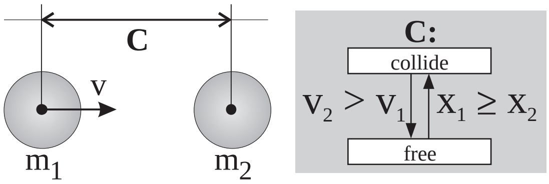


Figure 10: **A collision between two bodies.**

switching model uses *a priori* state values. In contrast, parameter abstraction abstracts away complex nonlinear behaviors, which 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 and have no real representation, and, therefore, do not affect the state vector, x. This is called the principle of *invariance of state* [14].

Mythical modes can be replaced by direct transitions to the final real mode (either a pinnacle, $\mathcal{P}_\alpha$, or a continuous mode, $\mathcal{F}_\alpha$). However, finding these direct transitions requires considerable effort and because of its global character needs to be performed whenever local changes to the model are made. A more pragmatic approach is to incorporate systematic techniques in the 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 [14, 15].

3.2 Physical Model Semantics

We can now present a mathematical model that embodies the physical abstraction semantics. It relies on a switching function γ_α^β that depends on values x_α, prior to the jump, and values x_α^+ just after the jump. The semantics are specified by the recursive relation between γ_α^β and g_α^β which takes the

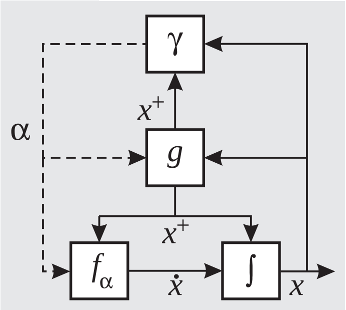


Figure 11: **A mathematical model based on a physical semantics.**

form

$$\begin{cases} x^+_{\alpha_k} = g^{\alpha_i}_{\alpha_k}(x_{\alpha_k}) \\ \gamma^{\alpha_{i+1}}_{\alpha_i}(x_{\alpha_k}, x^+_{\alpha_k}) \leq 0 \end{cases} \tag{11}$$

Note the α_k subscript in $g^{\alpha_i}_{\alpha_k}$. If g^{β}_{α} is independent of α, this results in the general sequence

$$\underbrace{\begin{cases} x^+ = g^{\alpha_1}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_1}(x,t) \end{cases}}_{\alpha_1} \xrightarrow{\gamma^{\alpha_2}_{\alpha_1}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_2}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_2}(x,t) \end{cases}}_{\alpha_2} \xrightarrow{\gamma^{\alpha_3}_{\alpha_2}(x,x^+)} \dots \xrightarrow{\gamma^{\alpha_m}_{\alpha_{m-1}}(x,x^+)} \underbrace{\begin{cases} x^+ = g^{\alpha_m}(x) \\ x = x^+ \\ \dot{x} = f_{\alpha_m}(x,t) \end{cases}}_{\alpha_m} \tag{12}$$

In this sequence, each mode, α, may be departed when any of the three assignment statements is executed. The difference between Eq. (3) and Eq. (12) is the use of (x, x^+) as argument to γ^{β}_{α}. This can be justified by physical system principles, and the resultant model is illustrated in Fig. 11. In relation to the model in Fig. 6, there is an additional feedback, x^+ into γ, which introduces a loop between g and γ. Overall, three cases can be distinguished (Fig. 12):

(a) *Mythical mode:* This occurs when $x^+ = g^{\alpha_i}(x)$ leads to $\gamma^{\alpha_{i+1}}_{\alpha_i}(x, x^+) \leq 0$. The immediate transition caused by x^+ results in the *integrator* ($\int$) element not playing a role in the computation of state vector x, which remains unchanged through the transition.

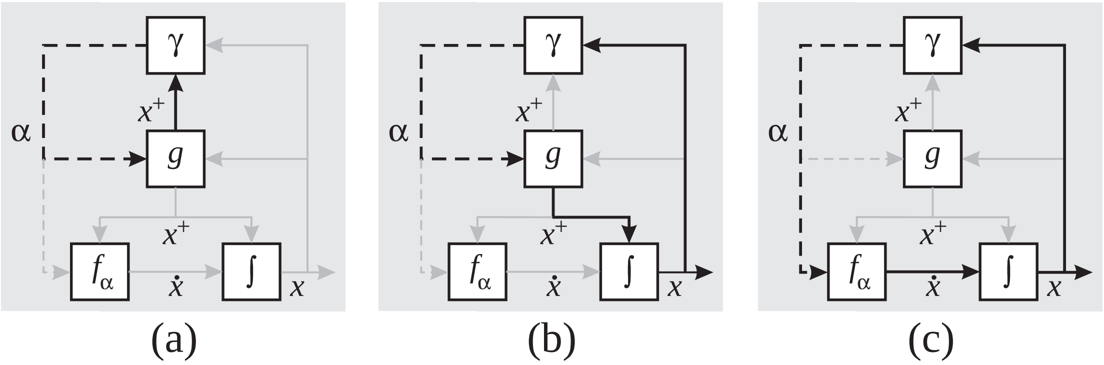


Figure 12: **Classes of modes of operation.**

(b) *Pinnacle:* This occurs when $x = x^+$ results in $\gamma_{\alpha_i}^{\alpha_{i+1}}(x, x^+) \leq 0$. Updating of x through the $\int$ element causes a mode transition, and, therefore, mode α_i only exists at a point in time.

(c) *Continuous mode:* This occurs when $\dot{x} = f(x, t)$ results in $\gamma_{\alpha_i}^{\alpha_{i+1}}(x, x^+) > 0$. The system goes into a period of continuous evolution, till $\gamma_{\alpha_i}^{\alpha_{i+1}}(x, x^+) \leq 0$, and the mode is departed.

Pinnacles and continuous modes are also referred to as *real* modes because they change x through the $\int$ element. Comparison with the general sequence in Eq. (3) shows that the g_α^β operation can be associated with the transition or the new operational mode. This is equivalent to the difference between Moore and Mealy state machines, and does not result in conceptual differences as far as behavior generation is concerned [23]. We chose to associate g_α^β with modes for two reasons: (i) it affects the energy distribution 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 implement a simulator if g is associated with a mode rather than a transition. For these reasons, we follow the Moore-type specification.

3.3 Notes

The point x_α initiates switching and controls point-interval evolution in time, whereas x_α^+ drives the recursive switching function that determines the intermediate mythical modes that are traversed before x_α is updated. Furthermore, switching conditions of the form $\gamma_{\alpha_i}^{\alpha_{i+1}} < 0$ are a special case of $\gamma_{\alpha_i}^{\alpha_{i+1}} \leq 0$. The latter includes $\mathcal{B}_{\alpha_i}$, the endpoint of flow $\mathcal{F}_{\alpha_i}$ in mode α_i. Consider a transition sequence $\alpha_k \longrightarrow \alpha_m \longrightarrow \alpha_n$. If $\gamma_{\alpha_m}^{\alpha_n}(x, g_{\alpha_k}^{\alpha_m}(x)) > 0$ then α_m is real, i.e., it is either a continuous mode or a pinnacle. If $(x = x_{\alpha_k}(t_s))$

$$\mathcal{F}_{\alpha_m} : g_{\alpha_k}^{\alpha_m}(x) = x \tag{13}$$

α_m is continuous.

If $g_{\alpha_k}^{\alpha_m}(x) \neq x$ then α_m is continuous and contains a flow if

$$\mathcal{F}_{\alpha_m} : \gamma_{\alpha_m}^{\alpha_n}(x, g_{\alpha_k}^{\alpha_m}(x)) > 0 \wedge \gamma_{\alpha_m}^{\alpha_n}(g_{\alpha_k}^{\alpha_m}(x), g_{\alpha_k}^{\alpha_m}(x)) > 0. \tag{14}$$

The first condition in Eq. 14 ensures the mode is real because there exists no α_n for which the transition function is ≤ 0. The second ensures that it is not a pinnacle.

α_m is a pinnacle if

$$\mathcal{P}_{\alpha_m} : \gamma_{\alpha_m}^{\alpha_n}(x, g_{\alpha_k}^{\alpha_m}(x)) > 0 \wedge \gamma_{\alpha_m}^{\alpha_n}(g_{\alpha_k}^{\alpha_m}(x), g_{\alpha_k}^{\alpha_m}(x)) \leq 0. \tag{15}$$

If α_m is continuous, the flow may not contain the boundary where it exits the mode to transition to α_n $(x = x_{\alpha_m}(t_s))$

$$\mathcal{F}_{\alpha_m} \setminus \mathcal{B}_{\alpha_m} : \mathcal{F}_{\alpha_m} \wedge \gamma_{\alpha_m}^{\alpha_n}(x, g_{\alpha_m}^{\alpha_n}(x)) \leq 0. \tag{16}$$

This implies that α_n is a pinnacle or contains a flow that includes its initial point,

$$\mathcal{F}_{\alpha_m} \setminus \mathcal{B}_{\alpha_m} \Rightarrow \mathcal{P}_{\alpha_n} \vee (\mathcal{F}_{\alpha_n} \cap \mathcal{B}^{\alpha_n}). \tag{17}$$

4 Model Verification

Our previous work [21] on hybrid modeling of physical systems has established three necessary conditions to ensure meaningful behavior generation: (i) *interval-point paradigm*, (ii) *divergence of time*, and (iii) *temporal evolution of state*. This section discusses these principles. Examples of colliding bodies and falling rod models are used to illustrate these principles.

4.1 The Interval-Point Paradigm

The interval-point paradigm requires that the generated system behavior cover all points on the real time line. Note that a hybrid model includes a number of piecewise continuous domains V_α, and discrete transitions between these domains. To ensure coverage, the transitions or jumps in behavior have to be one of two types:

1. `interval to point jump:` $\alpha_k \circ\!\!\rightarrow\!\!\bullet \alpha_m$. A discontinuity at t_s moves x_{α_k} from an interval where behavior is defined by a flow that does not include the boundary point $\mathcal{F}_{\alpha_k} \setminus \mathcal{B}_{\alpha_k}$, to a point, pinnacle $\mathcal{P}_{\alpha_{k_1}}$, or a new interval with flow $\mathcal{F}_{\alpha_{k_1}}$. In this case, $x_{\alpha_k}(t_s^-) = \lim_{t\uparrow t_s} \mathcal{F}_{\alpha_k}(t)$ is transported to $x^+_{\alpha_k} = g^{\alpha_{k_1}}_{\alpha_k}(x_{\alpha_k})$ (see Eq. (11)). The jump may generate recursive switching before the new flow or pinnacle is reached which terminates when $\gamma^{\alpha_m}_{\alpha_{k_i}}(x_{\alpha_k}, x^+_{\alpha_k}) > 0$. The state vector $x_{\alpha_m}(t_s) = x^+_{\alpha_k} = g^{\alpha_m}_{\alpha_k}(x_{\alpha_k})$.

2. `point to interval jump:` $\alpha_k \bullet\!\!\rightarrow\!\!\circ \alpha_m$. A discontinuity at t_s moves x_{α_k} from a point, pinnacle $\mathcal{P}_{\alpha_k}$, or boundary point $\mathcal{B}_{\alpha_k}$, to a flow, $\mathcal{F}_{\alpha_{k_1}}$. In this case, the state vector $x_{\alpha_k} = x_{\alpha_k}(t_s)$ is transported to $x^+_{\alpha_k}$ computed by Eq. (11) as $x^+_{\alpha_k} = g^{\alpha_{k_1}}_{\alpha_k}(x_{\alpha_k})$. The recursive switching terminates when $\gamma^{\alpha_m}_{\alpha_{k_i}}(x_{\alpha_k}, x^+_{\alpha_k}) > 0$ and the function value $x_{\alpha_m} = \lim_{t\downarrow t_s} \mathcal{F}_{\alpha_m}(t)$ is taken as $x^+_{\alpha_k} = g^{\alpha_m}_{\alpha_k}(x_{\alpha_k})$. A $\bullet\!\!\rightarrow\!\!\circ$ transition results in the activation of a flow, and the system evolves continuously before a new sequence of switches is initiated.

A trajectory of system behavior can be described as moving along a flow, $\mathcal{F}_{\alpha_k}$, with state vector $x_{\alpha_k}^+ = g_{\alpha_k}^{\alpha_k}(x_{\alpha_k}) = x_{\alpha_k}$. At time t_s, $\gamma_{\alpha_k}^{\alpha_{k_1}}(x_{\alpha_k}, x_{\alpha_k}^+) = 0$, i.e., a boundary point in the current patch is arrived at, and state vector $x_{\alpha_k} = lim_{t \uparrow t_s} \mathcal{F}_{\alpha_k}(t)$. This produces a discrete ○→● change, and the trajectory is transported from $x_{\alpha_k}(t_s^-)$ to the point $g_{\alpha_k}^{\alpha_{k_1}}(x_{\alpha_k}(t_s^-))$ which results in the *a posteriori* state vector $x_{\alpha_k}^+ = g_{\alpha_k}^{\alpha_{k_1}}(x_{\alpha_k}(t_s^-))$. If $x_{\alpha_k}^+ \notin U_{\alpha_{k_1}}$ the trajectory is *redirected* by the switching function $\gamma_{\alpha_{k_1}}^{\alpha_{k_2}}(x_{\alpha_k}, x_{\alpha_k}^+) \leq 0$ (see Fig. 5), and this immediately transports the trajectory to the point $g_{\alpha_k}^{\alpha_{k_2}}(x_{\alpha_k}(t_s^-))$. The trajectory may be redirected again by the switching function γ using $x_{\alpha_k}^+ = g_{\alpha_k}^{\alpha_{k_2}}(x_{\alpha_k}(t_s^-))$ to a new mode α_{k_3}. This recursive process continues until a state vector x_{α_m} is arrived at that is within the corresponding patch U_{α_m}, i.e., $\gamma_{\alpha_{k_i}}^{\alpha_m}(x_{\alpha_k}, x_{\alpha_k}^+) > 0$. After the successful transition is made, the *a priori* value is updated to $x_{\alpha_m}(t_s) = g_{\alpha_k}^{\alpha_m}(x_{\alpha_k}(t_s^-))$. If the new point is a pinnacle, again $\gamma_{\alpha_m}^{\alpha_{m_1}}(x_{\alpha_m}, x_{\alpha_m}^+) \leq 0$, ●→○ switching occurs, and the trajectory is transported from $x_{\alpha_m}(t_s)$ to $g_{\alpha_m}^{\alpha_{m_1}}(x_{\alpha_m}(t_s))$. The new value $x_{\alpha_m}^+$ can cause another sequence of recursive switches until a state vector $x_{\alpha_n}^+$ is generated within the domain of a patch U_{α_n}. When switching ends, a new flow, $\mathcal{F}_{\alpha_n}$ in V_{α_n}, is reached and the point $x_{\alpha_n}(t_s^+) = g_{\alpha_m}^{\alpha_n}(x_{\alpha_m}(t_s))$ becomes the initial point and system behavior continues to evolve. In real time, the active real modes (α_k, α_m, and α_n) follow each other immediately.

$$\overbrace{<\leftarrow, t_s>}^{\alpha_k}, \overbrace{[t_s]}^{\alpha_m}, \overbrace{< t_s, \rightarrow >}^{\alpha_n}. \tag{18}$$

Due to modeling abstractions, a system may move through a series of pinnacles before a new flow is arrived at. An example of four balls involved in a chain of collisions is illustrated in Fig. 13. Ball 1 which is initially moving collides with ball 2 resulting in a sequence of collisions, before ball 4 moves away. Fig. 14 shows the result of applying Newton's collision law to the sequence of collisions among the four balls. Ball 2 takes on the entire initial velocity of ball 1 before transferring it to ball 3, and so on. Each of the individual collisions represents a pinnacle, and the simulation indicates

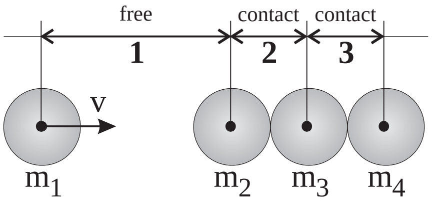


Figure 13: **A collision between four bodies.**

that the transfers in velocity all occur at the same point in time. Therefore, in simulation several pinnacles may be traversed at the same point in time. In reality, these collisions follow each other immediately in time [23]. Small elasticity coefficients in each ball store the kinetic energy passed on to it from the previous ball, before passing it on to the next ball, and this transformation process requires a small amount of time. Using a more complete model with the elasticity coefficients, simulation shows that the point where body 2 takes on all velocity does not coincide with the point where energy was transferred from body 2 to 3 (Fig. 15). No matter how small the elasticity coefficient used (this directly affects the transfer rate), it takes a finite amount of time for body 2 to take on the entire initial velocity from ball 1 before it can be passed on to ball 3. In reality, if this collision were modeled to occur at the exact same time, body 2 and body 3 would act as one body with mass $2m$, and the observed behavior would be distinctly different.

The sequence of collisions due to time scale abstraction should not be considered to occur at the same point in time as advocated by the traditional, mathematical, hybrid dynamic systems approach [1, 7, 25]. The sequences of pinnacles are captured by our the hybrid modeling semantics, but it violates the interval-point paradigm and may result in poorly defined models or even inconsistent models [23]. In certain situations, these models may predict a gain of energy [23].

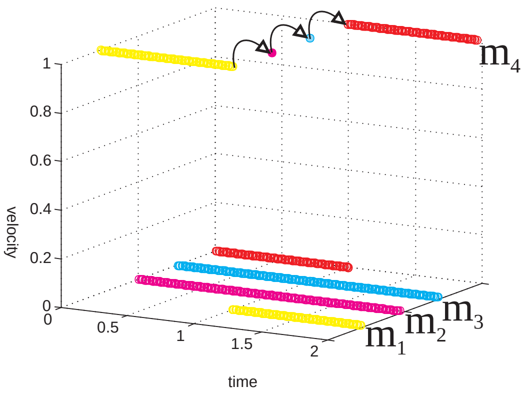


Figure 14: **Simulation of an ideal elastic collision between four bodies.**

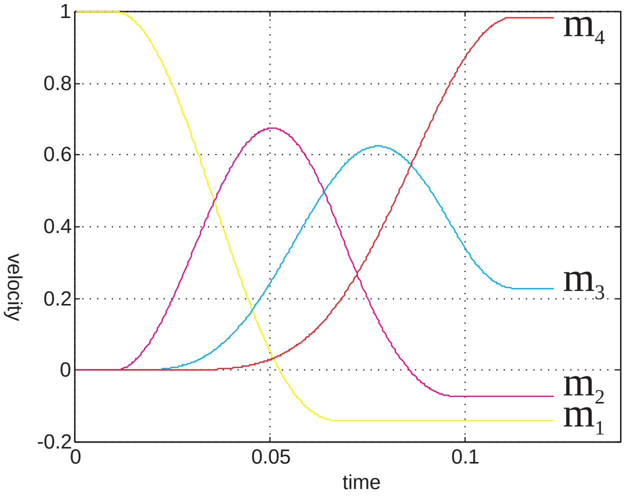


Figure 15: **An elastic collision between four bodies with small linear elasticity coefficients.**

4.2 Divergence of Time

A transition from an operational mode to another may generate a trajectory that goes through a sequence of mythical modes before a new real mode is reached. In this sequence, if a mythical mode is generated more than once, the trajectory can end up in a *loop* of discrete changes. The implication is that the behavior trajectory does not progress to a real mode where its behavior continues to evolve in time. This is in conflict with known behaviors of physical systems, i.e., they always evolve or diverge in time.

To illustrate divergence of time, consider the elastic collision between the two balls in Fig. 10. The transition function specifies that the two balls *collide* when $x_1 \geq x_2$. The collision rule is then applied to compute the new velocities v_1 and v_2, and this moves the system through a pinnacle into its *free* mode. Since no time has elapsed and $x_1 \geq x_2$ still holds, the transition function applies again, and the system switches to the *collide* mode.[2] This mode is departed immediately, which implies that it is mythical. However, the behavior trajectory is now caught in a loop of instantaneous mode changes where no time elapses, and the divergence of time principle is violated.

Divergence of time can be enforced by

- adding more detail to models so that discontinuous phenomena are now 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 may increase computational complexity of the model significantly. Modification of switching specifications require 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 impact velocities [3]. For collisions

[2] This also occurs if the state change due to the collision is modeled as a transition action rather than a separate state [23].

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 collision may be modified by adding the constraint $v_1 - v_2 \geq v_{th}$, In the limit as $v_{th} \rightarrow 0$ 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. Similarly, in the falling rod example, the condition for making contact can be extended by $v_{A,x} < 0$ (the floor is at rest, i.e., $v_{floor} = 0$) to ensure divergence of time. In previous work we have shown how a multi-dimensional energy phase space analysis can be applied to establish divergence of time [14].

4.3 Temporal evolution of state

When discontinuous state changes occur at t_s (Fig. 16), from an interval to a pinnacle ($x(t_s^-)$ to $x(t_s)$) or a pinnacle to an interval ($x(t_s)$ to $x(t_s^+)$), energy balance in the system may require the generation of Dirac pulses $\delta_1(t - t_s)$ and $\delta_2(t - t_s)$, respectively, to account for exchange of energy. 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, therefore, $\delta_c(t - t_s)$ is unknown and the conditions that 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 $\circ\!\!\rightarrow\!\!\circ$. To prevent ill-defined acausal models, an 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., $\delta_2 = 0$) [21]. The signal that is involved has to be continuous on the left-closed interval, $[t_s, \rightarrow>$ in time. This is the principle of temporal evolution of state.

Temporal evolution of state may be violated when one considers the stic-

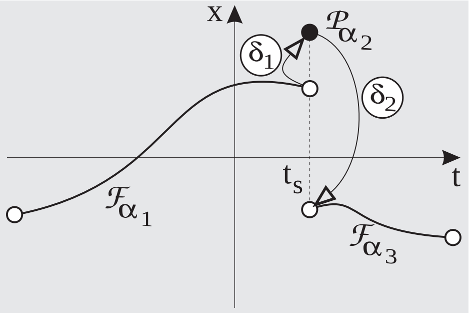


Figure 16: **Switching around a point may require two jumps.**

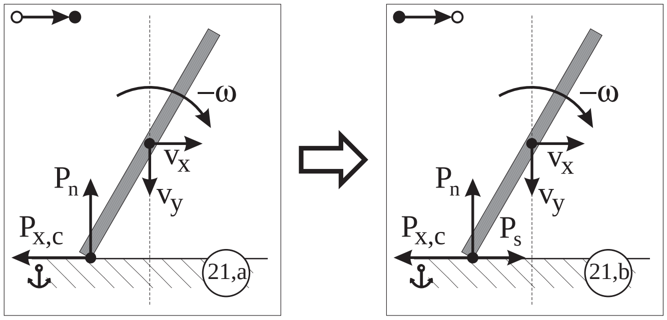


Figure 17: **Impulses upon collision when a sliding mode with stiction is reached.**

tion force between the rod and floor in Fig. 7. When the rod collides with the floor at time t_s, the horizontal and vertical velocities of the center of mass of the rod change discontinuously. So, $v_x(t_s^-) = \lim_{t\uparrow t_s} v_x(t)$ differs from $v_x(t_s)$ which results in a collision impulse $P_{x,c}(t_s)$ and $v_y(t_s^-) = \lim_{t\uparrow t_s} v_y(t)$ differs from $v_y(t_s)$ which results in a normal impulse $P_n(t_s)$ (see Fig 17). Since no other forces are active $P_{A,x}(t_s) = P_{x,c}(t_s)$. If $|P_{A,x}| > \mu P_n$ another mode change to $\alpha_{21,a}$ occurs, and the rod starts to slide. A stiction impulse may become active when the rod starts to slide (mode $\alpha_{21,b}$) 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 $v_x(t_s)$. 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, $|P_{A,x}| > \mu P_n$.

Applying the temporal evolution of state implies that the stiction impulse cannot become active after the rod-tip has started sliding. It has to be activated at the point in time t_s when it is determined that the rod starts sliding. In this case, the effect of the stiction impulse is taken into account along with $P_{x,c}(t_s)$, and $P_{A,x}(t_s)$ is derived correctly.

To ensure no δ pulses occur on $\bullet\!\!\rightarrow\!\!\circ$ switching, transition conditions that result in discontinuous changes in state variable values have to be of the form $\geq$ or $\leq$. 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 mechanical analysis [12]. For the falling rod, the state vector reduces in size upon collision, which is why the $=$ sign is included in the condition $y_A \leq 0$. Furthermore, the state vector may reduce in size when the rod gets stuck after sliding, which is why the $=$ sign is included in the $|v_{A,x}| \leq v_{th}$ condition.

5 A Hybrid Dynamic System Implementation Model

The mathematical specifications for hybrid system models developed in Sections 3 and 4 are applied to generate models of dynamic physical systems. For physical system models, the event generation function γ is based on continuous signals linked to the state variables in the system. A mode change may result in a change in functional relations between state variables and signals. The recomputed signal values can cause further mode changes.

The hybrid system model in Eq. (1) is extended to a complete hybrid physical system model defined by the 9-tuple [9, 21]

$$H =< I, \Sigma, \phi, X_\alpha, U_\alpha, f_\alpha, g^\alpha, h_\alpha, \gamma_\alpha^\beta > . \tag{19}$$

X_α and U_α the state and input vectors, and field f_α represent the continuous model in mode α. I the discrete indexing set corresponding to the possible modes in the system, Σ, the set of events that cause mode transitions, and

ϕ, the discrete state transition function represent the discrete model. γ and g defined earlier, and h, the signal generation function represent the interactions between the discrete and continuous models. These three components are described in greater detail next.

5.1 The Continuous Model

Continuous physical system behaviors are governed by energy interactions. Physical system behaviors are typically represented as state space models with the dynamic behaviors expressed as a set of ordinary differential equations (ODEs). For hybrid models, the continuous behavior in each real mode α is expressed as:

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

$t \in \Re$ and $\alpha \in \aleph$. $X_\alpha \in \Re^m$ is the continuous state vector, and $U_\alpha \in \Re^p$ is the vector of input signals. For every 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. 7) in the *stuck* mode, i.e., its rotational behavior at the point of contact, A is expressed as:

$$f_{\alpha_{01}} : \begin{cases} \dot{\omega} = \frac{-mlcos\theta}{J+ml^2} a_g \\ \dot{v}_x = lsin\theta\dot{\omega} \\ \dot{v}_y = -lcos\theta\dot{\omega} \end{cases} \tag{20}$$

v_x and v_y, the linear velocities, are not state variables since they are algebraic functions of ω. Note that the size of the state vector changes when the rod moves from the *free* fall mode, α_{00}, to the *contact* 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 *contact* mode to the *free* mode, and the state vector increases in size. No discontinuous changes can occur in this case.

5.2 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 transition function, ϕ, and transitions are invoked by events in a set Σ. In our compositional modeling approach, we systematically derive ϕ from a set of independent state machines that define local switching effects. A mode is defined as the combination of the individual states of the state machines. Theoretically, with n switching functions 2^n different modes are possible, but many of these modes do not have a physical representation. They may be traversed as mythical modes that occur between two real modes of operation. An important contribution of our work is to establish execution semantics that handle these sequences of mode changes correctly.

The discrete model can be implemented by Petri nets or finite state automata. It is represented as:

- $I = \{\alpha_0, \ldots, \alpha_k\}$, is a set of states describing the modes of the system.
- $\Sigma = \{\sigma_0, \ldots, \sigma_l\}$, is 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), $\Sigma = \Sigma_s \times \Sigma_x$.
- $\phi : I \times \Sigma \rightarrow I$, represents a discrete state transition function that defines the new mode after an event occurs.

5.3 Interactions

Lygeros, Godbole and Sastry [11] 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 verification 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:

- $S \in \Re^n$, the signals used for event generation.
- $h : X \times U \times I \rightarrow S$, returns signals from the input and state variable values in a given mode.
- $g : X \times I \rightarrow X^+$, 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^+.
- $\gamma : S \times S^+ \rightarrow \Sigma_s$, where Σ_s generates discrete events from the signal values. These signal values may be computed from the *a priori* state vector, S, or the *a posteriori* state vector, S^+.

The function γ generates discrete events when signals cross prespecified threshold values. The collision transition for the falling rod is defined by the following constraints:

$$\gamma : \begin{cases} y_A^+ \leq 0 \wedge v_{A,y} < 0 & \Rightarrow \sigma_{contact} \\ F_n^+ \leq 0 & \Rightarrow \sigma_{free} \end{cases} \tag{21}$$

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

$$h : \begin{cases} y_A = \int v_y dt - lsin\theta \\ v_{A,y} = m(v_y + l\omega cos\theta) \\ F_n = \begin{cases} 0 & \text{if } \alpha_{00} \\ m(\dot{v}_y - a_g) & \text{otherwise} \end{cases} \end{cases} \tag{22}$$

The generated events applied to the model may indicate that the system changes its mode of continuous operation. When mode switching occurs, the continuous state vector of the system may change. The function g transforms the continuous state vector as mode changes occur. In general, these transformations may be hard to derive, but for physical system models, this function has to satisfy the principle of *conservation of state.* When the falling rod first makes contact with the floor, conservation of state is applied to derive the state vector transformation function [17]:

$$g^{\alpha_{01}} : \left\{ \begin{array}{l} \omega^{+} = \frac{\omega J - ml(cos\theta v_y - sin\theta v_x)}{J + ml^2} \\ v_x^{+} = l\omega^{+} sin\theta \\ v_y^{+} = -l\omega^{+} cos\theta \end{array} \right. \tag{23}$$

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

$$g^{\alpha_{00}} : \left\{ \begin{array}{l} \omega^{+} = \omega \\ v_x^{+} = v_x \\ v_y^{+} = v_y \end{array} \right. \tag{24}$$

5.4 The Implementation Model

Fig. 18 shows a block diagram of the operation of a hybrid system corresponding to the 9-tuple model discussed above. Three loops can be identified in the interactions between continuous and discrete operations:

- Mythical modes can occur in the loops

  (i) $\phi \rightarrow h \rightarrow \gamma$. This results in mythical modes without discontinuous changes in the state vector. Sequences of mythical modes may occur on ●→○ transitions.

  (ii) $\phi \rightarrow g \rightarrow h \rightarrow \gamma$. This results in mythical modes with discontinuous changes in the state vector. The principle of temporal evolution of state indicates that this can only occur on ○→● transitions.

- Pinnacles arise from the $\phi \rightarrow g \rightarrow \int \rightarrow h \rightarrow \gamma$ loops since they require the state vector to be updated in the $\int$ element.

- Continuous modes arise from the $\phi \rightarrow f \rightarrow \int \rightarrow h \rightarrow \gamma$ loops. g is not part of this loop because $x^{+} = x$ during continuous behavior.

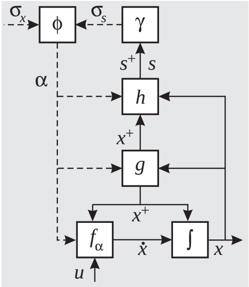


Figure 18: **A general hybrid system.**

5.5 Establishing Temporal Evolution of State

Dirac pulses occur at mode transitions when associated signals involved in derivative relationships change discontinuously because (i) the function that derives its value from the system state, h, changes between modes, or (ii) the system state itself undergoes a discontinuous change. For convenience, all signals used in event generation that have derivative relations are labeled h_d. In the falling rod example, $\dot{v}_y$ used in the F_n constraint (Eq. (22)) is in derivative form. So also is $\dot{v}_x$ used to determine transition to the sliding mode:

$$\gamma : \begin{cases} |F^+_{A,x}| - \mu F^+_n > 0 & \Rightarrow \sigma_{slide} \\ |v^+_{A,x}| - v_{th} \leq 0 & \Rightarrow \sigma_{stuck} \end{cases} \tag{25}$$

with signals other than F_n

$$h : \begin{cases} v_{A,x} = v_x - l\omega sin\theta \\ F_{A,x} = \begin{cases} 0 & \text{if } \alpha_{00} \\ m\dot{v}_x & \text{otherwise} \end{cases} \end{cases} \tag{26}$$

These represent all the mode changes $\alpha_k \rightarrow \alpha_m$ for which $h_d^{\alpha_k} \neq h_d^{\alpha_m}$. Equations (22) and (26) yield $\alpha_k = \alpha_{00} \wedge \alpha_m = \{\alpha_{01}, \alpha_{11}, \alpha_{21}\}$. To ensure that temporal evolution of state is not violated, it has to be verified that $\alpha_{00} \rightarrow \alpha_m$ and $\alpha_m \rightarrow \alpha_{00}$ are $\circ\!\!\rightarrow\!\!\bullet$ transitions.

Lemma 5.1 *$\alpha_{00} \rightarrow \alpha_m$ is a $\circ\!\!\rightarrow\!\!\bullet$ transition for $\alpha_m = \{\alpha_{01}, \alpha_{11}, \alpha_{21}\}$.*

Proof: $\alpha_i \rightarrow \alpha_{00} \rightarrow \alpha_m$ with $\alpha_i = \{\alpha_{01}, \alpha_{11}, \alpha_{21}\}$ (Equations (24) and (21)) establish

$$\left.\begin{array}{l} \alpha_{00} : (g_{\alpha_i}^{\alpha_{00}}(x) = x) \Rightarrow \mathcal{F}_{\alpha_{00}} \\ \sigma_{contact} : (\gamma_{\alpha_{00}}^{\alpha_m} \leq 0) \Rightarrow \mathcal{F}_{\alpha_{00}} \setminus \mathcal{B}_{\alpha_{00}} \end{array}\right\} \Rightarrow \alpha_{00} \circ\!\!\rightarrow\!\!\bullet\, \alpha_m$$

■

Lemma 5.2 *$\alpha_m \rightarrow \alpha_{00}$ is a $\circ\!\!\rightarrow\!\!\bullet$ transition for $\alpha_m = \{\alpha_{01}, \alpha_{11}, \alpha_{21}\}$.*

Proof: For $\alpha_m \rightarrow \alpha_{00}$ (Eq. (24) and Eq. (21))

$$\left.\begin{array}{l} \alpha_{00} : (g_{\alpha_m}^{\alpha_{00}}(x) = x) \Rightarrow \mathcal{F}_{\alpha_{00}} \\ \sigma_{free} : (\gamma_{\alpha_m}^{\alpha_{00}} \leq 0) \Rightarrow \mathcal{F}_{\alpha_{00}} \cap \mathcal{B}^{\alpha_{00}} \end{array}\right\} \Rightarrow \alpha_m \circ\!\!\rightarrow\!\!\bullet\, \alpha_{00}$$

■

Then dependent state variables, x_d, that cause discontinuous changes are identified. In the falling rod example, this occurs on a mode switch to α_{01} from either α_{11} or α_{21} (Eq. (23)). Since the $\alpha_{00} \rightarrow \alpha_{01}$ switching has been verified to satisfy temporal evolution of state, the implication is that $x_d \neq g_{\alpha_m}^{\alpha_{01}}(x_d)$ for $\alpha_m = \{\alpha_{11}, \alpha_{21}\}$ with $x_d = v_x$, and it has to be verified that $\alpha_m \rightarrow \alpha_{01}$ is a $\circ\!\!\rightarrow\!\!\bullet$ transition.

Lemma 5.3 *$\alpha_m \rightarrow \alpha_{01}$ is a $\circ\!\!\rightarrow\!\!\bullet$ transition for $\alpha_m = \{\alpha_{11}, \alpha_{21}\}$.*

Proof: From Eq. (25) γ generates σ_{slide} for $\alpha_m \rightarrow \alpha_{01} \rightarrow \alpha_i$ if $|F_{A,x}^+| - \mu F_n^+ > 0$, for $\alpha_i = \{\alpha_{00}, \alpha_{11}, \alpha_{21}\}, \alpha_m = \{\alpha_{11}, \alpha_{21}\}$. From Eq. (26) $F_{A,x}^+ = m\dot{v}_x^+$ and from Eq. (22) $F_n^+ = m\dot{v}_y^+ - ma_g$; for a mode change from $\alpha_{01} \rightarrow \alpha_i$, σ_{slide} is generated if ($\delta[a]$ represents a Dirac pulse with area a).

$$|m\delta[v_x^+ - v_x]| - \mu m\delta[v_y^+ - v_y] + \mu m a_g > 0.$$

For $x^+ = x = g_{\alpha_{01}}^{\alpha_i}$ this yields $\mu m a_g > 0$ with a_g the only negative constant, and, therefore, no immediate mode transition occurs. So,

$$\left.\begin{array}{l} \alpha_{01} : (\gamma_{\alpha_{01}}^{\alpha_i}(g_{\alpha_{01}}^{\alpha_i}(x), g_{\alpha_{01}}^{\alpha_i}(x))) > 0 \Rightarrow \mathcal{F}_{\alpha_{01}} \\ \sigma_{stuck} : (\gamma_{\alpha_m}^{\alpha_{01}} \leq 0) \Rightarrow \mathcal{F}_{\alpha_{01}} \cap \mathcal{B}^{\alpha_{01}} \end{array}\right\} \Rightarrow \alpha_m \circ\!\!\rightarrow\!\!\bullet\, \alpha_{01}$$

■

In this case $\alpha_{01} \rightarrow \alpha_m$ need not be proved because $x_d = g_{\alpha_{01}}^{\alpha_m}(x_d)$ and $h_d^{\alpha_{01}} = h_d^{\alpha_m}$.

5.6 Establishing Divergence of Time

For the colliding rod, divergence of time is violated in mode α_{01} if the horizontal rod-tip velocity falls below v_{th} but its angle and length are such that $|F_{A,x}| > \mu F_n$. This would cause σ_{slide} to be true, but in the sliding mode $|v_{A,x}| \leq v_{th}$ and the mode switch to σ_{stuck} occurs. This inconsistency can be eliminated by a modeling decision, where the constraints are modified so that σ_{stuck} is generated only if the forces in α_{01} are such that the switch to σ_{slide} will not occur instantaneously. This requires adding the constraint $|F_{A,x}^{\alpha_{01}}| \leq \mu F_n^{\alpha_{01}}$ to σ_{stuck}, where $F_{A,x}^{\alpha_{01}}$ and $\mu F_n^{\alpha_{01}}$ are calculated from $h(g^{\alpha_{01}}(x))$.

In general, transition conditions are likely to be more complex because of the greater interaction among modes. In such situations, an exhaustive multi-dimensional phase space analysis has to be performed [14, 15, 19].

6 The Hybrid Dynamic Simulator

The hybrid dynamic simulator implements the three components discussed in Section 5. The integrator is the core of the continuous simulation component. Since the state vector can change discontinuously when mode transitions occur, an integrator that looks back a number of time steps is inconvenient. To simplify the implementation, a forward Euler integrator that approximates derivatives by $\dot{x} = \frac{x_{k+1} - x_k}{\Delta t}$, or $x_{k+1} = f\Delta t + x_k$ is used because it looks back only one time step for each computation. For example, the state equations in Eq. (20) would be implemented as

$$f_{\alpha_{01}} : \begin{cases} \omega_{k+1} = \frac{-cos\theta_k}{J+ml^2} a_g \Delta t + \omega_k \\ v_{x,k+1} = lsin\theta_{k+1}\omega_{k+1} \\ v_{y,k+1} = -lcos\theta_{k+1}\omega_{k+1}. \end{cases} \tag{27}$$

The h function is also implemented in a similar manner. For the falling rod example, the h function implementation is listed below.

$$h : \begin{cases} y_A^+ = y_{M,k+1}^+ - lsin\theta_{k+1}^+ \\ v_{A,x}^+ = v_{x,k+1}^+ - lsin\theta_{k+1}^+\omega_{k+1}^+ \\ v_{A,y} = v_{y,k} + lcos\theta_k\omega_k \\ F_n^+ = \begin{cases} 0 & \text{if } \alpha_{00} \\ m(\frac{v_{y,k+1}^+ - v_{y,k}}{\Delta t} - a_g) & \text{otherwise} \end{cases} \\ F_{A,x}^+ = \begin{cases} 0 & \text{if } \alpha_{00} \\ m\frac{v_{x,k+1}^+ - v_{x,k}}{\Delta t} & \text{otherwise.} \end{cases} \end{cases} \tag{28}$$

The derivative terms in F_n and $F_{A,x}$ may produce Dirac pulses. If no discontinuous changes occur (e.g., $v_{y,k+1}^+ = v_{y,k+1}$) the forces F_n and $F_{A,x}$ can be estimated numerically using the above equation. When discontinuous changes occur (e.g., $v_{y,k+1}^+ \neq v_{y,k+1}$) the derivative term is a Dirac pulse of infinite magnitude that dominates all the other terms in the equation. However, the numerical approximation may compute a pulse magnitude that does not dominate the other terms in the equation, and this would result in incorrect simulation results. Therefore, discontinuous changes are tracked separately, and when they occur their time derivatives terms are replaced by Dirac pulses. Algorithm 3 implements the *derSignal* function. If $\Delta T = 0$ for a pinnacle, or if $h_d(x^+, \alpha) \neq h_d(x, \alpha)$ for a discontinuous change, the area of the corresponding Dirac pulse is returned. Otherwise, no discontinuous changes have occurred, and the Euler approximation is returned as the value of the derivative.

Algorithm 1 is the primary simulation module. The recursive relation, Eq. (11), is implemented as function *selMode* (Algorithm 2). To satisfy the interval-point requirement *selMode* is executed twice for each time step: (i) when continuous evolution terminates in a new mode,[3] and (ii) when pinnacles are traversed, and the ΔT argument in Algorithm 2 is set to 0.

The simulated trajectories of the rod in phase space for three different values of the friction coefficient (μ) are shown in Fig. 19. The system is

[3] Precision is improved by varying δt and employing a bisectional search.

Algorithm 1 Hybrid Simulation.

```
$x_{k+1} = x(0); x^+_{k+1} = x_{k+1}; x_k = x_{k+1}$
$t = 0; \delta t \leftarrow$ simulation time step
$\alpha_{m+1} = selMode(x^+_{k+1}, x_{k+1}, x_k, \alpha_0, 0)$
if $\alpha_{m+1} \neq \alpha_0$ then
  repeat
    $x_{k+1} = x^+_{k+1}$
    $\alpha_m = \alpha_{m+1}$
    $\alpha_{m+1} = selMode(x^+_{k+1}, x_{k+1}, x_k, \alpha_m, 0)$
  until $\alpha_{m+1} = \alpha_m$
end if{Initialization Completed}
repeat
  $t = t + \delta t$
  $x_{k+1} = f(x_k, t)\delta t$
  $x^+_{k+1} = x_{k+1}$
  $\alpha_{m+1} = selMode(x^+_{k+1}, x_{k+1}, x_k, \alpha_m, \delta t)$
  if $\alpha_{m+1} \neq \alpha_m$ then
    repeat
      $x_{k+1} = x^+_{k+1}$
      $\alpha_m = \alpha_{m+1}$
      $\alpha_{m+1} = selMode(x^+_{k+1}, x_{k+1}, x_k, \alpha_m, 0)$
    until $\alpha_{m+1} = \alpha_m$
  end if
  $x_k = x_{k+1}$
until simulation end
```

Algorithm 2 *selMode* $(x^+_{k+1}, x_{k+1}, x_k, \alpha_m, \Delta T)$.

```
$\alpha_{m+1} = \phi(\alpha_m, derSignal(x^+_{k+1}, x_{k+1}, x_k, \alpha_m, \Delta T))$
if $\alpha_{m+1} \neq \alpha_m$ then
  $x^+_{k+1} = g^{\alpha_{m+1}}(x_{k+1})$
  $ret\_val = selMode(x^+_{k+1}, x_{k+1}, x_k, \alpha_{m+1}, \Delta T)$
else
  $ret\_val = \alpha_m$
end if
```

Algorithm 3 *derSignal* $(x^+_{k+1}, x_{k+1}, x_k, \alpha_m, \Delta T)$.

```
if $\Delta T = 0$ then
  $s = h_i(x_{k+1}, \alpha_m)$
else
  $s = (h_d(x_{k+1}, \alpha_m) - h_d(x_k, \alpha_m))/\Delta T + h_i(x_{k+1}, \alpha_m)$
end if
if $\Delta T = 0 \vee h_d(x^+_{k+1}, \alpha_m) \neq h_d(x_{k+1}, \alpha_m)$ then
  $s^+ = h_d(x^+_{k+1}, \alpha_m) - h_d(x_{k+1}, \alpha_m)$
else
  $s^+ = (h_d(x^+_{k+1}, \alpha_m) - h_d(x_k, \alpha_m))/\Delta T + h_i(x^+_{k+1}, \alpha_m)$
end if
```

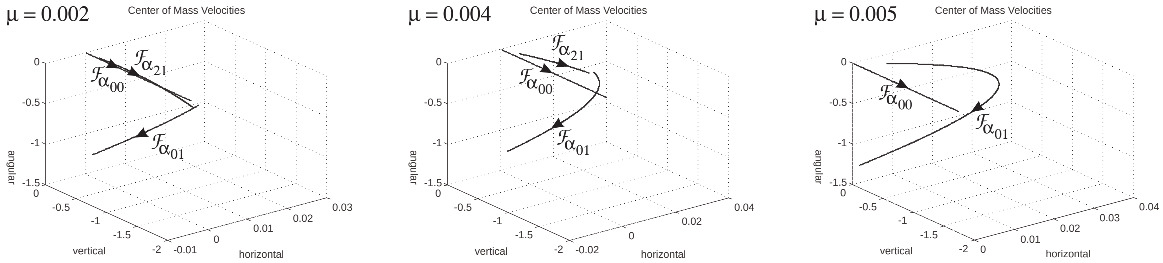


Figure 19: **A number of trajectories in phase space of the colliding rod,** $v_{th} = 0.0015, \theta = 0.862, l = -0.1, y_0 = 0.23$**.**

initialized with zero angular and linear velocities $((0, 0, 0))$. Once the rod is released, 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 $\mathcal{F}_{\alpha_{21}}$ (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. 19. Also, for simulations with $\mu = 0.002$ and $\mu = 0.004$, the sliding mode, α_{21} 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 $\mathcal{F}_{\alpha_{01}}$. 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 sliding starts, 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

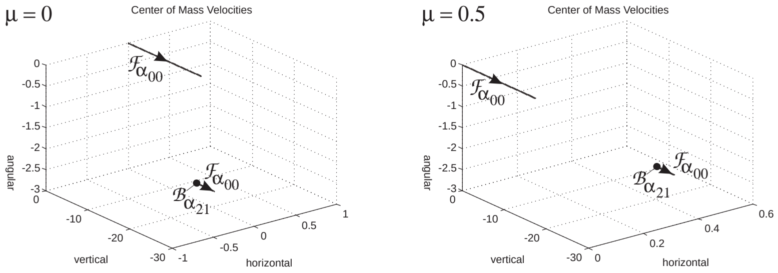


Figure 20: **A boundary in phase space of the colliding rod,** $v_{th} = 0.0015, \theta = 0.862, l = -10, y_0 = 23$**.**

rod is modeled to be perfectly non-elastic, i.e., there is no restitution of momentum difference in any of the operational modes ($\epsilon = 0$). Simulation results for this example are shown in Fig. 20. This simulation demonstrates how $\mathcal{B}_{\alpha_{21}}$ changes the state vector between the two flows in α_{00}. Note that a field governs behavior in α_{21}, so the corresponding point in phase space is a boundary point rather than a pinnacle.

7 Conclusions

This paper has developed a formal mathematical framework for analyzing hybrid behaviors of dynamic physical systems. This extends our previous work on compositional modeling of hybrid systems combining bond graph models with local discrete finite state automata [20]. *Parameter* and *time scale* abstractions are the key to developing systematic switching specifications that are governed by both *a priori* and *a posteriori* state vector values. Hybrid dynamic behaviors are piecewise continuous, and combine continuous behavior over intervals of time with pinnacles that represent a real behavior at a point in time (time scale abstraction) and mythical modes that have no real existence on the time line (artifact of parameter abstraction). The formal model specifications are made up of three components: (i) the continuous model, (ii) the discrete model, and (iii) the interaction model define a methodology for developing hybrid dynamic models of physical systems. A hybrid dynamic simulator is also developed from the mathematical execution semantics.

State vectors have to satisfy the principle of invariance of state across mythical mode changes, whereas discontinuous changes in state variables that occur at pinnacles are derived using the principle of conservation of state and explicitly defined interactions with the environment (represented as Dirac pulses). When mode changes occur, global specifications are derived dynamically from local switching functions. Model verification is performed by ensuring that the principles of divergence of time and temporal evolution of state are not violated. The use of local specifications often results in the system going through a sequence of discrete changes, but the modeling task is simplified because compositional modeling principles can be applied. This contrasts other approaches that have been employed for hybrid modeling (e.g., [1]), which require pre-defined global specifications of continuous system behavior in terms of differential equations. Our current research efforts are directed toward extending this methodology to design and analysis of embedded (computer-based) control of complex physical systems [21, 22, 24].

References

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

[2] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn,

and H. Rischel, editors, *Lecture Notes in Computer Science*, volume 736, pages 209–229. Springer-Verlag, 1993.

[3] Raymond M. Brach. *Mechanical Impact Dynamics.* John Wiley and Sons, New York, 1991.

[4] F.E. Cellier, H. Elmqvist, and M. Otter. Modelling from physical principles. In W.S. Levine, editor, *The Control Handbook*, pages 99–107. CRC Press, Boca Raton, FL, 1991.

[5] Theo J.A. de Vries, Peter C. Breedveld, and Piet Meindertsma. Polymorphic modeling of engineering systems. In *Proceedings of the International Conference on Bond Graph Modeling*, pages 17–22, San Diego, California, 1993.

[6] Akash Deshpande and Pravin Varaiya. Viable control of hybrid systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, volume 999, pages 128–147. Springer-Verlag, 1995. Lecture Notes in Computer Science.

[7] John Guckenheimer and Stewart Johnson. Planar hybrid systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, volume 999, pages 202–225. Springer-Verlag, 1995. Lecture Notes in Computer Science.

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

[9] Bengt Lennartson, Michael Tittus, Bo Egardt, and Stefan Pettersson. Hybrid systems in process control. *IEEE Control Systems*, pages 45–56, October 1996.

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

[11] John Lygeros, Datta Godbole, and Shankar Sastry. Simulation as a tool for hybrid system design. In *1994 AIS Conference on Distributed Interactive Simulation Environments*, 1994.

[12] Pieter J. Mosterman. *Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis*. PhD dissertation, Vanderbilt University, 1997.

[13] Pieter J. Mosterman and Gautam Biswas. Modeling Discontinuous Behavior with Hybrid Bond Graphs. In *Qualitative Reasoning Workshop*, pages 139–147, Amsterdam, May 1995. University of Amsterdam.

[14] Pieter J. Mosterman and Gautam Biswas. A Formal Hybrid Modeling Scheme for Handling Discontinuities in Physical System Models. In *AAAI-96*, pages 985–990, Portland, Oregon, August 1996. AAAI Press, 445 Burgess Drive, Menlo Park, CA, 94025.

[15] Pieter J. Mosterman and Gautam Biswas. Verification of dynamic physical system models. In *ASME-96*, pages 707–714, Atlanta, GA, November 1996.

[16] Pieter J. Mosterman and Gautam Biswas. Formal Specifications for Hybrid Dynamical Systems. In *IJCAI-97*, August 1997.

[17] Pieter J. Mosterman and Gautam Biswas. Formal Specifications from Hybrid Bond Graph Models. In *Qualitative Reasoning Workshop*, pages 131–142, Cortona, Italy, June 1997.

[18] Pieter J. Mosterman and Gautam Biswas. Principles for Modeling, Verification, and Simulation of Hybrid Dynamic Systems. In *Fifth International Conference on Hybrid Systems*, Notre Dame, Indiana, September 1997.

[19] Pieter J. Mosterman and Gautam Biswas. Hybrid modeling specifications for dynamic physical systems. In *1997 International Conference*

on Bond Graph Modeling and Simulation (ICBGM '97), pages 162–167, Phoenix, AZ, January 1997. Society for Computer Simulation, Simulation Councils, Inc.

[20] Pieter J. Mosterman and Gautam Biswas. A theory of discontinuities in physical system models. *Journal of the Franklin Institute*, 335(3), 1998.

[21] Pieter J. Mosterman, Gautam Biswas, and Janos Sztipanovits. Hybrid modeling and verification of embedded control systems. In *Proceedings of the* 7^{th} *IFAC CACSD '97 Symposium*, pages 21–26, Gent, Belgium, April 1997.

[22] Pieter J. Mosterman, Gautam Biswas, and Janos Sztipanovits. A hybrid modeling and verification paradigm for embedded control systems. *Control Engineering Practice*, 1998.

[23] Pieter J. Mosterman, Jan F. Broenink, and Gautam Biswas. Model Semantics and Simulation of Time Scale Abstractions in Collision Models. In *Eurosim 98*, Helsinki, Finland, April 1998. in review.

[24] Pieter J. Mosterman, Feng Zhao, and Gautam Biswas. Model semantics and simulation for hybrid systems operating in sliding regimes. In *AAAI Fall Symposium on Model Directed Autonomous Systems*, 1997.

[25] Xavier Nicollin, Joseph Sifakis, and Sergio Yovine. From atp to timed graphs and hybrid systems. In J.W. Bakkers, C. Huizing, W.P. de Roeres, and G. Rozenberg, editors, *Lecture Notes in Computer Science*, volume 600, pages 549–571, Mook, The Netherlands, June 1991. Real Time: Theory and Practice.

[26] T. Nishida and S. Doshita. Reasoning about discontinuous change. In *Proceedings AAAI-87*, pages 643–648, Seattle, Washington, 1987.

[27] Henry M. Paynter. *Analysis and Design of Engineering Systems.* The M.I.T. Press, Cambridge, Massachusetts, 1961.

[28] W. Rudin. *Principles of Mathematical Analysis.* McGraw-Hill, New York, 3 edition, 1976.